МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем

и технологий

**Отчет**

**по лабораторной работе № 14**

по дисциплине: ”Системное программирование”

`

# на тему: *”*Работа с очередью сообщений (Message Queue)*”*

Выполнил**:** студент группы *10702121* Нестюк Е. А.

Огризко А.В.

Принял**:** пр. Хвитько Е.А.

Минск 2023

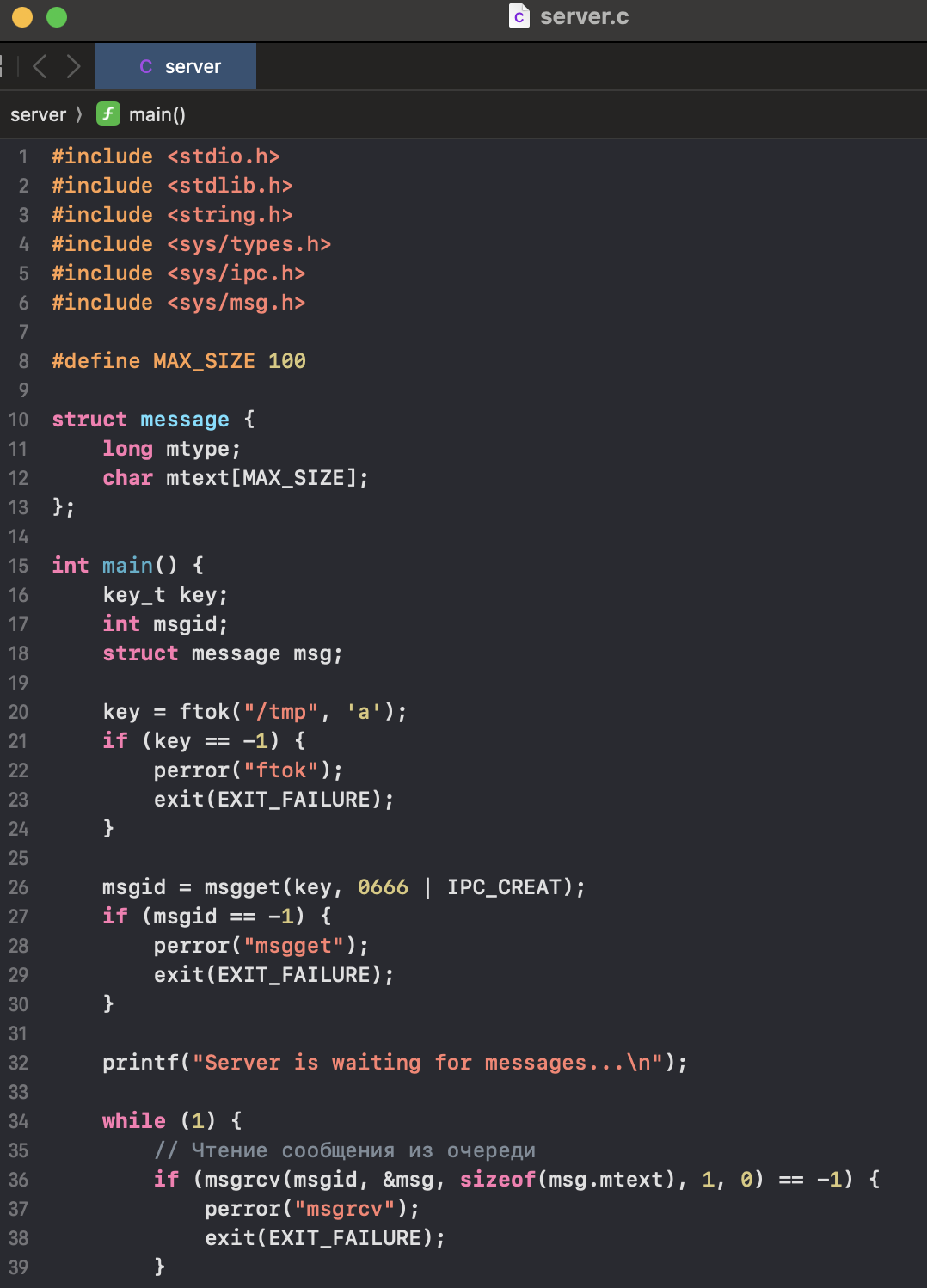
# Лабораторная работа №14.

