|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

|  |  |
| --- | --- |
| ФАКУЛЬТЕТ | «Информатики и систем управления» |
| КАФЕДРА | «Защиты информации» |

**РАСЧЕТНО−ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОЙ РАБОТЕ***

***НА ТЕМУ:***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Разработка программы для децентрализованного*  *обмена текстовыми сообщениями*  *на языке программирования Си* | | | | | | |
| Студент | *ИУ10-21* |  |  |  |  | *А.А. Галактионов* |
|  | (Группа) |  |  | (Подпись, дата) |  | (И.О. Фамилия) |
|  |  |  |  |  |  |  |
| Руководитель курсового проекта | | | |  |  | *А.В. Астрахов* |
|  |  |  |  | (Подпись, дата) |  | (И.О. Фамилия) |
|  |  |  |  |  |  |  |
| Консультант | | | |  |  | *И.Р. Матакаев* |
|  |  |  |  | (Подпись, дата) |  | (И.О. Фамилия) |

*2021 г.*

**Министерство образования и науки Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой

(Индекс)

(И.О. Фамилия)

« \_\_\_\_ » 20 \_\_\_\_ г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине *Информатика*

Студент группы *ИУ10-21*

Галактионов *Андрей Андреевич*

(Фамилия, имя, отчество)

Тема курсовой работы *Разработка программы для децентрализованного обмена текстовыми сообщениями на языке программирования Си*

Направленность КР (учебная, исследовательская, практическая, производственная, др.)

*учебная*

Источник тематики (кафедра, предприятие, НИР) *кафедра*

График выполнения КР: 25% к *4*  нед., 50% к  *9*  нед., 75% к  *12*  нед., 100% к  *15*  нед.

***Задание:*** :*программа разрабатывается на языке программирования Си с использованием*

*сетевых сокетов и должна реализовать децентрализованную передачу и получение текстовых сообщений в локальной сети*

***Оформление курсовой работы:***

Расчетно-пояснительная записка на  *20*  листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

*текст программы на языке программирования Си*

Дата выдачи задания « *19* » *февраля* 20 *21* г.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Руководитель курсового проекта** |  |  |  | *А.В. Астрахов* |
|  |  | (Подпись, дата) |  | (И.О. Фамилия) |
|  |  |  |  |  |
| **Студент** |  |  |  | *А.А. Галактионов* |
|  |  | (Подпись, дата) |  | (И.О. Фамилия) |

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

目录

[1 Введение 4](#_Toc1411463558)

[2 Теоретическая часть 5](#_Toc1656486128)

[3 Практическая часть 7](#_Toc615073828)

[4 Заключение 11](#_Toc1818868837)

[5 Список использованных источников информации 12](#_Toc710887989)

[Приложение. Исходные тексты программ 13](#_Toc1914635591)

# Введение

Данная курсовая работа направлена на закрепление и обобщение знаний, которые были получены в ходе прохождения курса по языку программирования Си.

Целью курсовой работы является разработка программы для децентрализованного обмена сообщениями на языке программирования Си.

Задачи курсовой работы:

1. Повторить изученный материал по языку программирования Си.
2. Подробнее рассмотреть тему «использование сетевых сокетов для обмена сообщениями в локальной сети».
3. Разработать программу для децентрализованного обмена текстовыми сообщениями в локальной сети.
4. Создать UML–диаграмму.

# Теоретическая часть

Разработанная в ходе выполнения курсовой работы программа для децентрализованного обмена текстовыми сообщениями представляет собой клиент-серверное приложение, в котором и клиент и сервер совмещены.

Клиент-серверное приложение – это программа, работающая с сетью, в которой взаимодействуют «заказчики услуг» клиенты и «исполнитель услуг» сервер. Под этим подразумевается, что клиент заказывает «услугу», что означает отправку чего-либо определенному пользователю (клиенту), а сервер, получив заказ, исполняет его. Клиент и сервер обычно – разные программы

Объединение клиента и сервера означает, что одна программа и заказывает и исполняет «услугу».

