|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_ ***ИУК «Информатика и управление»\_\_\_\_\_\_\_\_\_***

**КАФЕДРА** \_\_ ***ИУК5 «Системы обработки информации»***

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе на тему:**

***Разработка централизованного чата на основе Windows API***

по дисциплине ***Системное программирование***

Студент гр. ИУК5-42Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Дорофеев И.К. )

(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

Оценка руководителя \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка защиты \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка проекта \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка по пятибалльной шкале)

Комиссия: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

Калуга, 2021

Калужский филиал   
федерального государственного бюджетного образовательного учреждения высшего образования

***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»   
(КФ МГТУ им. Н.Э. Баумана)***

УТВЕРЖДАЮ

Заведующий кафедрой **\_\_ИУК5\_\_\_**

\_\_\_\_\_\_\_\_\_\_\_(Е.В. Вершинин)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине ***Системное программирование***

Студент\_\_Дорофеев И.К. ИУК5-42Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы, индекс группы)

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы)

График выполнения проекта: 25% к\_4\_нед., 50% к\_7\_нед., 75% к\_10\_нед., 100% к\_14\_нед.

***1. Тема курсового проекта***

***Разработка централизованного чата на основе Windows API***

***2. Техническое задание***

*Разработать приложение с использованием функций Windows API для отправки текстовых сообщений по средствам Интернета с использованием централизованного чата\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***3. Оформление курсового проекта***

3.1. Расчетно-пояснительная записка на\_\_\_\_\_\_\_ листах формата А4.

3.2. Перечень графического материала КП (плакаты, схемы, чертежи и т.п.)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021г.

Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись) (Ф.И.О.)

Задание получил\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_2021г.

(подпись) (Ф.И.О.)

Примечание:

Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

**СОДЕРЖАНИЕ**

[1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ 4](#_Toc72626527)

[1.1. Наименование 4](#_Toc72626528)

[1.2. Основание для разработки 4](#_Toc72626529)

[1.3. Исполнитель 4](#_Toc72626530)

[1.4. Цель разработки 4](#_Toc72626531)

[1.5. Содержание работы 5](#_Toc72626532)

[1.5.1. Задачи, подлежащие решению: 5](#_Toc72626533)

[1.5.2 Требования к архитектуре АСОИ 5](#_Toc72626534)

[1.5.3. Требования к составу программных компонентов 5](#_Toc72626535)

[1.5.4. Требования к прикладным программам 6](#_Toc72626536)

[1.5.5. Требования к входным/выходным данным 6](#_Toc72626537)

[1.5.6. Требования к временным характеристикам 6](#_Toc72626538)

[1.5.7. Требования к составу технических средств 6](#_Toc72626539)

[1.6. Этапы разработки 7](#_Toc72626540)

[1.7. Техническая документация, предъявляемая по окончании работы 7](#_Toc72626541)

[1.7. Дополнительные условия 8](#_Toc72626542)

[2. НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ 9](#_Toc72626543)

[2.1. Постановка задачи проектирования 9](#_Toc72626544)

[2.2. Описание предметной области 9](#_Toc72626545)

[2.3. Анализ аналогов и прототипов 10](#_Toc72626546)

[2.4. Перечень задач подлежащих решению в процессе разработки 13](#_Toc72626547)

[2.5. Обоснование выбора инструментов и платформы для разработки 13](#_Toc72626548)

[3. ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ 17](#_Toc72626549)

[3.1. Разработка структуры приложения 17](#_Toc72626550)

[3.2. Используемые функции Win32API 18](#_Toc72626551)

[3.3. Разработка архитектуры приложения 20](#_Toc72626552)

[3.4. Разработка и реализация алгоритмов приложения 21](#_Toc72626553)

[3.5. Разработка интерфейса взаимодействия пользователя с системой 26](#_Toc72626554)

[4. ПРОЕКТНО-ТЕХНОЛОГИЧИСКАЯ ЧАСТЬ 28](#_Toc72626555)

[4.1. Разработка руководства пользователя 28](#_Toc72626556)

[ЗАКЛЮЧЕНИЕ.................................................................................................................................34](#_Toc72626557)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТУРАТУРЫ 35](#_Toc72626558)

[ПРИЛОЖЕНИЕ А............................................................................................................................ 36](#_Toc72626559)

[ПРИЛОЖЕНИЕ Б............................................................................................................................. 38](#_Toc72626560)

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

## Наименование

Прикладное программное обеспечения на основе Win Api для централизованного чата.

## Основание для разработки

В совеменном мире онлайн общение играет важную роль. Люди не всегда имеют возможность позвонить друг другу или встретится лично. Чаты, форумы, службы мгновенных сообщений - все эти возможности делают общение современного человека удобным, быстрым и простым. Преимущества такого вида общения очевидны: это и скорость обработки данных, возможность простмотра предыдущих сообщений, гибкость индивидуальной настройки.

## Исполнитель

Студент группы СОИ.Б-42 Дорофеев И.К.

## Цель разработки

Задачи проектирования:

1. овладение первичными навыками ведения научно-исследовательской, проектной и производственно-технологической деятельности, развитие творческих навыков;
2. подготовка к выполнению выпускной квалификационной работы;
3. усвоение методов грамотного ведения, оформления и редактирования технической документации;
4. исследование предметной области, изучение принципов системного программирования и работы WinAPI функций;
5. тестирование разработанного программного обеспечения.

Целью разработки является исследование и анализ методов обмена информацией через сеть Интернет.