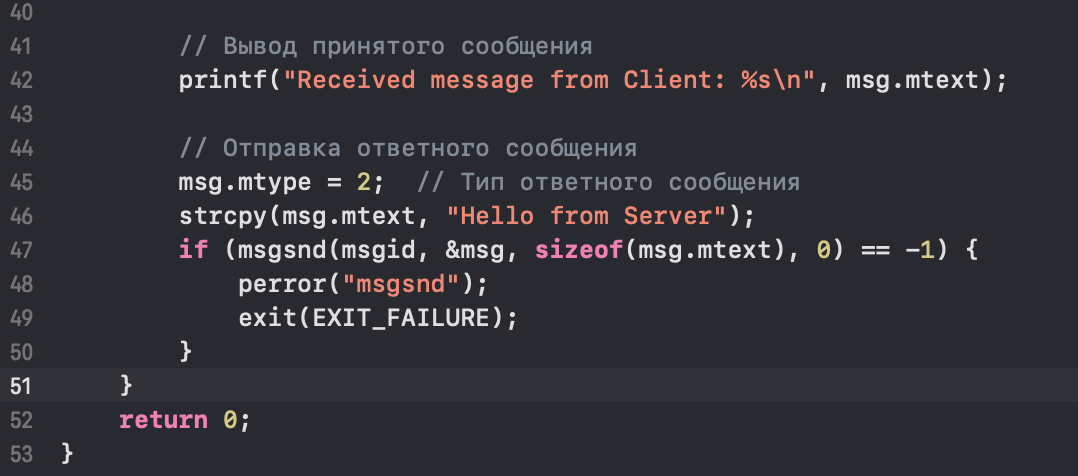
**Цель работы:** Изучить механизм очередей сообщений в LINUX.

**Задание 1**. Подключить к очереди двух клиентов (один из них выполняется как поток).

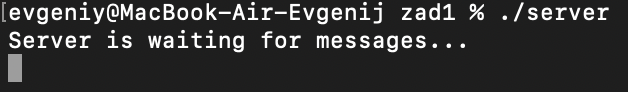
Решение:

Пишем код, где создаем очередь при помощи функции msgget();

