

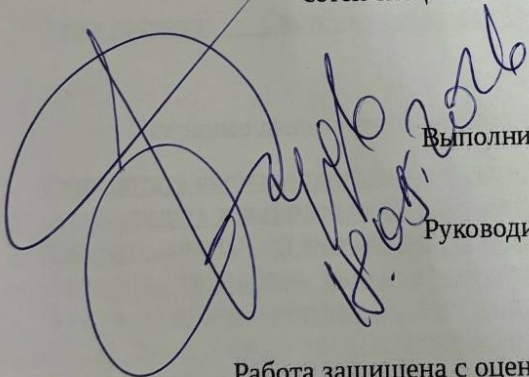
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Курсовая работа
по дисциплине «Системное программирование»
на тему «Разработка системы программ обмена сообщениями»

ПГУ 1.09.05.01

Специальность подготовки: 09.05.01 «Применение и эксплуатация
автоматизированных систем специального назначения»
Специализация «Эксплуатация вычислительных машин, комплексов, систем и
сетей специального назначения»

Выполнил: студент гр. 22ВОЭ1

 Николаев Д.А.

Руководитель: к.т.н., доцент каф. ВТ

Егоров В.Ю.

Работа защищена с оценкой _____

Преподаватели _____

Дата защиты _____

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет Вычислительной техники

Кафедра "Вычислительная техника"

"УТВЕРЖДАЮ"

Зав. кафедрой ВТ

профессор М. А. Митрохин

« ____ » _____ 20 __ г

ЗАДАНИЕ

на курсовое проектирование по курсу

Системное программирование

Студенту Николаеву Дмитрию Александровичу Группа 22ВОЭ1

Тема проекта: Система программ обмена сообщениями

Исходные данные (технические требования) на проектирование

Разработать систему программ клиент-серверной архитектуры, позволяющую осуществлять взаимодействие пользователей на основе обмена сообщениями в рамках локальной вычислительной сети. Взаимодействие программ осуществляется на основе вновь разрабатываемого протокола взаимодействия. Протокол взаимодействия должен включать в себя команды управления, команды пререправки данных, информационные команды. Может быть представлен и состоять из серверной и клиентской частей. Клиентская программа включает в себя графический интерфейс пользователя и реализацию протокола взаимодействия. Серверная программа может вообще не иметь интерфейса, либо иметь консольный интерфейс реализующий серверную часть протокола взаимодействия. Необходимо предусмотреть передачу текстовых сообщений (личные и общие), файлов. Опционально идентификация, аутентификация пользователя, обмен мультимедиа информацией.

Язык программирования – C, C++, C#

Среда программирования – Visual Studio

Операционная система – любая

Объем работы по курсу

1. Расчетная часть

Разработка протоколов взаимодействия программ
Разработка и покомпонентная отладка программ
Комплексная отладка системы

2. Графическая часть

Диаграммы языка UML:

- 1) диаграмма классов;
- 2) диаграмма деятельности;
- 3) диаграмма вариантов использования;
- 4) диаграмма развёртывания;
- 5) диаграмма последовательности.

3. Экспериментальная часть

Проведение тестирования на институтской локальной вычислительной сети

Срок выполнения проекта по разделам

В соответствии с графиком выполнения работы

Дата выдачи задания "19" сентября 2025г.

Дата защиты проекта "___" _____

Руководитель Егоров В.Ю. В.Ю.

Задание получил "19" сентября 2025г.

Студент Иванов Д.А.

Реферат

Пояснительная записка состоит из 92 листов, 30 рисунков и 5 источников.

Цель работы – закрепление знаний и приобретение практических навыков разработки системы программ клиент-серверной архитектуры, позволяющей осуществлять взаимодействие пользователей на основе обмена сообщениями в рамках локальной вычислительной сети.

Курсовая работа направлена на разработку системы программ клиент-серверной архитектуры, позволяющей осуществлять взаимодействие пользователей на основе обмена сообщениями в рамках локальной вычислительной сети.

Объект исследования для данной курсовой работы – это система программ клиент-серверной архитектуры, позволяющей осуществлять взаимодействие пользователей на основе обмена сообщениями в рамках локальной вычислительной сети

В результате курсовой работы будет разработано приложение "Клиент-серверный чат".

