

Министерство науки и образования Российской Федерации

Пензенский государственный университет

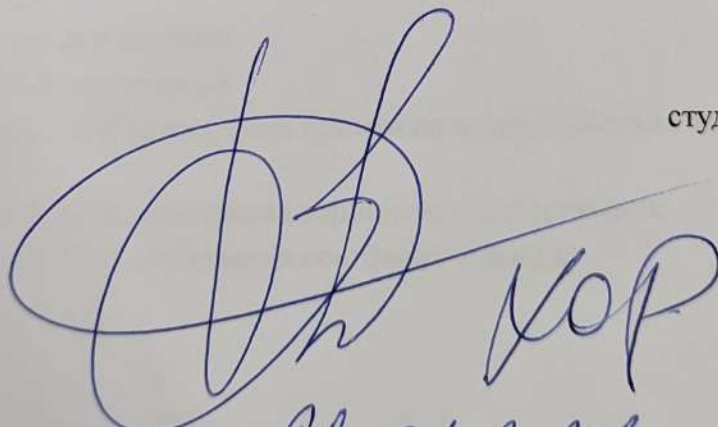
Кафедра "Вычислительная техника"

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

по курсу "Основы операционных систем"

на тему "Система программ обмена текстовыми сообщениями"



26.01.2024 г.

Выполнил:

студент группы 21ВВП1

Сорокина Е.А.

Принял:

Егоров В.Ю.

Пенза 2023

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

Студенту Сорокин Е.А. Группа 21ВВП1
Тема проекта Система программ обмена
текстовыми 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 может быть представлен как текстовый протокол, как бинарный протокол, и реализация протокола должна предусматривать клиентскую и серверную часть в виде конечных автоматов. Клиентская часть содержит графический пользовательский интерфейс, сервер реализуется в виде консольного интерфейса. Необходимо предусмотреть передачу индивидуального сообщения между пользователями и широковещательного сообщения. Опционально реализует следующий функционал:

- 1) аутентификация пользователя
- 2) обмен мультимедийной информацией

Операционная система: Любая.

Языки программирования: С, С++, С#.

Объем работы по курсу

1. Расчетная часть

- 1) Разработка протокола взаимодействия
- 2) Разработка и покомпонентная отладка программ
- 3) Комплексная отладка системы программ

2. Графическая часть

Не менее 5 UML диаграмм, иллюстрирующих работу системы

3. Экспериментальная часть

Отсутствует

Срок выполнения проекта по разделам

- 1 В соответствии с графиком
- 2
- 3
- 4
- 5
- 6
- 7
- 8

Дата выдачи задания " 5 " октября 23г.

Дата защиты проекта " "

Руководитель *Вз*

Задание получил

" 5 " октября

2023г.

Студент

Сорокина Елена Андреевна Сорок

Содержание

Содержание	3
Введение	5
1. Постановка задачи	6
2. Описание алгоритма работы	7
3. Описание протокола передачи данных	9
4. Описание программы	9
4.1 Сервер	10
4.2 Клиент	11
5. Описание работы программы	16
Заключение	17
Список используемых источников	17
Приложение А.	18
Листинг программы	18
Приложение А.1.	
Файл «Program.cs» для Сервера	18
Приложение А.2.	
Файл «Server.cs» для Сервера	19
Приложение А.3.	
Файл «ClientObject.cs» для Сервера	20
Приложение А.4.	
Файл «Program.cs» для Клиента	25
Приложение А.5.	
Файл «Form1.cs» для Клиента	26
Приложение Б. UML-диаграммы	32
Приложение Б.1. UML-диаграмма протокола взаимодействия клиента и сервера	32
Приложение Б.2. UML-диаграмма вариантов использования	33
Приложение Б.3. UML-диаграмма состояний сервера	34

Введение

В наши дни возросли потребности в общении и мгновенной доставки сообщений адресату, а вместе с ним увеличилось и количество предоставляемых услуг. Современные мессенджеры позволяют не только обмениваться текстовыми сообщениями через интернет, но и передавать файлы с одного компьютера на другой и даже выражать эмоции с помощью смайликов. Да и сами смайлики существенно изменились за прошедшие годы, теперь это качественные анимированные картинки, или даже отрывки видеороликов, которые позволяет использовать для выражения эмоций. Функция группового общения позволяет общаться, обсуждать задачи коллективам людей, работающих над одной задачей в разных концах земного шара. Сейчас начали появляться целые голосовые комнаты в мессенджерах для семейных бесед или деловых встреч.

