**Лабораторная работа №7**

**Задание 1**

**Необходимые знания**

1.Компиляция программ на языке С.

-Язык программирования С является компилируемым языком. Это означает, что программа, написанная на С, во время компиляции преобразуется в набор инструкций, которые могут быть исполнены непосредственно процессором.

-Существует несколько различных компиляторов языка С. Одним из наиболее популярных компиляторов является GNU C Compiler (gcc).

-Gcc– это свободно доступный оптимизирующий компилятор для языков C, C++.

-Программа gcc, запускаемая из командной строки, представяляет собой надстройку над группой компиляторов.

-В зависимости от расширений имен файлов, передаваемых в качестве параметров, и дополнительных опций, gcc  запускает необходимые препроцессоры, компиляторы, линкеры.

-Clang может использоваться как прямая замена gcc и предлагает несколько крутых инструментов статического анализа.

2.Аргументы командной строки.

-Аргументы командной строки — это необязательные строковые аргументы, передаваемые операционной системой в программу при её запуске. Программа может их использовать в качестве входных данных, либо игнорировать. Подобно тому, как параметры одной функции предоставляют данные для параметров другой функции, так и аргументы командной строки предоставляют возможность людям или программам предоставлять входные данные для программы.

-Чтобы принять аргументы командной строки, используются два специальных встроенных аргумента: argc и argv.

-Int main (int argc, char \*argv[]) argc (англ. «argument count» = «количество аргументов») — это целочисленный параметр, содержащий количество аргументов, переданных в программу. argc всегда будет как минимум 1, так как первым аргументом всегда является имя самой программы.

-Каждый аргумент командной строки, который предоставляет пользователь, заставит argc увеличиться на 1.

-Argv (англ. «argument values» = «значения аргументов») — это место, где хранятся фактические значения аргументов. Длина этого массива — argc.

3.Протокол TCP.

-TCP - это протокол основанный на соединении, обеспечивающий коррекцию ошибок и гарантирующий доставку данных с использованием так называемого управления потоком. Управление потоком определяет когда передачу потока данных необходимо остановить и когда послать пакеты данных повторно, например, из-за сбоев в канале, тем самым убеждаясь в завершенности и аккуратности переданных данных. TCP обычно используется для обмена важной информации, таких как транзакции базы данных.

4.Протокол UDP.

-В отличие от TCP UDP — очень быстрый протокол, поскольку в нем определен самый минимальный механизм, необходимый для передачи данных. Конечно, он имеет некоторые недостатки. Сообщения поступают в любом порядке, и то, которое отправлено первым, может быть получено последним. Доставка сообщений UDP вовсе не гарантируется, сообщение может потеряться, и могут быть получены две копии одного и того же сообщения. Последний случай возникает, если для отправки сообщений в один адрес использовать два разных маршрута.

-UDP не требует открывать соединение, и данные могут быть отправлены сразу же, как только они подготовлены. UDP не отправляет подтверждающие сообщения, поэтому данные могут быть получены или потеряны. Если при использовании UDP требуется надежная передача данных, ее следует реализовать в протоколе более высокого уровня.

-Так в чем же преимущества UDP, зачем может понадобиться такой ненадежный протокол? Чтобы понять причину использования UDP, нужно различать однонаправленную передачу, широковещательную передачу и групповую рассылку.