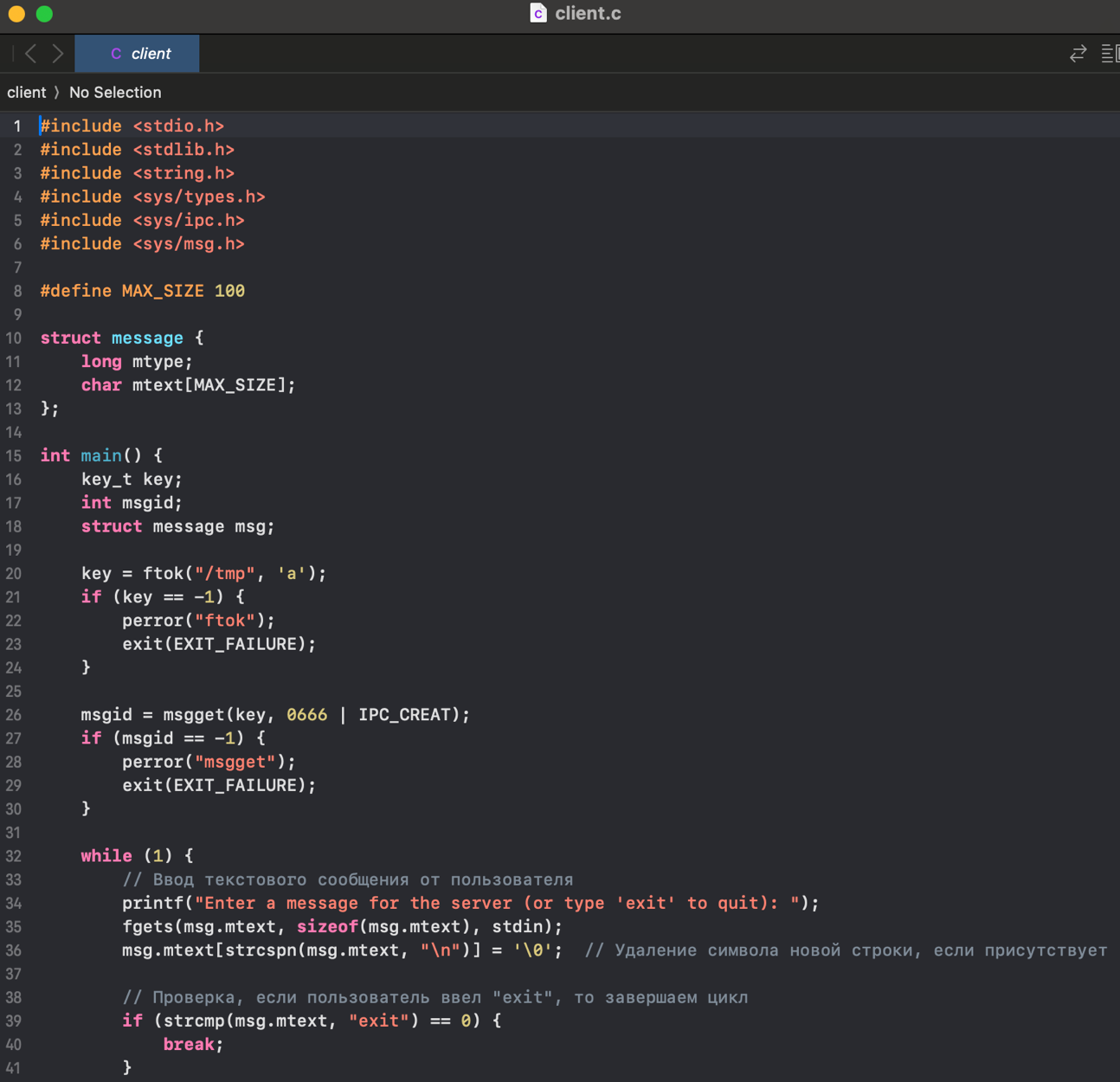




Запускаем наш код.