## Содержание работы

### 1.5.1. **Задачи, подлежащие решению:**

исследование существующих методов обмена информацией в сети Интернет;

выбор критерий для оценки методов;

оценка методов в соответствии с выбранными критериями;

1. анализ полученных результатов;
2. разработка многопользовотельского чата;
3. подготовка расчетно-пояснительной записки и графических листов;
4. подготовка презентации и речи для защиты курсовой работы;
5. защита курсовой работы.

### 1.5.2 Требования к архитектуре АСОИ

К архитектуре предъявляются следующие требования:

модульная архитектура программного обеспечения, в которой каждый модуль представляет собой один из методов внедрения программного кода;

каждый метод должен быть представлен в виде отдельного класса;

классы должны быть реализованы в виде иерархии классов, в которой классы, реализующие методы внедрения программного кода   
в сторонний процесс, наследуется от абстрактного виртуального класса.

### 1.5.3. Требования к составу программных компонентов

Программный комплекс должен состоять из следующих программных компонентов:

1. исполняемый файл реализующий сервер ;
2. исполняемый файл реализующий клиент многопользовательского чата.

### 1.5.4. Требования к прикладным программам

Для работы программного комплекса необходимы:

1. Microsoft Windows 8/10;
2. Microsoft Visual Studio 2019;
3. минимальный набор драйверов, обеспечивающих   
   работоспособность ПК.

### 1.5.5. Требования к входным/выходным данным

Входные данные:

1. Имя пользователя;
2. Сообщение пользователя.

Выходные данные:

1. IP адрес и порт в окне cервера;
2. Список сообщений.

### 1.5.6. Требования к временным характеристикам

Требования к временным характеристикам программы   
не предъявляются.

### 1.5.7. Требования к составу технических средств

Для функционирования системы необходимы:

* процессор: 1 ГГц и быстрее с поддержкой PAE, NX и SSE2;
* RAM: 1 Гбайт (32 бит) или 2 Гбайт (64 бит);
* HDD: 500 Мбайт (32 бит) или 700 Мбайт (64 бит);
* видеокарта: поддержка Microsoft DirectX 9 с драйвером WDDM;
* монитор;
* сетевая карта: частота более 500 МГц, Тип видеопамяти: DDR3;
* клавиатура;
* мышь.

## **Этапы разработки**

исследование существующих планировщиков задач;

изучение принципов системного программирования и работы WinAPI функций;

выбор методов планирования задач по определенному времени;

реализация методов планирования задач;

тестирование приложения;

## 1.7. **Техническая документация, предъявляемая по окончании работы**

По окончанию работы предъявлена расчетно-пояснительная записка в состав которой входят:

* техническое задание;
* научно-исследовательская часть;
* проектно-конструкторская часть;
* проектно-технологическая часть.

Также должна быть предоставлена графическая часть работы, выполненная формате А1 на 2 листах, в которую входят:

* демонстрационные чертежи;
* алгоритмические схемы.

## **Дополнительные условия**

язык программирования С/С++;

использование Windows API функций для реализации методов планировщика задач;

среда разработки Visual Studio 2019;

тип приложения – оконное;

1. интерфейс должен предоставлять пользователю окно клиента с информацией о IP сервера, механизмом аворизации, полем для введения сообщения, кнопкой откравки сообщения и списком сообщений, которые были отправлены ранее. В клиенте сервера интерфейс представляет собой поле введения IP и кнопкой начала работы сервера.

# НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

## Постановка задачи проектирования

Задача курсовой работы состоит в разработке программы –многопользовательского чата на основе Windows API, который позволяет отправлять текстовые сообщения по стредствам интернета.

Необходимо разработать удобный пользовательский интерфейс для работы с программой.

## Описание предметной области

Многопользовательский чат предназначен для общения двух и более пользователей в одном чате. Ориентирован для общения или интеграции в более сложные многопользовательсие системы для общения пользователей, такие как бизнес програмы, месседжеры, социальные сети.

## Анализ аналогов и прототипов

Сегодня повсеместное использование многопользовательских чатов существенно облегчает работу человека в самых различных сферах. Впрочем, представить себе жизнь без таких приложений практически невозможно - ведь в современном мире существует много месседжеров и социальных сетей, а текже многопользовательские чаты интегрируются в почти в любой сайт или игру..

6 августа 1991 года можно считать вторым днём рождения сети Интернет. В этот день Тим Бернерс-Ли запустил первый в мире веб-сайт на первом в мире веб-сервере, доступном по адресу [info.cern.ch](http://info.cern.ch/). Ресурс определял понятие «Всемирной паутины», содержал инструкции по установке веб-сервера, использования браузера и т.п. Этот сайт также являлся первым в мире интернет-каталогом, потому что позже Тим Бернерс-Ли разместил и поддерживал там список ссылок на другие сайты. Это было знаковое начало, которое сделало интернет таким, каким мы его знаем сейчас.

**Первая социальная сеть Classmates**

Первая социальная сеть в мире — «Одноклассники». Речь идёт об американской сети Classmates, которая появилась в 1995 году и являла собой то же самое, о чём вы подумали в первом предложении абзаца. На старте пользователь выбирает штат, школу, год выпуска и после регистрации погружается в особую атмосферу такой социальной сети. Кстати, сайт претерпел редизайн и существует и сейчас — более того, он по-прежнему очень популярен.