Любая программа, работающая с сетью, подразумевает использование сетевых сокетов.

Сетевые сокеты или сокеты Беркли – название программного интерфейса для обеспечения обмена данными между процессами. Процессы при таком обмене могут исполняться как на одном устройстве, так и на различных, связанных между собой сетью. Сокет – абстрактный объект, представляющий конечную точку соединения.

Для работы с сетью были использованы broadcast адреса – специальные адреса, которые могут принимать все пользователи локальной сети.

Для реализации программы были использованы структуры данных.

Данные – это зарегистрированные сигналы. Они могут принимать вид битов, байтов и представлять собой целые числа, символы, числа с плавающей запятой и так далее.

Структура данных (структура) (struct) – это совокупность данных (различных или одинаковых), которые в памяти хранятся рядом друг с другом, что позволяет удобно к ним обращаться.

Указатель – это тип данных, который представляет собой адрес на ячейку памяти. Через операцию разыменования можно обратиться к данным, содержащимся там. Указатель, который указывает на «ничего» – NULL.

Односвязный список, который также используется в реализации, основан на структуре данных, но одним из элементов является указатель на такую же структуру, обычно имеет название next. Это позволяет хранить только одну структуру и переходить к другим через этот указатель.

Файлом называют именованный набор данных. Функции работы с файлами находятся в <stdio.h>. Работа с файлом включает в себя: открытие файла, работа с ним и закрытие файла.

В программе необходимо использование нескольких потоков. Это реализуется через библиотеку <pthread.h>. В нее входит метод start\_pthread(), который запускает в новом потоке соответствующую функцию.

# Практическая часть

В ходе работы над курсовым проектом была разработана программа для децентрализованного обмена текстовыми сообщениями на языке программирования Си.

Для удобной передачи текстовых сообщений была создана структура packet (пакет), которая состоит из заголовка и самого текста сообщения. Заголовок включает в себя тип пакета, num – номер или количество пакетов и идентификатор сообщения msg\_id.

Программа реализует следующие функции: отправка сообщения, отправка сигнала, прием пакета, обработка пакета, регистрация и «вход».

Под сигналом подразумевается пакет, в котором не заполнено поле – сообщение, а заполнена только часть заголовка, а именно тип – сигнал и идентификатор сообщения. Он нужен для того, чтобы передать клиент-отправителю информацию о том, что пакет дошел или что клиент-получатель готов принять остальные пакеты (для сообщения состоящего из нескольких пакетов).

Регистрация – это чтение имени пользователя, запись его в абонентскую книжку (id = 1, имя пользователя, адрес = 127.0.0.1) и «вход». Эта регистрация позволяет отправлять сообщение самому себе. Если абонентская книжка существует, значит регистрация была произведена.

Абонентская книжка представляет собой файл, содержащий информацию о контактах пользователя – id, имя и адрес соответственно.

«Вход» – это 10 broadcast пакетов, которые содержат имя нового пользователя.

Отправка, получение и обработка пакетов разделены на потоки, именуемые менеджерами.

Когда запускается программа, происходит инициализация системы – создание сокета, который принимает все пакеты, и создание связи с адресом, по которому к пользователю будут обращаться для отправки сообщения. Затем инициализируются структуры и переменные, необходимые для нормального функционирования. Далее происходит старт системы – запуск менеджеров отправки, получения и обработки, после чего выполняется «вход» и начинается считывание сообщения, которое пользователь желает отправить.

В программе введены типы пакетов.

Когда приходит пакет типа «вход», то программа проверяет, есть ли этот пользователь в абонентской книге, если нет, то добавляет id, имя и адрес нового пользователя соответственно. Id – выбирается последовательно, от конкретного значения id не зависит работоспособность программы, что делает ее более «гибкой».

Остановимся на каждом менеджере поподробнее. Начнем с менеджера приема.