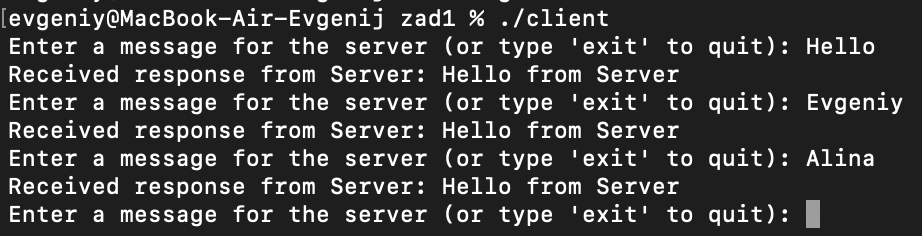


Пишем код для создания клиента

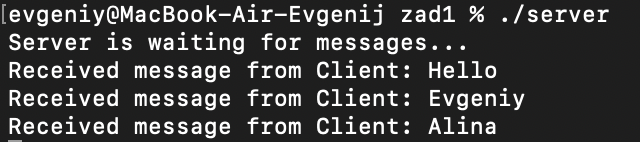




Запускаем наш клиент и передаем сообщения в очередь

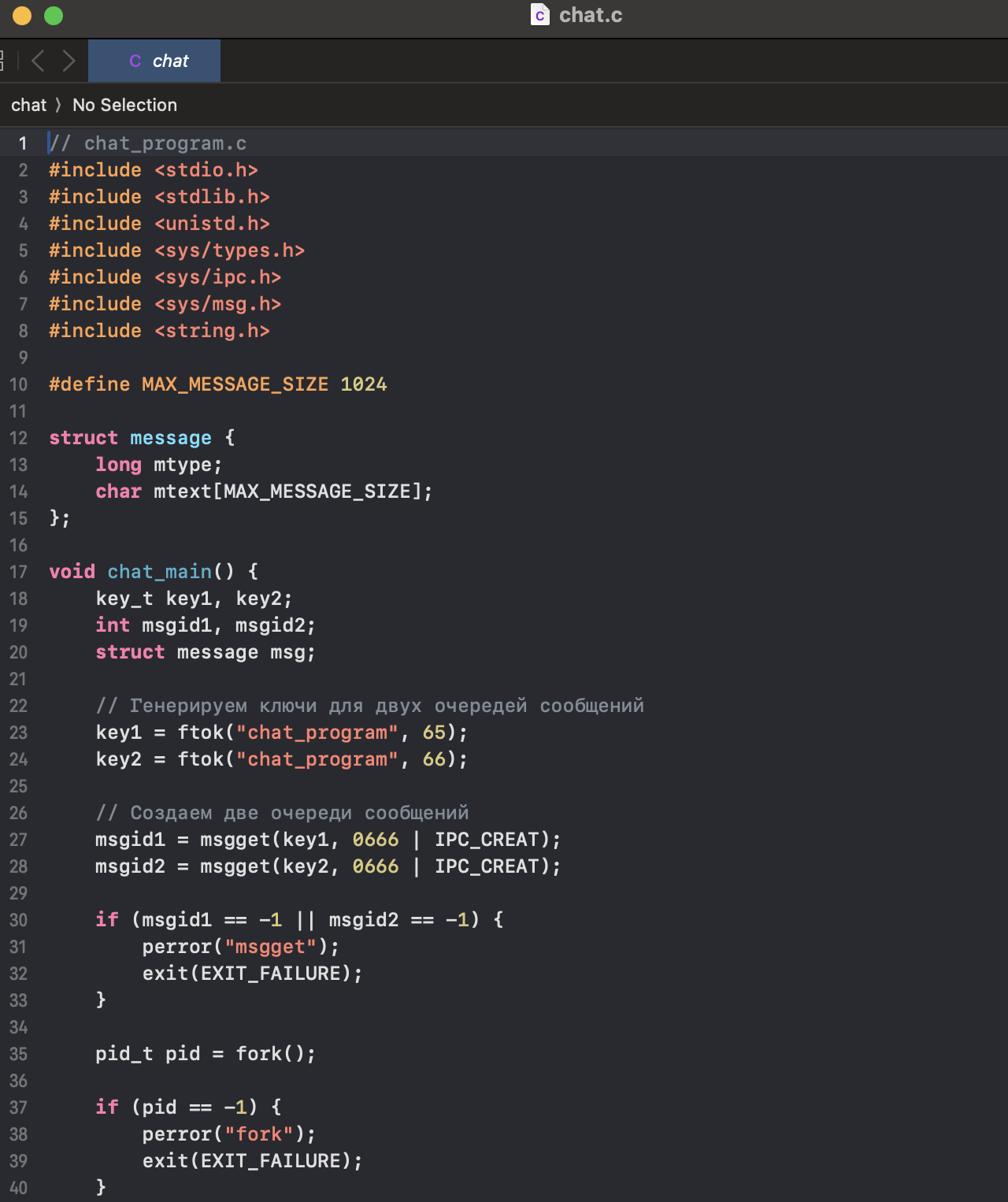


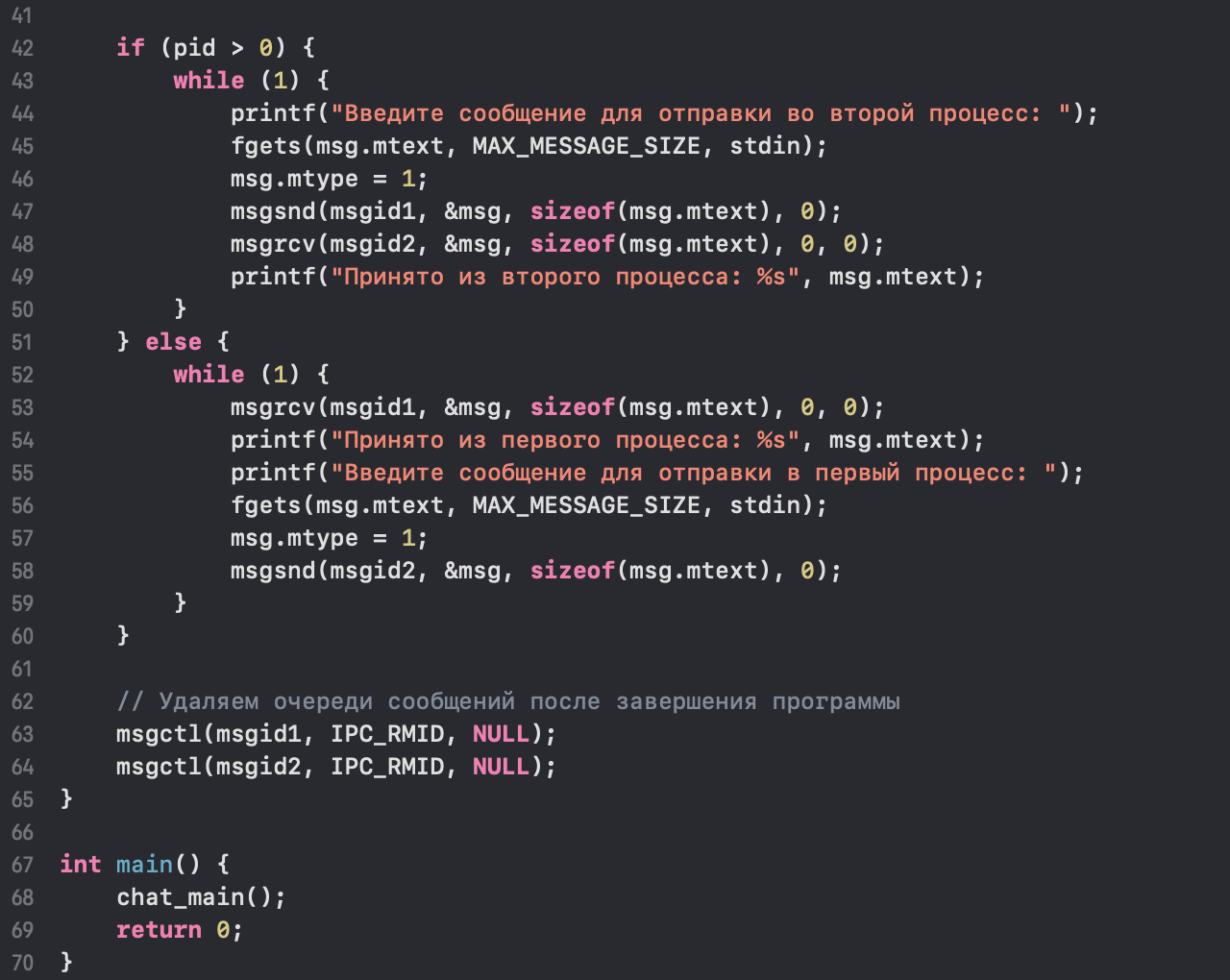
Получаем сообщения



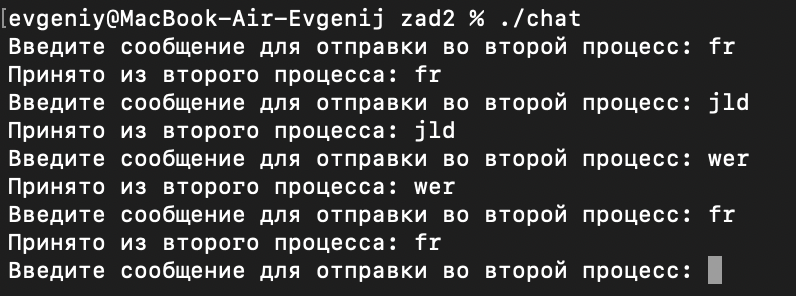
**Задание 2.** Создать две очереди сообщений. Первый процесс пишет в первую очередь и читает из второй, второй процесс, наоборот, читает из первой очереди и пишет во вторую (программа-чат).