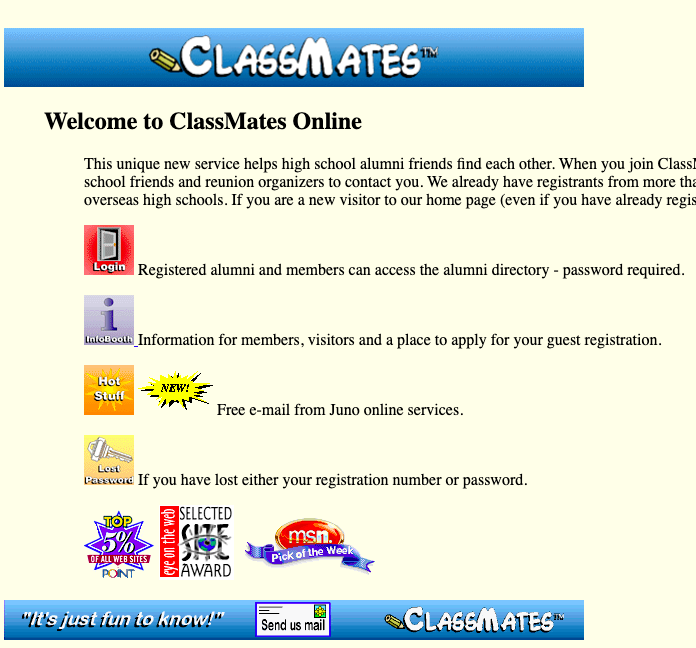


Рис.1 Пользовательский интерфейс сайта " Classmates".

**Месседжер ICQ.**

Первым месседжером является программа ICQ. История мессенджера начинается в 1996 году, когда израильская компания Mirabilis запустила ICQ. В ней были и многопользовательские чаты, и поддержка передачи файлов, и поиск по пользователям, и много чего ещё.

Первые пользователя Интернета были в восторге от мессенджера с ярким цветком в логотипе, они помнят свой номер ICQ (UIN) еще и сегодня. В то время к UIN относились как к номеру своего телефона, а шестизначные номера ICQ стали настоящим раритетом, их продают за большие деньги на аукционе eBay. UIN был важен, прежде всего, потому, что только с его помощью можно было найти друзей. Привычной сейчас возможности пролистать «списки контактов» тогда еще не было. Именно этот принцип работы мессенджера делал атмосферу в «аське» такой непринужденной. Спустя всего лишь два года после появления ICQ (от английского: «I seek you» — «Я тебя ищу») мессенджер был куплен американским медиа-концерном AOL за 407 миллионов долларов.

Технические возможности ICQ

Когда вы открывали приложение, первое, что вы видели, - список, содержащий все ваши контакты и их статусы, указывающие «В сети» ли они.

Большие мигающие иконки сообщали вам, кто в данный момент доступен, занят или отошёл.

Звуковые уведомления позволяли узнать об изменении статуса ваших друзей.

По умолчанию, при входе в систему вы тоже транслировали всему миру, что вы в сети.

Что-то обсуждать с собеседником можно было, только когда он находился в сети. Были и групповые чаты, но их использование также было связано со множеством проблем и ограничений. Вы могли присоединиться к групповому чату лишь войдя в сеть, но при этом участники чата не обязательно могли быть в сети одновременно с вами. Любой человек мог быть внезапно отключен по какой-либо причине (неустойчивость соединения или телефонный звонок). Вы пропускали все разговоры, происходившие в то время, пока вы были оффлайн. Кроме того, групповые чаты были постоянно завалены автоматическими сообщениями об изменении статуса контактов: Иван присоединился, Антон отключился.

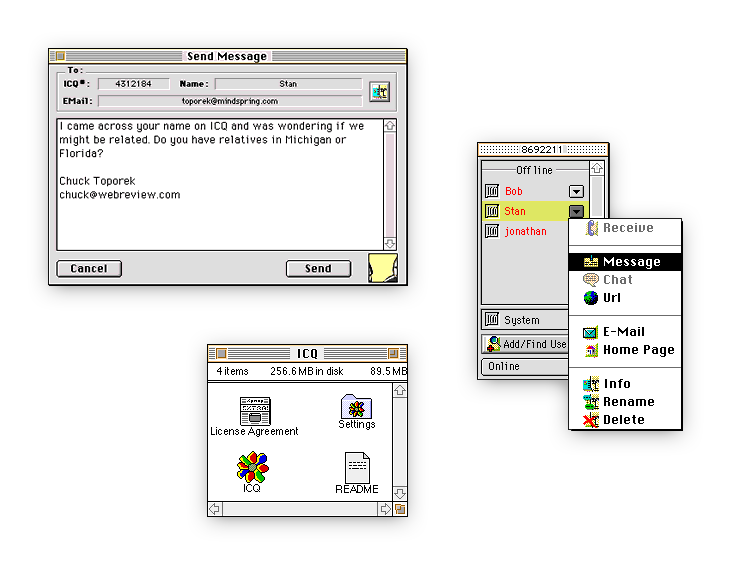


Рис.2 Интерфейс программы " ICQ ".

**Социальная сеть «Фейсбук»**

  Страсть к компьютерам у самого молодого миллиардера современности — Марка Цукерберга — появилась ещё в 6-ом классе. Он начинал свой тернистый путь с известного языка программирования С++ и в юные годы разработал компьютерную игру «Риск». Затем была программа, которую Цукерберг создал для своего отца. С помощью неё сотрудники его компании могли общаться между собой.

4 февраля 2004 г. в небольшой комнатке общежития Гарвардского университета, была запущена социальная сеть «The Facebook».

