

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Лабораторная работа №3  
“Системное программное обеспечение”

Выполнил: Можанова Т Е

Группа: Р33122

Преподаватель: Кореньков Ю. Д.

Санкт-Петербург

2021г

<b>Цель работы</b>	<b>3</b>
<b>Описание работы</b>	<b>3</b>
Порядок выполнения:	3
Варианты заданий	4
<b>Выполнение</b>	<b>5</b>
<b>Вывод</b>	<b>5</b>

## Цель работы

Изучение способов взаимодействия между сетевыми службами низкого уровня в асинхронном режиме.

## Описание работы

Разработать клиент-серверное приложение. Для организации взаимодействия по сети, поддержки множества соединений использовать программные интерфейсы (API) операционной системы. Сервер и клиенты взаимодействуют по протоколу, реализованному на базе сокетов (использовать TCP, если в варианте задания не указано обратное). Сервер должен поддерживать условно неограниченное количество клиентов. На всех этапах взаимодействия клиента и сервера должна быть предусмотрена обработка данных независимо от их размера.

## Порядок выполнения:

1. Выполнить анализ предметной области, которая задается вариантом к лабораторной работе. Результатом анализа должен быть набор сущностей, которые будут в качестве элементов данных (типов, структур, операций) и/или составных частей программной архитектуры (компонентов, модулей) при реализации программы.
2. Составить диаграмму, на которой схематически будут показаны результаты анализа: сущности, их атрибуты и взаимосвязи.
3. Составить план постепенного выполнения задания: какие части функциональности, в каком порядке предполагается реализовывать, в каком порядке и как проверять их работоспособность.
4. Загрузить все полученные артефакты в отдельную директорию «docs» репозитория, в корневую директорию положить readme.md с номером варианта и кратким описанием.

5. Продемонстрировать составленные диаграмму и план преподавателю. Для этого достаточно просто отправить преподавателю ссылку на репозиторий.
6. После проверки и получения рекомендаций приступить к реализации программы, создавая на каждый этап выполнения отдельную ветку в репозитории.
7. По завершении каждого этапа создать pull-request, включив преподавателя в число reviewer-ов.
8. Если pull-request отклоняется преподавателем, выполнить необходимые правки и обновить его, запросив повторное ревью.
9. Когда pull-request одобрен, «слить» его с основной веткой кода, после чего создать новую ветку для работы над следующим этапом.

## Варианты заданий

### 1. Чат

Программа может выполняться в двух режимах: сервер или клиент. Режим определяется аргументом командной строки. В режиме сервера линейно отображается журнал всех входящих и исходящих сообщений, завершение программы-сервера выполняется по нажатию ключевой клавиши (например, Q). При запуске в режиме клиента программе в качестве аргументов командной строки также передается имя пользователя и адрес сервера. Необходимо предусмотреть возможность отправки «приватного» сообщения, которое увидит только адресат, которому оно предназначено. При подключении отображать последние 20 сообщений, предусмотреть возможность просмотра истории сообщений (не сохраняя её при этом на стороне клиентского приложения). Новые сообщения выводятся в конце списка с автопрокруткой по мере появления новых сообщений. Клавишами «стрелка вверх/вниз», Page Up/Page Down выполняется прокрутка списка сообщений, отменяющая автопрокрутку при появлении новых сообщений. Клавишей End автопрокрутка возобновляется. Собственное сообщение вводится в отдельной строке в нижней части окна, не блокируя историю сообщений.

## Выполнение

Ссылка на github - [https://github.com/tanabahh/SPO\\_LAB3](https://github.com/tanabahh/SPO_LAB3)

## Вывод

Во время выполнения лабораторной работы я изучила работу приложений на основе клиент серверной архитектуры, работы в различных потоках и концепцию реализации чатов на низкоуровневом языке программирования