Пишем код для создания двух очередей



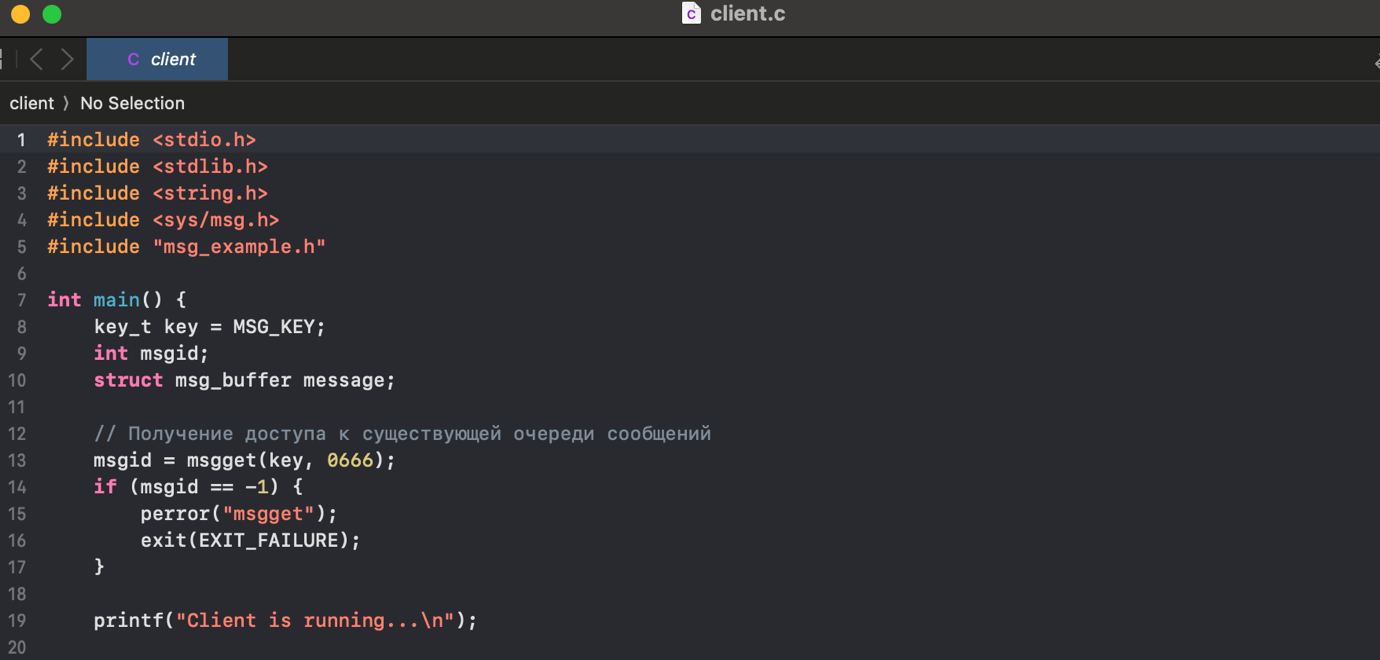


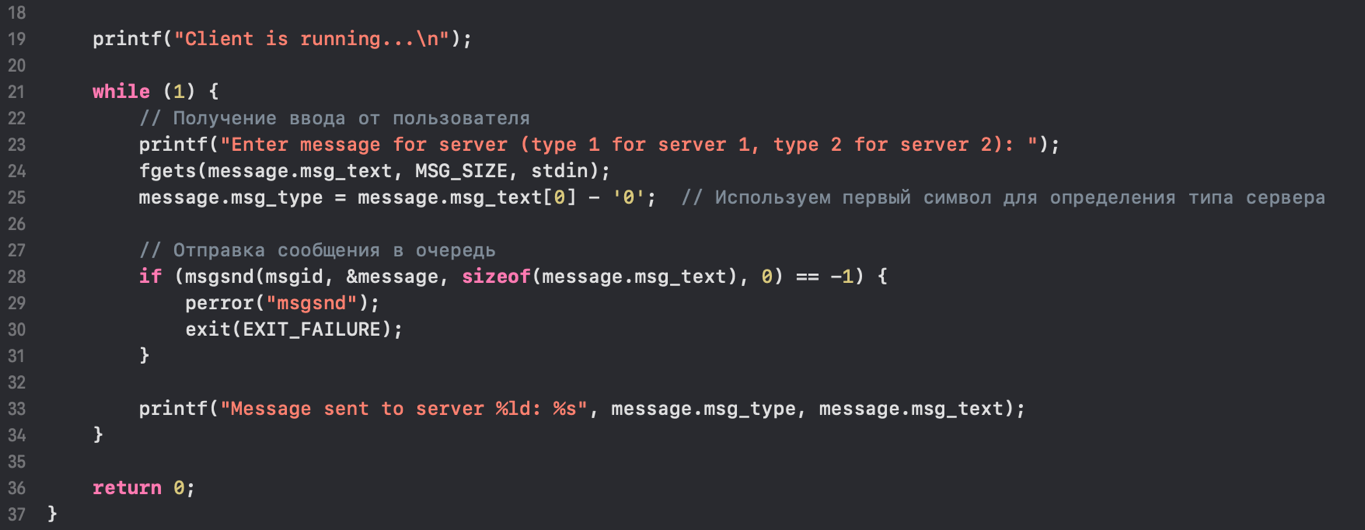
Отправляем сообщения в первую очередь и читаем из второй и наоборот