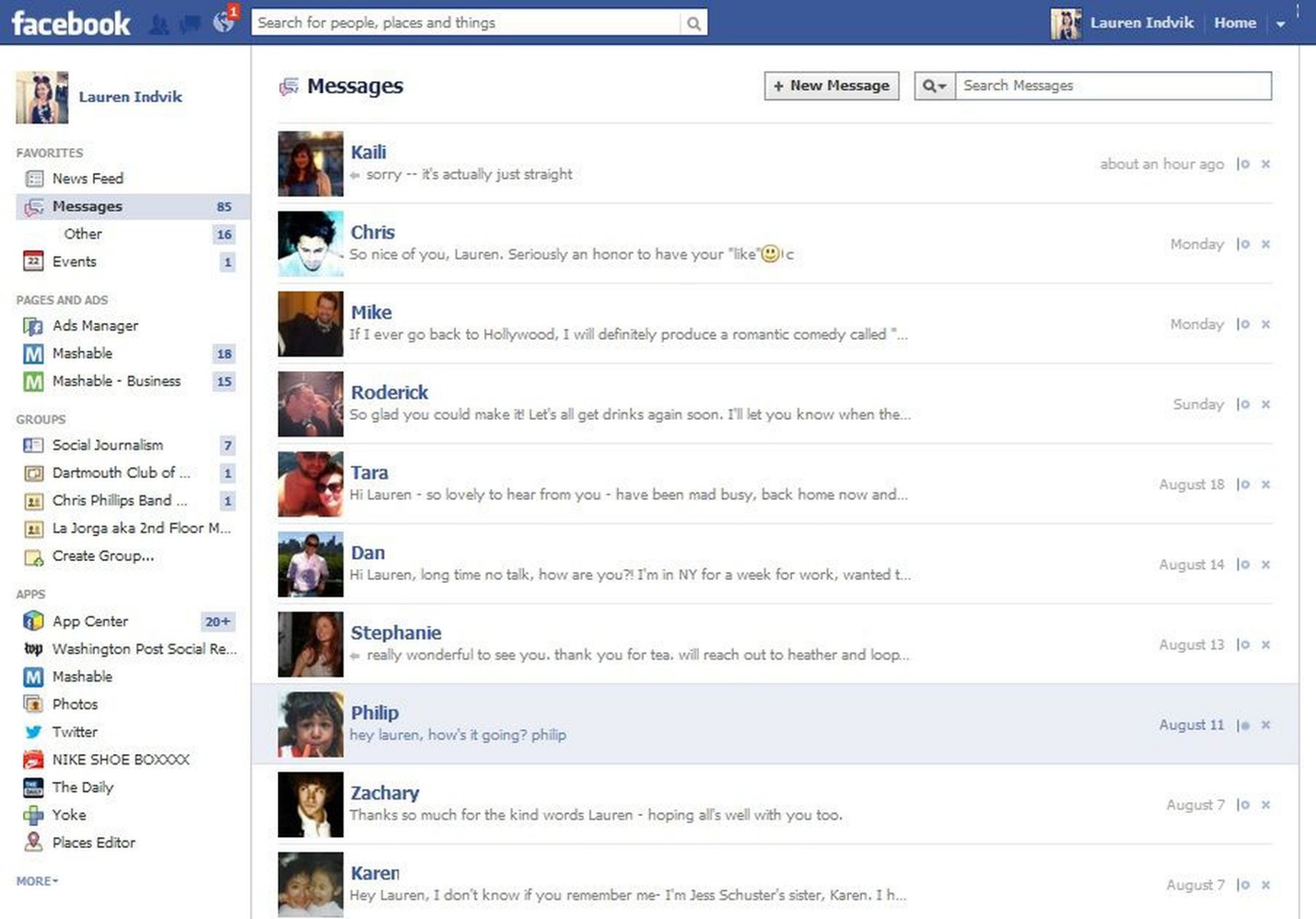
Изначально «The Facebook» ориентировался исключительно на Гарвардский университет. За первый месяц существования социальной сети в ней зарегистрировалось около половины студентов Гарварда.  
  Но уже в марте 2004 года «The Facebook» вышел за стены Гарварда и подключил к своей сети студентов Стэнфорда, Колумбийского университета и даже Йеля. Марк Цукерберг нацелился на университеты так называемой «[Лиги Плюща](https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D0%B3%D0%B0_%D0%BF%D0%BB%D1%8E%D1%89%D0%B0)», в которую входили все элитные учебные заведения. Затем к «Фейсбуку» постепенно было подключено большинство крупнейших учебных заведений США и Канады.  
  


Рис.2 Интерфейс социальной сети " The Facebook ".

## Перечень задач подлежащих решению в процессе разработки

Задачи, подлежащие решению в процессе разработки, следующие:

1) соблюдение правильности отправки сообщений без потерь данных;

2) программа должна иметь простой, но в то же время понятный и наглядный интерфейс, который не должен перегружать ресурсы компьютера;

3) программа должна иметь возможность выбора никнейма пользователя;

5) программа не должна занимать большой объем памяти и не должна требовать установки на жесткий диск компьютера;

7) работоспособность приложения в среде Windows;

## Обоснование выбора инструментов и платформы для разработки

Microsoft Windows (англ. windows — окна) — семейство операционных систем корпорации Майкрософт (Microsoft), ориентированных на применение графического интерфейса при управлении. Изначально были представлены многофункциональными надстройками для MS-DOS.

В настоящее время Windows является одной из самых распространенных операционных систем в мире. По официальному сообщению компании Microsoft, к середине 2020 года ОС Windows 10 теперь установлена на один млрд персональных компьютеров, согласно информации компании о количестве активных устройств (ноутбуков, ПК и планшетов) с этой ОС.