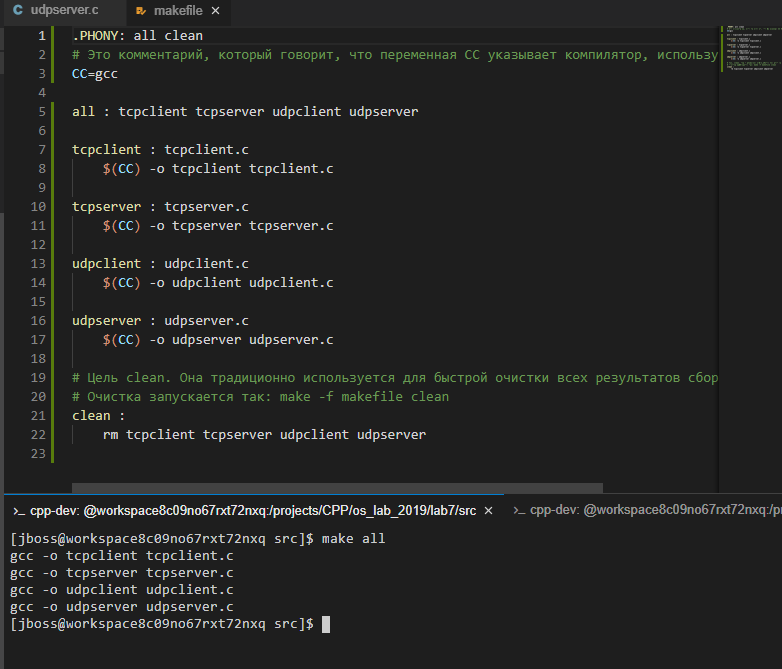
UDP может использоваться для однонаправленной связи, если требуется быстрая передача, например для доставки мультимедийных данных, но главные преимущества UDP касаются широковещательной передачи и групповой рассылки.

***В этой лабораторной работе вам предстоит потрогать два клиент-серверных приложения. Первое использует проткол TCP, второй UDP. Вам необходимо:***

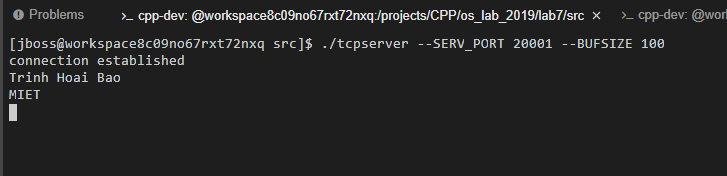
* ***Скомпилировать все четыре программы.***
* ***Вынести все константы (объявленные через #define) в аргументы командной строки.***

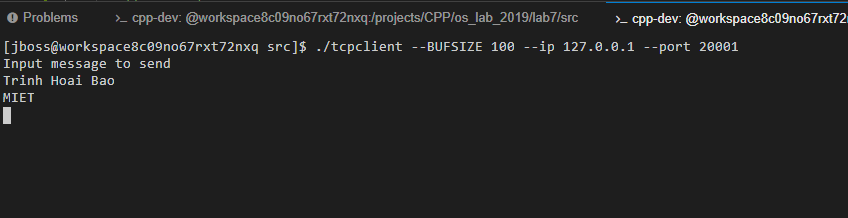
***Скомпилировать оба приложения через makefile***

***(Code in github)***

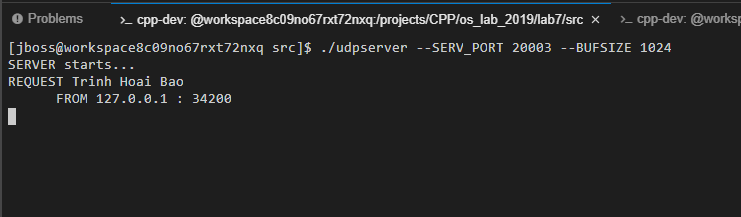
https://github.com/lehoanggia2k/os\_lab\_2019/tree/hg******