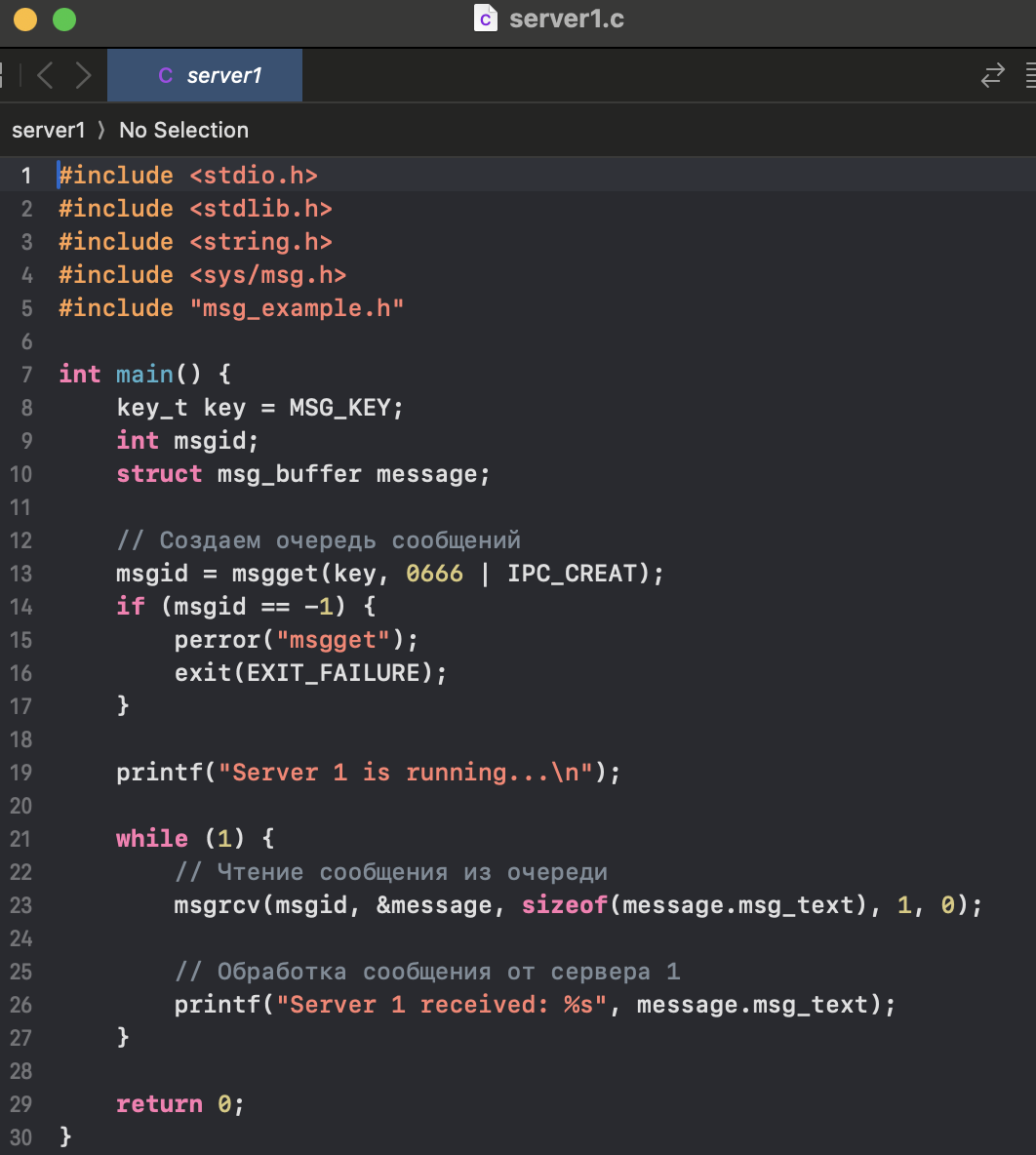
**Задание 3.** Сделать два сервера сообщений и одного клиента. Серверы используют одну и ту же очередь. Клиент читает и определяет, от какого сервера пришло сообщение.