Менеджер приема опирается на односвязный список packet\_list, который помимо самого пакета и указателя на следующий элемент содержит еще адрес, с которого был отправлен пакет и id, который по умолчанию равен 0 (UNKNOWN), id заполняется далее. Хранится начало и конец вышеупомянутого списка. Все это представляет собой некоторую очередь, в которую будут записываться пакеты.

Получение новых пакетов выглядит следующим образом: создается новая структура packet\_list, пакет сохраняется в нее, заполняется адрес отправителя, после чего она добавляется в односвязный список (очередь).

В процессе отправки сообщения используется односвязный список wait\_packet\_list (очередь ожидания), но в отличие от packet\_list он содержит дополнительные поля – статус и время отправки. Передача сообщений начинается со считывания текста в главном потоке, последующему присваиванию ему уникального msg\_id, который будет записан в каждый пакет сообщения, после чего уже происходит отправка – текст разбивается на пакеты, если сообщение помещается в один пакет – это одиночное сообщение, иначе – мультисообщение. Для первого – пакет добавляется в очередь ожидания с пометкой «ожидает отправки». Для второго – пакет типа «запрос», который содержит отправляемое количество пакетов и часть сообщения, добавляется в очередь ожидания с пометкой «ожидает отправки», а остальные пакеты становятся пакетами типа мультисообщение с определенным номером и добавляются в очередь ожидания с пометкой «ожидает сигнала».

Сам менеджер отправки – пробегает по очереди ожидания и выполняет соответствующие действия. Если пакет:

1. Ждет отправки, то пакет отправляется и помечается «ожидает доставки», после чего выставляется время отправки.
2. Ждет доставки, то, если время «ожидания» закончилось, пакет отправляется повторно, время обновляется.
3. Ждет сигнала, то ничего не происходит.
4. Помечен как «доставлено», то этот wait\_packet\_list удаляется из очереди ожидания.

Отправка непосредственно пакетов подразумевает собой создание нового сокета на адрес, который берется из абонентской книжки, отправку пакета и закрытие сокета.

Менеджер обработки работает как с packet\_list, так и с wait\_packet\_list, а также со структурой mult\_msg, которая предназначена для хранения всех пакетов мультисообщения и их обработки. Когда появляется новый пакет и он добавляется в очередь, менеджер обработки замечает его в очереди и запускает обработку. Первым делом узнается, какой id у отправителя, если его нет в книжке, то тогда id становится UNKNOWN и этот пакет просто игнорируется, если это не пакет типа «вход». В зависимости от типа пакета обработка будет разной. Если тип пакета:

Сообщение, то оно сохраняется в файл «id.txt» пользователя, от которого был получен пакет и отправляется сигнал, который означает, что сообщение доставлено.

Запрос, то создается новое мультисообщение, инициализируется нужное количество пакетов, и отправляется сигнал.

Мультисообщение, то оно добавляется в mult\_msg. Когда дошли все пакеты – запускается обработка мультисообщения. Аналогично обычному сообщению, результат запишется в файл «id.txt».

Вход, то проверяется, есть ли такой пользователь, если нет, то создается новая запись в абонентской книжке.

Сигнал, то в очереди ожидания ищутся пакеты, с таким же отправителем, msg\_id и типом «ожидание доставки», после чего этот тип меняется на «доставлено». Аналогично выполняется и с типом «ожидание сигнала», только меняется на «ожидает отправки».

# Заключение

В ходе выполнения данной курсовой работы были усвоены и закреплены полученные в результате прохождения курса знания в области программирования на языке «Си», получены навыки разработки клиент-серверного приложения, а также разработана программа для децентрализованного обмена текстовыми сообщениями.