По состоянию на апрель 2020 года, операционная система Windows заняла второе место в общей мировой статистике использования ОС. Ее распространение составило 32,31% от количества всех операционных систем, существующих в настоящее время.

По статистике ОС для компьютеров операционная система Windows заняла первое место. Ее распространение составило 76,58% от количества всех ОС.

Общий рейтинг операционных систем в России, включая десктопы, мобильные, планшеты и игровые приставки, показывает, что лидером также является Windows, который установлен на 55,58% устройств. Операционные системы, установленные на компьютерах в России также в большей мере на Windows — 83,83%.

Стиль программирования Windows-приложений принципиально отличается от того, который сложился в операционных системах раннего поколения.

В MS-DOS программа монопольно владеет всеми ресурсами системы и является инициатором взаимодействия с операционной системой.

Совсем иначе дело обстоит в операционной системе Windows, которая строилась как многозадачная, и именно операционная система является инициатором обращения к программе. Все ресурсы Windows являются разделяемыми, и программа (приложение), не может владеть ими монопольно. В связи с такой идеологией построения операционной системы приложение должно ждать посылки сообщения операционной системы и лишь после его получения выполнить определенные действия, затем вновь перейти в режим ожидания очередного сообщения

Windows генерирует множество различных сообщений, которые направляются приложению, например, щелчок кнопки мыши или нажатие клавиши на клавиатуре. Если приложение не обрабатывает какие-то сообщения, реакция на них осуществляется операционной системой стандартным способом, так что задачей программиста является обработка лишь тех сообщений, которые необходимы приложению.

Разработчиками операционной системы Windows была создана библиотека функций, при помощи которых и происходит взаимодействие приложения с операционной системой, так называемые функции Программного интерфейса приложений (Application Program Interface, API).

Библиотека API-функций разрабатывалась в расчете на то, что ее можно использовать для любого языка программирования.

Вся идеология построения Windows-приложения ориентирована на взаимодействие с пользователем.

WinAPI спроектирован для использования в языке С++ для написания прикладных программ, предназначенных для работы под управлением операционной системы MS Windows.

Для выполнения курсовой работы был выбран язык программирования С++.

Язык С++ является одним из самых распространенных языков.

C++ — компилируемый, статически типизированный язык программирования общего назначения. Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование.

С++ — это язык программирования общего назначения, хорошо известный своей эффективностью, экономичностью, и переносимостью. Указанные преимущества С++ обеспечивают хорошее качество разработки почти любого вида программного продукта. Использование С++ в качестве инструментального языка позволяет получать быстрые и компактные программы.

С++ сочетает эффективность и мощность в относительно малом по размеру языке. Хотя С++ не содержит встроенных компонент языка, выполняющих ввод-вывод, распределение памяти, манипуляций с экраном или управление процессами, тем не менее, системное окружение С++ располагает библиотекой объектных модулей, в которой реализованы подобные функции. Библиотека поддерживает многие из функций, которые требуются.

Это решение позволяет изолировать языковые особенности от специфики процессора, на котором выполняется результирующая программа. Строгое определение языка делает его независимым от любых деталей операционной системы.

C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр.

# ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ

## Разработка структуры приложения

Для разработки приложения использовалась среда Microsoft Visual Studio, включающая в себя широкий набор функциональных инструментов.

Ввод данных в приложении осуществляется следующими способами:

1. через графический интерфейс с использованием клавиатуры. При этом вводимые данные могут быть на любов языке с использованием чисел и системных символов;

Посте входа в приложение пользователь должен ввести свой псевдоним или имя в соответствующее текстовое поле, после чего может начать общение в чате.

После введения текстого сообщения в соответствующее нажимается кнопка отправки сообщения после чего сообщение отправляется через сеть Интернет.

Общее назначение программного средства -общение через сеть интернет по средствам отправки текстовых сообщений

## Используемые функции Win32API

* **SetWindowText** - изменяет текст органа управления или окна где:
  + - HWND – главное окно приложения;
    - “Недопустимый аргумент” - текст который заменяет исходный;
* **CreateWindow** (

LPCTSTR lpClassName,   // указатель на зарегистрированное имя класса  
LPCTSTR lpWindowName,  // указатель на имя окна  
DWORD dwStyle,         // стиль окна  
int x,                 // горизонтальная позиция окна  
int y,                 // вертикальная позиция окна  
int nWidth,            // ширина окна  
int nHeight,           // высота окна  
HWND hWndParent,       // дескриптор родительского или окна владельца  
HMENU hMenu,           // дескриптор меню или ID дочернего окна  
HANDLE hInstance,      // дескриптор экземпляра приложения  
LPVOID lpParam         // указатель на данные создания окна) – создание окна сообщения где:)

Функция CreateWindow создает перекрывающее, выскакивающее или дочернее окно. Она определяет класс, заголовок, стиль окна и (необязательно) начальную позицию и размер окна. Функция также определяет и окно родителя или владельца, если таковые имеются и меню окна.

* **GetMessage** (BOOL GetMessage(

 LPMSG lpMsg,

HWND hWnd,

UINT wMsgFilterMin,

 UINT wMsgFilterMax)

Функция GetMessage извлекает сообщение из очереди сообщений вызывающего потока и помещает его в заданную структуру. Эта функция  регулирует поступление отправленных сообщений до тех пор, пока помещенное в очередь сообщение доступно для извлечения.

* **DispatchMessage**

LRESULT DispatchMessage(        
    const MSG\* lpmsg  
);

Функция DispatchMessage распределяет сообщение оконной процедуре. Обычно она  используется, чтобы доставить сообщение, извлеченное функцией [GetMessage](http://www.vsokovikov.narod.ru/New_MSDN_API/Message_queue/fn_getmessage.htm).