					ПГУ 1.09.05.01			
Изм.	Лист	№ докум.	Подпись	Дата	«Разработка системы программ обмена сообщениями»	Лит.	Лист.	Листов
Разраб.		Николаев Д.А.					3	92
Провер.		Егоров В.Ю.						
					Пояснительная записка	Гр. 22ВОЭ1		
Н.Контр.								
Утверд.		Митрохин М.А.						

Содержание

Введение.....	6
1 Постановка задачи.....	8
2 Выбор решения.....	9
3 Описание программы.....	10
4 Описание организации сетевого взаимодействия приложения.....	13
5 Описание способа организации пользовательского интерфейса.....	16
6 Описание результатов работы программы.....	24
Заключение.....	29
Список используемых источников.....	31
Приложение А. Листинг программы.....	32
Приложение Б. UML-диаграммы.....	86

Введение

Современный цифровой ландшафт характеризуется стремительным развитием коммуникационных технологий, где системы обмена сообщениями стали критически важной инфраструктурой как для повседневного взаимодействия, так и для профессиональной деятельности. В условиях растущих требований к производительности, надежности и функциональности таких систем, особую актуальность приобретает использование языков программирования, обеспечивающих прямой контроль над аппаратными ресурсами и сетевыми взаимодействиями. Язык C++, благодаря своей философии нулевых накладных расходов, детерминированному управлению памятью и возможностям низкоуровневого программирования, утвердил себя как фундаментальный инструмент для создания высокопроизводительных сетевых приложений. Его стандартизация (ISO), поддержка множества платформ и возможность сочетать низкоуровневые операции с высокоуровневыми абстракциями делают C++ предпочтительным выбором для разработки систем, где критически важны скорость обработки данных, минимальная задержка передачи и эффективное использование ресурсов.

Актуальность темы обусловлена необходимостью создания не только функциональных, но и оптимизированных, масштабируемых клиент-серверных решений, способных обеспечивать мгновенный обмен сообщениями и файлами в реальном времени с поддержкой множества одновременных подключений. Разработка системы обмена сообщениями на C++ позволяет глубоко изучить основы сетевых коммуникаций на уровне сокетов операционной системы, модели управления параллельными потоками выполнения, принципы проектирования прикладных протоколов и механизмы обеспечения целостности передаваемых данных. Многомодульная архитектура системы, предполагающая четкое разделение на независимые компоненты (серверное ядро, клиентский графический интерфейс, протокол передачи), способствует оптимизации процесса разработки: повышает читаемость и сопровождаемость кода, упрощает тестирование отдельных компонентов и обеспечивает возможность их повторного использования в проектах.

Целью данной курсовой работы является разработка и практическая реализация клиент-серверной системы обмена сообщениями с графическим интерфейсом пользователя на языке C++. Достижение данной цели позволит не только освоить ключевые аспекты сетевого и многопоточного программирования, но и получить комплексный опыт проектирования архитектуры распределенного приложения, обработки бинарных протоколов и создания интуитивного пользовательского интерфейса для платформы Windows.

Объектом исследования выступает процесс разработки клиент-серверных систем обмена данными в реальном времени, а предметом — методы и инструменты, применяемые для создания таких систем на C++ с использованием низкоуровневых сетевых API и объектно-ориентированного подхода.

В рамках курсовой работы применялись следующие методы:

1. анализ специализированной литературы и документации по сетевому программированию (Winsock API) и многопоточности;
2. проектирование структуры системы и формата передаваемых пакетов;
3. практическая реализация и поэтапное тестирование функциональных модулей.

В рамках проекта будет разработано приложение, включающее три основных функциональных модуля: серверный модуль для управления подключениями и маршрутизации данных, клиентский модуль с графическим интерфейсом для взаимодействия с пользователем, и модуль коммуникационного протокола, обеспечивающий надежную передачу текстовых сообщений и файлов. Такое разделение позволит продемонстрировать преимущества модульной архитектуры, упрощающей разработку, отладку и возможное расширение системы.

Таким образом, данная курсовая работа направлена не только на теоретическое изучение принципов построения распределенных систем, но и на получение практических навыков создания полнофункционального клиент-серверного приложения, сочетающего высокую производительность сетевого ядра с удобным пользовательским интерфейсом.

1 Постановка задачи

Основная задача проекта заключается в создании системы программ клиент-серверной архитектуры, позволяющую осуществлять взаимодействие пользователей на основе обмена сообщениями в рамках локальной вычислительной сети. Взаимодействие программ осуществляется на основе вновь разрабатываемого протокола взаимодействия. Протокол взаимодействия должен включать в себя команды управления, команды передачи данных, информационные команды. Может быть представлен в текстовом или бинарном виде и состоять из серверной и клиентской частей.