# Список использованных источников информации

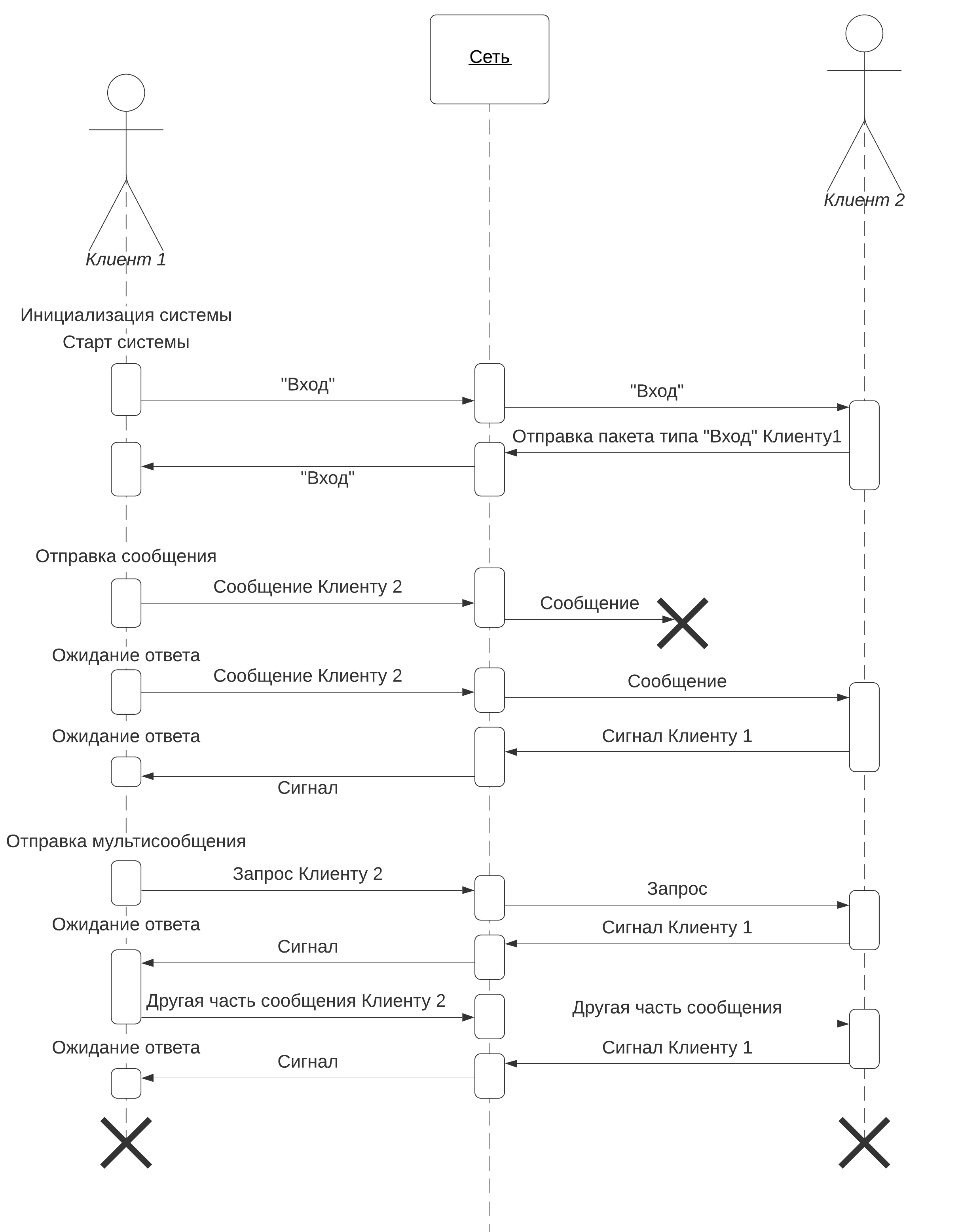
Б. Кешиган, Д. Ритчи Язык программирования Си. - 3-е изд. - Спб.: Невский

диалект, 2003. - 355 с.

# Приложение. Исходные тексты программ

## Исходный текст программы

## UML – диаграмма последовательностей



## Use-case диаграмма