Клиент-сервер (Client-server) – вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг (сервисов), называемыми серверами, и заказчиками услуг, называемыми клиентами. Нередко клиенты и серверы взаимодействуют через компьютерную сеть и могут быть как различными физическими устройствами, так и программным обеспечением. Сама технология очень проста. Например, пользователь делает запрос (в google ищет информацию), а сервер выдает ответ (в виде списка сайтов по этой информации).

Целью курсового проектирования является разработка системы распределённого доступа к текстовому документу, состоящей из сервера и клиентов, которые взаимодействуют между собой по сети.

1. Постановка задачи

В курсовой работе необходимо разработать систему программ клиент-серверной архитектуры, позволяющей осуществлять параллельную передачу сообщений между клиентами посредством работы с TCP/IP протоколами.

Необходимо создать сервер и клиент. Клиенту обрабатывать команды, поступающие от сервера. Сервер в этот момент синхронизирует клиентов, собирает и обрабатывает данные и выдает их клиентам.

Функции клиента:

- Интерфейс пользователя
- Взаимодействие с сервером

Функции сервера:

- Синхронизация клиентских приложений
- Отслеживание подключений и отключений клиентов
- Рассылка сообщений клиентам

Операционная система – Windows.

Язык программирования – C/C#.

Среда программирования: Microsoft Visual Studio 2022.

2. Описание алгоритма работы

Разработанная программа основывается на TCP-протоколе, состоящая из клиента и сервера.

Сначала запускаем сервер, для прослушивания входящих соединений по порту 8888 на IP-адресе 192.168.56.1. Устанавливается соединение между сигналом сервера и слотом, который вызывается при подключении нового клиента. При отключении клиента вызывается функция и освобождается память, занятая сокетом клиента.

В зависимости от содержания сообщения пришедшего на адрес сервера происходит различная обработка:

- Регистрация нового пользователя.
- Отправка сообщений.
- Обработка закрытия диалога и обновление списка пользователей.

3. Описание протокола передачи данных

Протокол - это набор соглашений, которые определяют обмен данными между сервером и клиентом. Протокол передачи данных, с помощью которого сервер и клиент обмениваются сообщениями, в данном коде основан на использовании TCP-сокетов на языке C#.

Клиент-серверное взаимодействие осуществляется с помощью протокола передачи данных через сокет. Сам протокол обмена данными между сервером и клиентом основан на передаче текстовых сообщений: клиент отправляет текстовый запрос на сервер, впоследствии сервер отвечает соответствующими текстовыми сообщениями.

Клиент подключается к серверу с использованием метода connect класса TcpClient.

Сервер слушает новые подключения с использованием метода start() класса TcpListener. Запускается функция, присваивающая каждому пользователю свой сокет.

Сервер принимает информацию о пользователе от клиента. Клиент отправляет текстовое сообщение, которое впоследствии декодируется в массив байтов.

Протокол обмена данными между клиентом и сервером включает следующие действия:

Клиент - сервер

1. Подключение

client.Connect <Имя пользователя>

2. Отправка сообщения

<Имя пользователя> <Сообщение>

3. Отключение

Disconnect <Имя пользователя>

Сервер - клиент

1. Оповещение пользователей о подключении нового клиента
<Имя пользователя> "connected!"
2. Отправка обновленного списка пользователей при подключении клиента
SendName() <Имена пользователей>
3. Отправка сообщения от клиента остальным пользователям
<Имя пользователя> <Сообщение>
4. Отключение клиента от сервера
killClient <Имя пользователя>
SendName() <Имена оставшихся пользователей>
5. Оповещение пользователей об отключении клиента
<Имя пользователя> "disconnected!"

Заключение

При выполнении данного курсового проекта были изучены принципы работы протоколов TCP/IP, разработка клиента и сервера. Организовали синхронную работу программы клиент - серверной архитектуры. Усвоили механизм обработки событий в формах и пользовательских инструментах.

В результате выполнения данной курсовой работы была разработана система программ клиент-серверной архитектуры, позволяющей осуществлять параллельную передачу сообщений между клиентами с помощью сервера посредством работы с TCP/IP протоколами.

Программа написана на языке C# с использованием среды программирования Visual Studio 2022.

Список используемых источников

1. Бертран М. Объектно-ориентированное конструирование программных систем / М. Бертран.-Москва: Русская Редакция, 2005 – 1204с.
2. Эндрюс Г.Р. Основы многопоточного, параллельного и распределенного программирования.: Пер. с англ.— М.: Издательский дом «Вильямс», 2003. — 512 с.
3. Шарп Джон – Microsoft Visual C#. Подробное руководство. 8-е изд. — СПб.: Питер, 2017. — 848 с.: ил. — (Серия «Библиотека программиста»).
4. METANIT.COM Сайт о программировании [Электронный ресурс]. Режим доступа: <http://www.cyberforum.ru/>, свободный.