Клиентская программа включает в себя графический интерфейс пользователя и протокол взаимодействия. Серверная программа может вообще не иметь интерфейса. Необходимо предусмотреть передачу текстовых сообщений, файлов.

Средой разработки была выбрана Visual Studio — свободная интегрированная среда разработки приложений (IDE) на языках программирования C, C++, C#, которая будет использоваться для написания, компиляции и отладки кода.

Список использованных источников

1. Руководство по созданию приложений WinForm — C++ [Электронный ресурс]. — Режим доступа: <https://learn.microsoft.com/ru-ru/visualstudio/ide/create-a-visual-basic-winform-in-visual-studio?view=visualstudio> (дата обращения: 01.11.2025).
2. Создание Mutex C++ [Электронный ресурс]. — Режим доступа: https://www.reddit.com/r/C_Programming/comments/135udh5/how_do_you_implement_a_mutex/?tl=ru (дата обращения: 06.11.2025).
3. Разработка своего сетевого протокола C++ [Электронный ресурс]. — Режим доступа: <https://programmersforum.ru/showthread.php?t=302051> (дата обращения: 10.11.2025).
4. Microsoft Docs. Winsock (Windows Sockets) [Электронный ресурс]. — Режим доступа: <https://docs.microsoft.com/en-us/windows/win32/winsock/windows-sockets-start-page-2> (дата обращения: 12.11.2025).
5. TheForger's WinAPI Programming Tutorial [Электронный ресурс]. — Режим доступа: <https://winprog.org/tutorial/start.html> (дата обращения: 13.11.2025).

Заключение

В ходе выполнения курсовой работы была успешно разработана и реализована клиент-серверная система обмена сообщениями и файлами «Клиент-серверный», предназначенная для взаимодействия пользователей в рамках локальной вычислительной сети. Поставленная цель — закрепление знаний и приобретение практических навыков в области сетевого и многопоточного программирования, проектирования архитектуры распределённых приложений и создания графических интерфейсов — достигнута в полном объёме.

Разработано клиент-серверное приложение на языке C++, состоящее из двух модулей: серверной части, управляющей подключениями, регистрацией пользователей по уникальному имени, ретрансляцией публичных и частных сообщений, пересылкой файлов с подтверждением, а также поддерживающей многопоточную обработку клиентов, и клиентской части с интуитивно понятным графическим интерфейсом, реализованным на базе Win32 API, обеспечивающим удобное взаимодействие с пользователем.

Реализован бинарный протокол поверх TCP с использованием упакованных структур, поддерживающий регистрацию пользователей (HelloPacket), отправку публичных и частных сообщений (MessagePacket с полем pub_priv), передачу файлов с разбиением на чанки по 64 КБ (FileStartPacket, FileChunkPacket, FileEndPacket), динамическое обновление списка онлайн-пользователей (UserListPacket), а также обработку ошибок и уведомлений в реальном времени.

Обеспечена модульность архитектуры, что упрощает сопровождение, тестирование и возможное расширение функциональности системы. Критические секции гарантируют потокобезопасный доступ к разделяемым данным, а динамические буферы предотвращают переполнение стека при передаче больших файлов.

Проведено тестирование работы системы в локальной сети, подтвердившее корректность взаимодействия клиентов и сервера, стабильность передачи файлов с автоматическим созданием уникальных имён при конфликтах и отказоустойчивость при разрывах соединения.

В процессе работы были изучены и применены ключевые технологии и инструменты: Winsock2 API для сетевого взаимодействия, многопоточное программирование (CreateThread, WaitForSingleObject) для обработки одновременных подключений, Win32 API для создания графического интерфейса (CreateWindowExW, WM_* сообщения), критических секций для синхронизации потоков, а также принципы проектирования пользовательских интерфейсов и гибридных сетевых протоколов.

Таким образом, разработанная система является законченным, функциональным решением, демонстрирующим возможность создания производительных и масштабируемых клиент-серверных приложений на C++. Работа подтвердила теоретические знания, полученные в рамках дисциплины «Системное программирование», и предоставила ценный практический опыт, который может быть использован в дальнейшем для решения более сложных задач в области сетевых технологий и распределённых систем.