* **UpdateWindow** BOOL UpdateWindow(

HWND hWnd // дескриптор окна

);

Функция UpdateWindow обновляет рабочую область заданного окна, отправляя сообщение [WM\_PAINT](http://www.vsokovikov.narod.ru/New_MSDN_API/Paint_draw/msg_wm_paint.htm) окну, если регион обновления окна не пуст. Функция отправляет сообщение WM\_PAINT непосредственно оконной процедуре указанного окна, обходя очередь приложения. Если регион обновления пуст, никакое сообщение не отправляется.

* **ShowWindow** BOOL ShowWindow(

HWND hWnd, // дескриптор окна

int nCmdShow // состояние показа окна

);

Функция ShowWindow устанавливает состояние показа определяемого окна.

* **EndPaint** BOOL EndPaint(

HWND hWnd, // дескриптор окна

CONST PAINTSTRUCT \*lpPaint // данные об окраске);

Функция EndPaint отмечает конец окрашивания в заданном окне.

* **WSAStartup** (int WSAStartup(

WORD wVersionRequired,

LPWSADATA lpWSAData

)

Функция WSAStartup инициирует использование Winsock DLL процессом.

* **WSAGetLastError** (int WSAGetLastError();))

Функция WSAGetLastError возвращает статус ошибки для последней неудачной операции с сокетами Windows.

* **inet\_addr** (unsigned long inet\_addr(

const char \*cp

);)

Функция inet\_addr преобразует строку, содержащую десятичный адрес IPv4 с точками, в правильный адрес для структуры [IN\_ADDR](https://docs.microsoft.com/en-us/windows/desktop/api/winsock2/ns-winsock2-in_addr) .

* **Htons**

(u\_short htons(

u\_short hostshort

);)

Функция htons преобразует u\_short из хоста в сетевой порядок байтов TCP / IP (который является прямым порядком байтов).

* **Bind**

(int bind(

SOCKET s,

const sockaddr \*addr,

int namelen

))

Функция привязки связывает локальный адрес с сокетом.

* **WSAAsyncSelect**

(int WSAAsyncSelect(

SOCKET s,

HWND hWnd,

u\_int wMsg,

long lEvent

))

Функция WSAAsyncSelect запрашивает уведомление Windows о сетевых событиях на основе сообщений для сокета.

* **Listen**

int WSAAPI listen(

SOCKET s,

int backlog

);

Функция прослушивания переводит сокет в состояние, в котором он ожидает входящего соединения.

* **Accept**

SOCKET WSAAPI accept(

SOCKET s,

sockaddr \*addr,

int \*addrlen

);

Функция accept разрешает попытку входящего подключения к сокету.

* **Gethostbyaddr**

hostent \*WSAAPI gethostbyaddr(

const char \*addr,

int len,

int type

);

Функция gethostbyaddr извлекает информацию о хосте, соответствующую сетевому адресу.

* **Send**

int WSAAPI send(

SOCKET s,

const char \*buf,

int len,

int flags

);

Функция отправки отправляет данные на подключенный сокет.

* **Recv**

int recv(

SOCKET s,

char \*buf,

int len,

int flags

);

Функция recv получает данные из подключенного сокета или привязанного сокета без установления соединения.

## Разработка архитектуры приложения

Приложение разрабатывалась по архитектуре многоуровневого шаблона.

Данное приложение использует многоуровневый архитектурный шаблон. Данный шаблон используется для структурирования программ, которые можно разложить на группы неких подзадач, находящихся на определенных уровнях абстракции. Каждый слой представляет службы для следующего, более высокого слоя. На рисунке 2 представлена схема приложения.

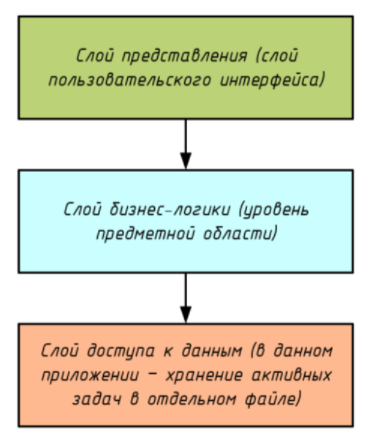


Рис. 2. Схема построения приложения по многоуровневому шаблону.

Слой представления в разрабатываемом приложении представлен пользовательским интерфейсом с полями для ввода информации (имя пользователя и текст отправляемого сообщения) и кнопкой для отправки

Слой бизнес-логики описан основным исполняемым файлом sourse.cpp и Исполняемый файл sourse.cpp состоит из функций обработки нажатия клавиш и логики работы сокетов для отправки сообщений.

## Разработка и реализация алгоритмов приложения

Для создания программы необходимо реализовать алгоритмы, позволяющие отправлять сообщения по средствам интернет а также принимать их на сервере и обрабатываеть.

Блок-схемы алгоритмов представлены на рисунках 3-8.



Рис. 3. Функция работы сервера с клиентами



Рис. 4. Блок-схема работы WM\_ACCEPT .



Рис. 5. Блок-схема функции принимающий сообщения на клиенте.



Рис. 6. Блок-схема функции запуска сервера.

## Разработка интерфейса взаимодействия пользователя с системой

Главное окно программы клиента [(Приложение А)](#приложение1) должно состоять из кнопки для отправки сообщения и поля для просмотра ранее отправленых сообщений (Рисунок 7).



Рисунок 10 – Макет главного окна приложения

Главное окно программы сервера [(Приложение А)](#приложение1) должно состоять из поля для просмотра лога сервера(Рисунок 8).