Код клиента

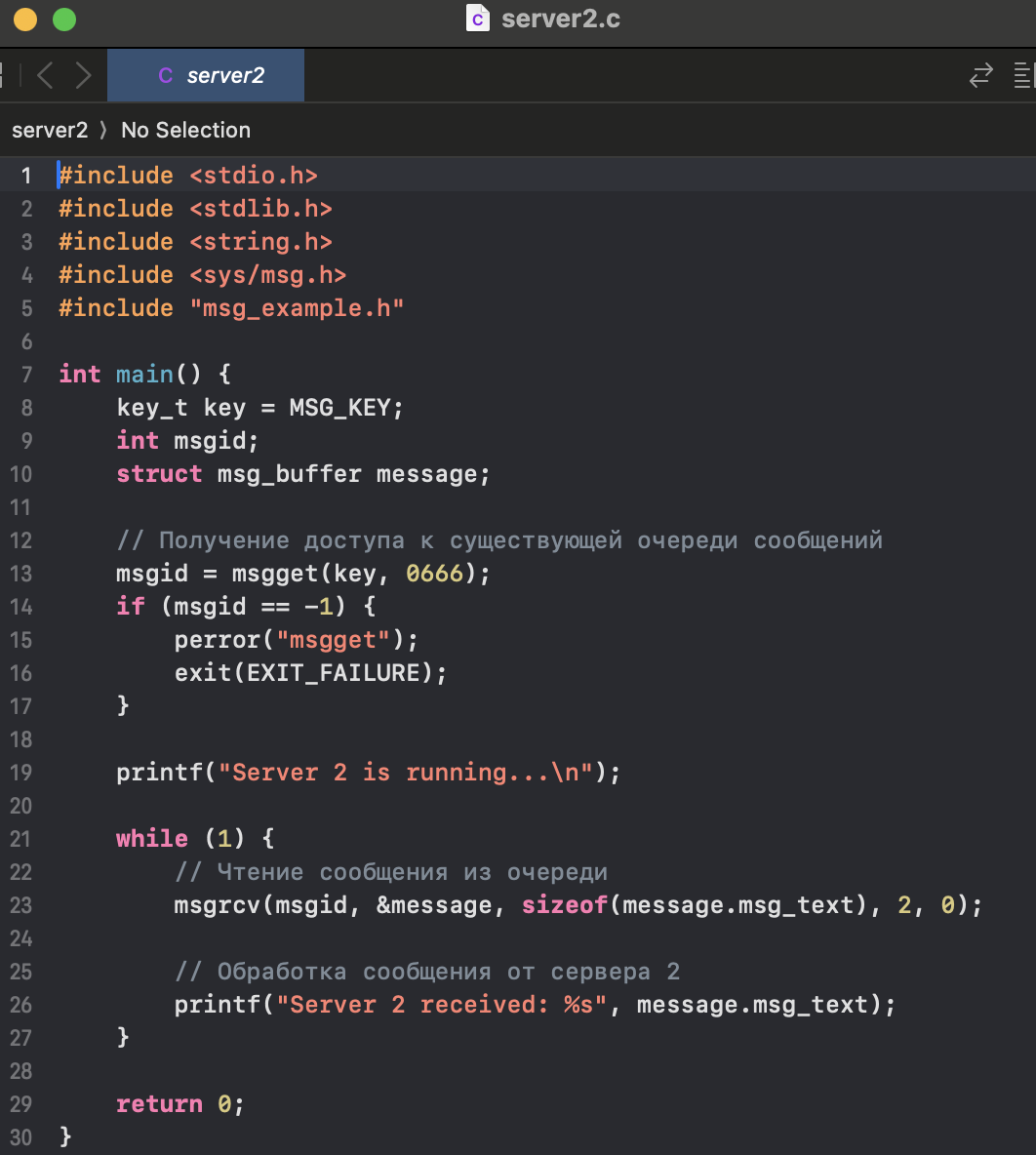




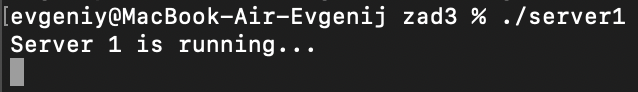
Код первого сервера



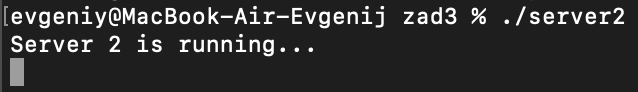
Код второго сервера



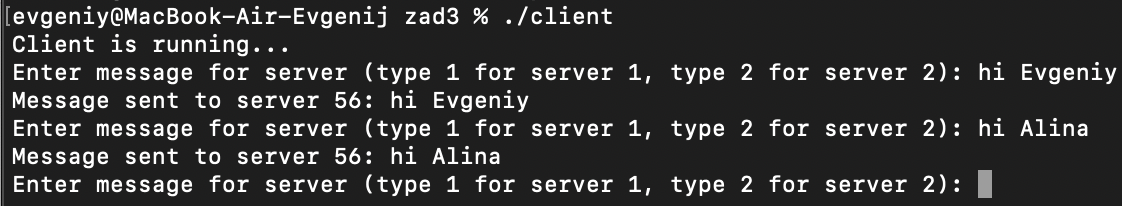
Запускаем первый сервер

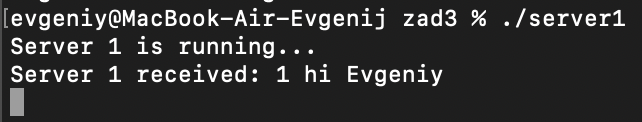


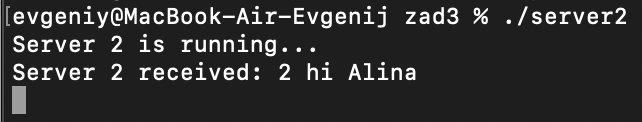
Запускаем второй сервер



Запускаем клиента и отправляем сообщения серверам







**Контрольные вопросы:**

1. **Механизм очередей сообщений в LINUX**

Механизм очередей сообщений в Linux предоставляет буфер, где процессы могут асинхронно отправлять и получать сообщения в рамках межпроцессного взаимодействия. Linux реализует два типа очередей сообщений - **System V IPC Messages** и **POSIX Message Queue**.

1. **System V IPC Messages**: Linux реализует сообщения System V IPC в виде связанных списков, отклоняясь от строгого принципа FIFO. Для использования System V IPC процесс вызывает функцию msgsnd() для отправки сообщения. Эта функция принимает идентификатор IPC очереди сообщений, размер сообщения и структуру сообщения, содержащую тип сообщения и текст. С другой стороны, другой процесс вызывает функцию msgrcv() для получения сообщения. Эта функция требует идентификатор IPC очереди сообщений, буфер для хранения сообщения, размер буфера и значение t.
2. **POSIX Message Queue**: Linux также реализует очередь сообщений POSIX для приложений, расширяя функциональность очередей сообщений. Стандарт POSIX вводит простой файловый интерфейс, который облегчает взаимодействие приложений с очередями сообщений. Одной из заметных особенностей является поддержка приоритетов сообщений.