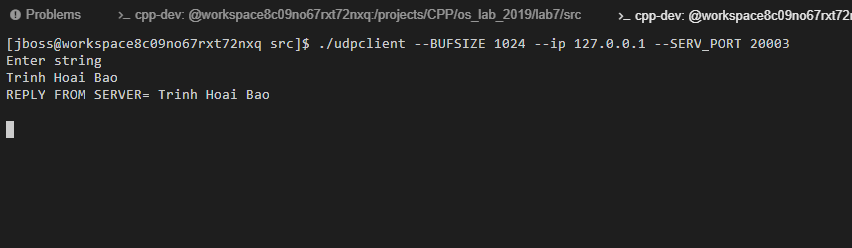
***TCP***

******

******

***UDP***

******

******

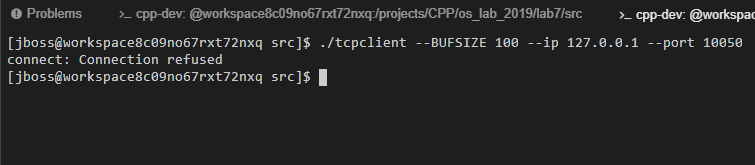
**Задание 2**

*1.Что делают оба приложения?.*

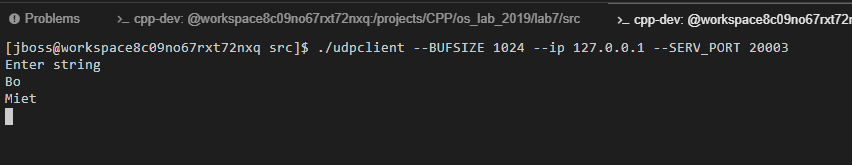
Клиент передает информацию на сервер. Сервер получает и обрабатывает информацию.

*2.Что произойдет, если tcpclient отправит сообщение незапущенному серверу?*

-Выводится сообщение об ошибке



*3.Что произойдет, если udpclient отправит сообщение незапущенному серверу?*

-Вы все еще можете отправлять информацию, но, конечно, ни один сервер не получит.

*4.Что произойдет, если tcpclient отвалится во время работы с сервером?*

-Отключение tcpclient приведёт к ожиданию сервера.

*5.Что произойдет, если udpclient отвалится во время работы с сервером?*

-Сервер находится в режиме ожидания.

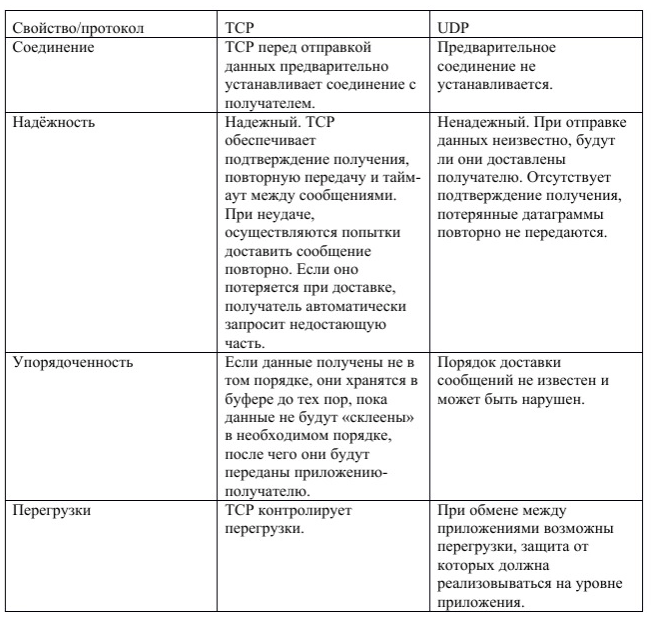
*6.Что произойдет, если udpclient отправит сообщение на несуществующий / выключенный сервер?*

-Если этот хост «не существует» (или отключен, или не работает, или его сетевой стек не работает), пакет поступит на последний маршрутизатор до этой подсети, не сможет получить локальный адрес L2 из ARP и упадет на этаж. Маршрутизатор должен вернуть ICMP-сообщение «Узел назначения недоступен».

*7.Что произойдет, если tcpclient отправит сообщение на несуществующий / выключенный сервер?*

-Если этот хост «не существует» (или отключен, или не работает, или его сетевой стек не работает), пакет поступит на последний маршрутизатор до этой подсети, не сможет получить локальный адрес L2 из ARP и упадет на этаж. Маршрутизатор должен вернуть ICMP-сообщение «Узел назначения недоступен».

*8.В чем отличия UDP и TCP протколов?*

**

-UDP обычно используется в таких приложениях, как передача аудио- и видео-потоков, поскольку он значительно быстрее TCP из-за отсутствия коррекции ошибок и управления потоком, где потеря нескольких пакетов обычно не катастрофично.

-TCP применяется там, где требуется точная и подтверждаемая передача данных – например, отправка фотографий, или переписка между пользователями. UDP, в свою очередь, нужен для общения в голосовом формате, или при передаче потокового видео, например, с веб-камер или IP-камер.