Рисунок 8 – Макет главного окна приложения

При нажати на кнопку отправки сообщения, оно отправится на другие клиенты и появился в окне для просмотра сообщений.

# ПРОЕКТНО-ТЕХНОЛОГИЧИСКАЯ ЧАСТЬ

## Разработка руководства пользователя

Запуск программы осуществляется по открытию файла Client.exe

После запуска открывается главное окно приложения (Рисунок 9 .)

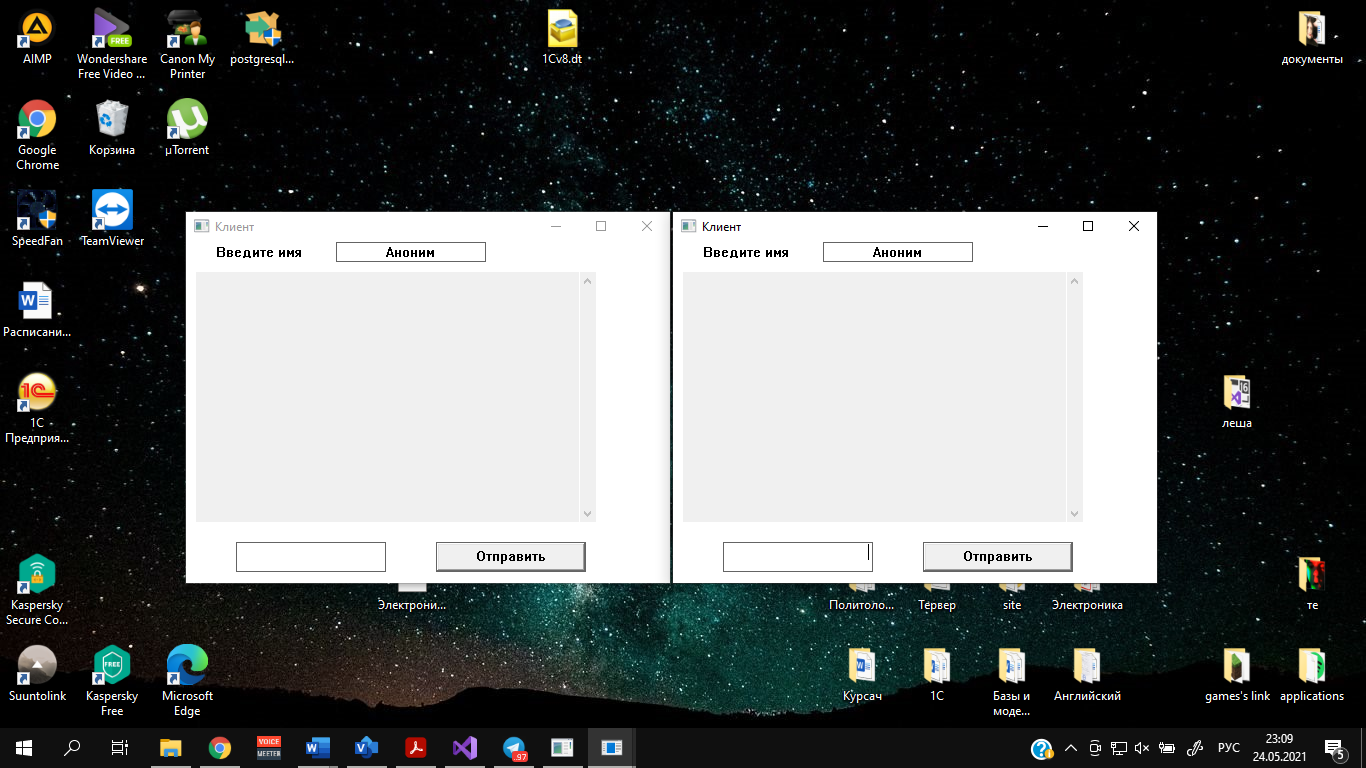


Рисунок 12 – Главное окно приложения

После входа в приложение требуется ввести имя в соответствующее поле и сообщение в поле внизу, а далее нажать на кнопку «Отправить»

Пример работы приложения на двух клиентах(Рисунок 10):

