**ФГАОУ ВО «Национальный исследовательский университет ИТМО»**

**Факультет программной инженерии и компьютерной техники**

**Системное и прикладное программное обеспечение**

**Лабораторная работа №3**

**Разработка клиент-серверного приложения**

**Вариант 1**

**Студент: Александров Ю.В. гр. Р4114**

**Преподаватель: Логинов И.П.**

**Санкт-Петербург**

**2023**

**Цель работы**: необходимо реализовать клиент-серверное приложение на языке Си, который для передачи данных будет использовать протокол TCP. Моделью обслуживания согласно варианту 1 будет являться: одно соединение – один дочерний процесс (на каждое соединение создается новый подпроцесс).

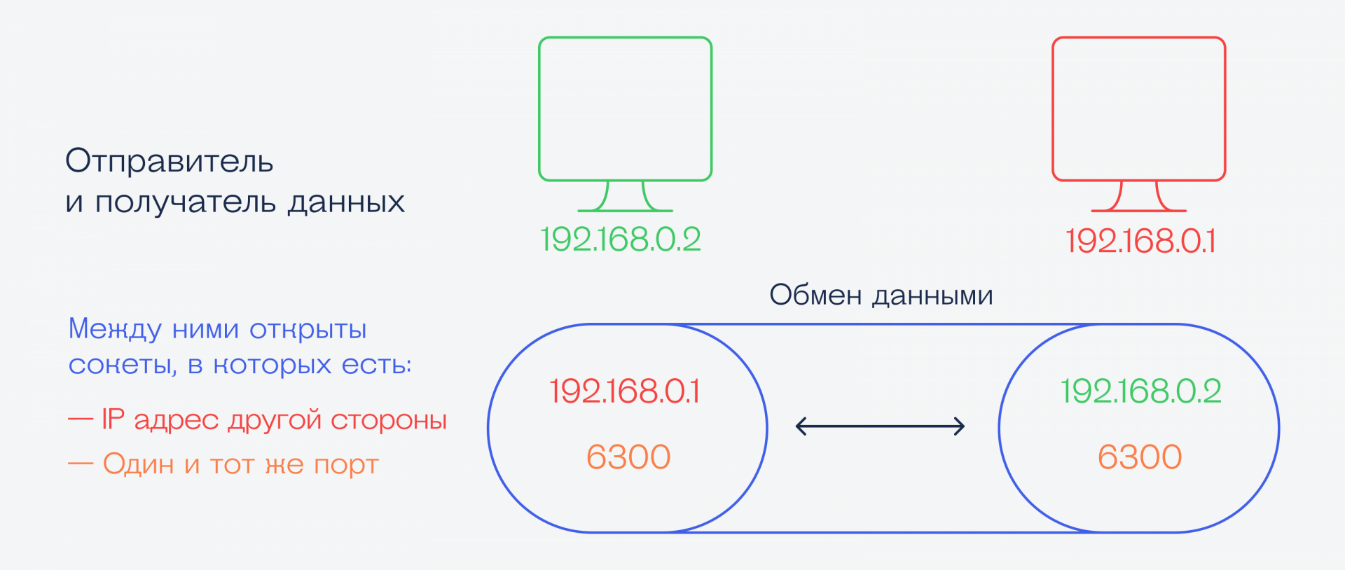
Так, необходимо реализовать сервер, на который будут приходить запросы от клиента, обрабатывать запрос и отправлять ответ клиенту. Осуществление связи клиента и сервера происходят через сокет. Сокет представляет из себя виртуальную конструкцию из IP-адреса и номера порта (рисунок 1). 

Рисунок 1 – Сокеты

При создании сокета происходит выбор типа транспортного протокола. В данной лабораторной работе был выбран протокол TCP Transmission Control Protocol). В отличии от UDP, TCP обеспечивает надежную доставку данных. Важно понимать основные функции работы с сокетами:

* socket - создание сокета;
* bind - действие используется на стороне сервера. В стандартных терминах - это открытие порта на прослушивание
* listen - используется для перевода сокета в прослушивающее состояние.
* connect - используется для инициализации соединения;
* accept - используется сервером, создает новое соединение для клиента;
* send/recv - используется для работы с отправкой/приемом данных;
* close - разрыв соединения, уничтожение сокета.

При каждом новом соединении клиента с сервером создается новый подпроцесс. Данное ответвление (подпроцесс) создаётся при помощи функции fork(). Создается подпроцесс и далее вызывается функция обрабатывающая запрос клиента read\_client.

Для реализации программы потребует 2 командные строки, одна строка будет представлять сервер, а другая клиента.

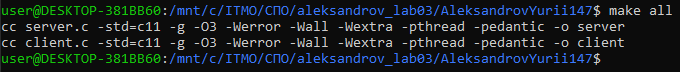


Рисунок 2 – Запуск программы

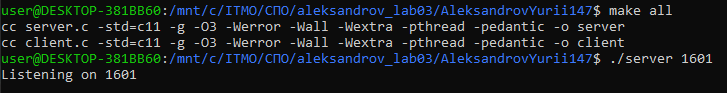
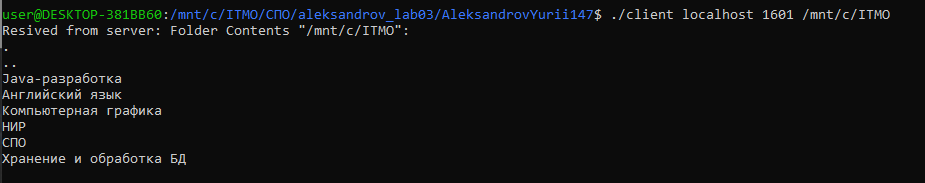


Рисунок 3 – Задание порта со стороны сервера (1601)

 Рисунок 4 – Задание IP и порта со стороны клиента, вывод результатов

Команда Telnet также позволяет отправить запрос к серверу. Сервер совместим с telnet и может работать с ним.

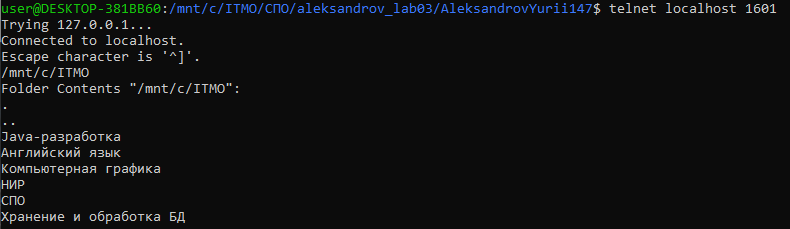


Рисунок 4 – Использование команды telnet для запроса к серверу

**Вывод:** в ходе выполнения лабораторной работы было разработано клиент-серверное приложение на языке Си. Данная работа была полезна для общего понимания работы клиент-серверных приложений.