Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Курсовой проект по курсу**

**«Операционные системы»**

Студент: Хомяков Иван Андреевич

Группа: М8О-207Б-21

Вариант: -

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2022

**Содержание**

1. Репозиторий
2. Постановка задачи
3. Общие сведения о программе
4. Общий метод и алгоритм решения
5. Исходный код
6. Демонстрация работы программы
7. Выводы

**Репозиторий**

https://github.com/EbumbaE/OS\_LAB/kp

**Постановка задачи**

**Цель работы**

Цель курсового проекта

1. Приобретение практических навыков в использовании знаний, полученных в течении курса

2. Проведение исследования в выбранной предметной области

**Задание**

Вариант на удовлетворительно (может быть выбран студентом по собственному усмотрению):

Необходимо написать 3-и программы. Далее будем обозначать эти программы A, B, C. Программа A принимает из стандартного потока ввода строки, а далее их отправляет программе С. Отправка строк должна производится построчно. Программа C печатает в стандартный вывод, полученную строку от программы A. После получения программа C отправляет А сообщение о том, что строка получена. До тех пор, пока программа А не примет «сообщение о получение строки» от программы С, она не может отправлять следующую строку программе С. Программа B пишет в стандартный вывод количество отправленных символов программой А и количество принятых символов программой С. Данную информацию программа B получает от программ A и C соответственно. Способ организация межпроцессорного взаимодействия выбирает студент.

**Общие сведения о программе**

Создаются 3 программы из a.c, b.c, c.c, include.h. Также к ним линкуется библиотека ZeroMQ.

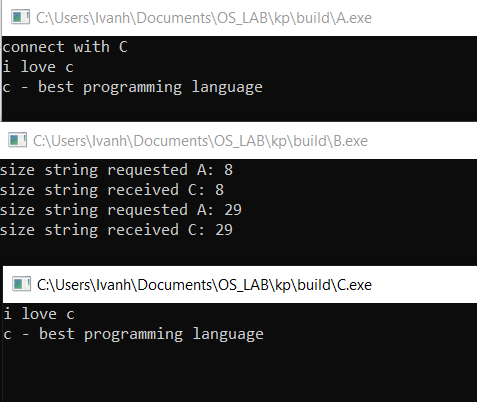
В программе используются следующие системные вызовы:

1. zmq\_ctx\_new – создает новый контекст zmq
2. zmq\_connect – создает входящее соединение на сокет
3. zmq\_disconnect – отсоединяет сокета от заданного адреса
4. zmq\_socket – создает zmq сокет
5. zmq\_close – закрывает zmq сокет
6. zmq\_ctx\_destroy – удаляет контекст zmq

**Исходный код**

В репозитории.

**Демонстрация работы программы**



**Выводы**

В курсовой работе я научился работать с библиотекой ZeroMQ. Я понял, как сделать так, чтобы программы на си асинхронно взаимодействовали друг с другом.