

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

Мегафакультет компьютерных технологий и управления  
Факультет программной инженерии и компьютерной техники

ОТЧЁТ  
По учебно-исследовательской работе №«3» по дисциплине

«Системное программное обеспечение»

Группа: Р33122

Студент: Данков Антон Игоревич

Преподаватель: Кореньков Юрий Дмитриевич

Санкт-Петербург

2021

## Оглавление

Задание .....	3
Код .....	3
<a href="https://github.com/Antondstd/Server-Client-Total-Commander">https://github.com/Antondstd/Server-Client-Total-Commander</a> .....	3
Вариант .....	3
Описание работы.....	3
Аспекты реализации: .....	4
Вывод:.....	4

## Задание

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

## Код

<https://github.com/Antondstd/Server-Client-Total-Commander>

## Вариант

7 - Remote FS browser

## Описание работы

Программа имеет два режима – клиент и сервер. На сервере указывается начальный путь и после он начинает слушать сообщения, приходящие от пользователей. На клиенте же при запуске создается новая консоль и подсоединяется к серверу, получая информацию о файлах в корневом каталоге. После пользователь может взаимодействовать с консолью, нажимая определенные клавиши на клавиатуре, и тем самым отправляя запросы к серверу.

### Аспекты реализации:

На клиенте создается буфер под консоль, через который после и идёт вся отрисовка. Размер буфера соответствует полному размеру консоли. В самом начале также идет установление двух соединений с сервером, один под небольшие запросы, другой под передачу файлов, и так же для каждого соединения по мьютексу. При установлении соединения, сервер отправляет клиенту id, которое клиент должен или одобрить, или отнести своё соединение как относящееся к другому id.

Сервер написан через select, в котором идешь ожидание на определенные сокеты, и как только с ними происходит действие на чтение, он считывает сообщение и в зависимости от этого, уже выполняет какие то действия.

### Вывод:

Рисовать в консоли какие-либо данные оказалось не так сложно, по сути каждый наш символ представляет собой пиксель, а номер символа в буфере – координаты этих пикселей. Так же в целом пришлось почитать Windows Api для получения файлов, их времени, перевода в нынешний часовой пояс и т.д.

С сервером было сложно понять, как реализовать многопользовательскую поддержку из-за файлов. Потому между клиентом и сервером для одного пользователя хранятся два соединения и мьютексы, чтобы можно было обрабатывать данные в отдельных потоках.