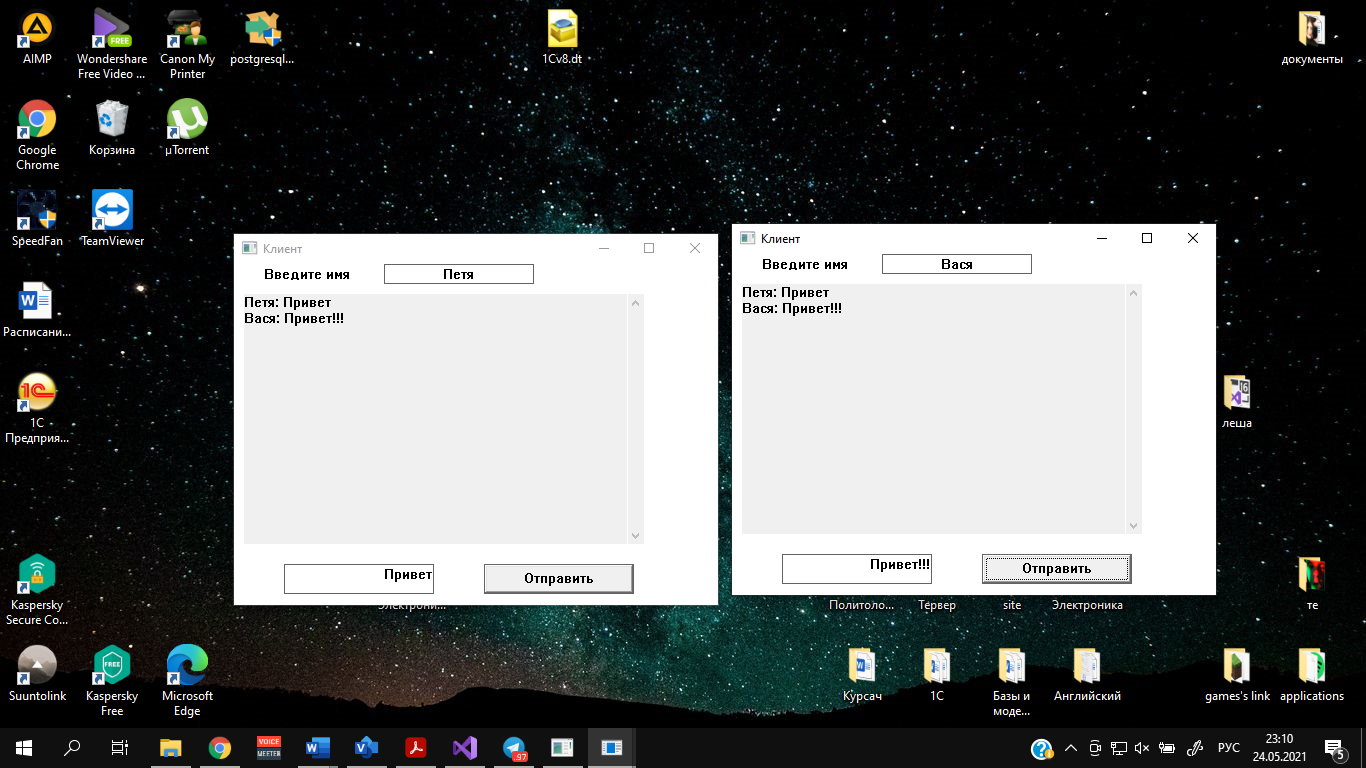


Рисунок 12 – Работа на двух клиентах

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной курсовой работы был разработан централизованный чат на основе WinAPI.

Работа выполнялась в несколько этапов: была разработана структура системы и реализовано прикладное оконное приложение на основе WinAPI.

Были сформированы навыки по разработке и реализации программного приложения с использованием интерфейса прикладного программирования (АРI) операционных систем.

В данной работе выполнены все поставленные задачи.

Спроектированное приложение подойдет для задач, в ходе которых необходимо запустить файл в заданное пользователем время.

В будущем можно усовершенствовать разработанное приложение путем добавления дополнительного функционала (например добавление дополнительных переписок с чатами).

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТУРАТУРЫ

1. <https://docs.microsoft.com/ru-ru/windows/win32/taskschd/task-scheduler-start-page>

2. <http://window.edu.ru/catalog/pdf2txt/389/67389/40509>

# ПРИЛОЖЕНИЕ А

Создание главного окна:

hInst = ((LPCREATESTRUCT)lparam)->hInstance; // дескриптор приложения

// Создаем и показываем первое поле редактирования

hStat = CreateWindow(L"edit", L"", WS\_CHILD | WS\_VISIBLE | WS\_VSCROLL | ES\_MULTILINE | ES\_READONLY,

0, 40, 400, 250, hwnd, (HMENU)MAIN\_STATICBOX, hInst, NULL);

// Создаем и показываем поле текста для результата

hStat = CreateWindow(L"static", L"", WS\_CHILD | WS\_VISIBLE,

150, 150, 120, 20, hwnd, 0, hInst, NULL);

ShowWindow(hStat, SW\_SHOWNORMAL);

hBtn = CreateWindow(L"button", L"Запустить сервер",

WS\_CHILD | WS\_VISIBLE | WS\_BORDER,

100, 100, 150, 30, hwnd, 0, hInst, NULL);

ShowWindow(hBtn, SW\_SHOWNORMAL);

break;

}

# ПРИЛОЖЕНИЕ Б

Создание окна клиента

hInst = ((LPCREATESTRUCT)lparam)->hInstance; // дескриптор приложения

// Создаем и показываем первое поле редактирования

hEdt1 = CreateWindow(L"edit", L"80",

WS\_CHILD | WS\_VISIBLE | WS\_BORDER | ES\_RIGHT, 50, 300, 150, 30,

hwnd, 0, hInst, NULL);

ShowWindow(hEdt1, SW\_SHOWNORMAL);

// Создаем и показываем второе поле редактирования

hEdt2 = CreateWindow(L"edit", L"Аноним",

WS\_CHILD | WS\_VISIBLE | WS\_BORDER | ES\_CENTER ,150, 0, 150,

20, hwnd, 0, hInst, NULL);

ShowWindow(hEdt2, SW\_SHOWNORMAL);

// Создаем и показываем кнопку

hBtn = CreateWindow(L"button", L"Отправить",

WS\_CHILD | WS\_VISIBLE | WS\_BORDER,

250, 300, 150, 30, hwnd, 0, hInst, NULL);

ShowWindow(hBtn, SW\_SHOWNORMAL);

// Создаем и показываем поле текста для результата

hStat = CreateWindow(L"edit", L"", WS\_CHILD | WS\_VISIBLE | WS\_VSCROLL | ES\_MULTILINE | ES\_READONLY ,

10, 30, 400,250, hwnd, (HMENU)MAIN\_STATICBOX, hInst, NULL);

ShowWindow(hStat, SW\_SHOWNORMAL);

break;