Приложение 1. Отчет по лабораторной работе

**Федеральное государственное автономное образовательное учреждение высшего профессионального образования   
"Национальный исследовательский университет   
"Высшая школа экономики"**

  Факультет компьютерных наук

Департамент программной инженерии

Отчёт по лабораторной работе

**Лабораторная работа 9**

**Очереди сообщений в UNIX**

по курсу «Операционные системы»

Выполнил:

Принял

Фомин Сергей Сергеевич

Оценка:

Москва 2020г.

Оглавление

[Задание 1 3](#_Toc40096244)

[Задание 2 5](#_Toc40096245)

[Задание 3 8](#_Toc40096246)

[Задание 4 11](#_Toc40096247)

[Задание 5 14](#_Toc40096248)

**Для каждого задания указывается:**

1. Формулировка задания.
2. Исходный текст программы.
3. Формат вызова программы.
4. Результат работы программы.
5. Объяснения полученных результатов.

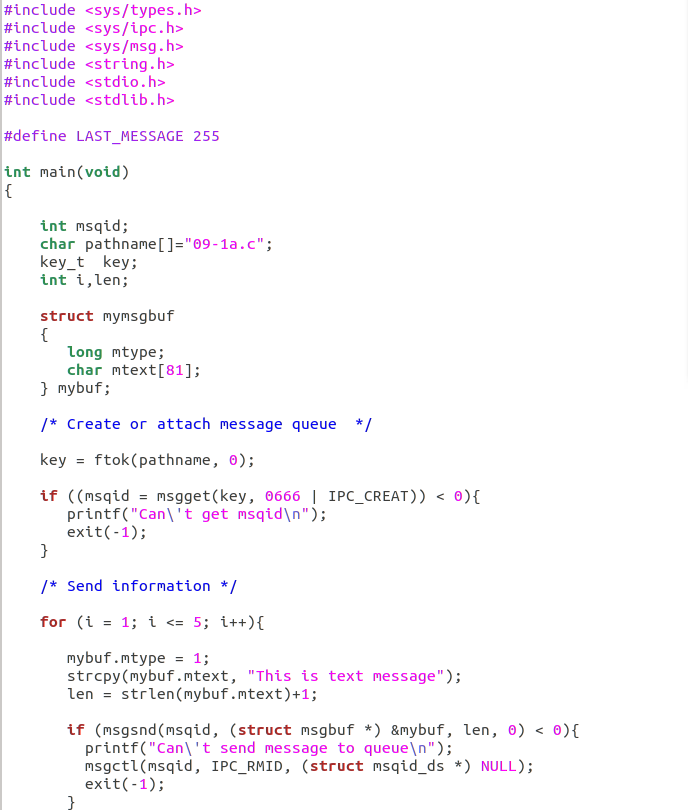
# Задание 1

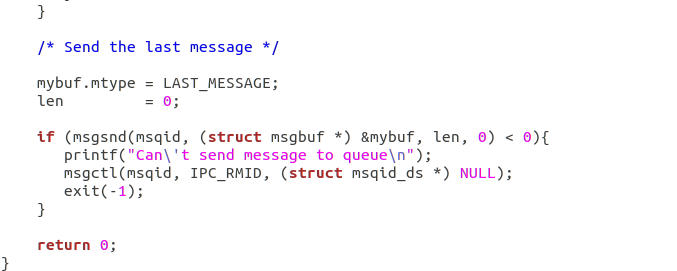
Откомпилируйте программы 09-1а.с и 09-1b.c и проверьте правильность их поведения.

(3 балла)

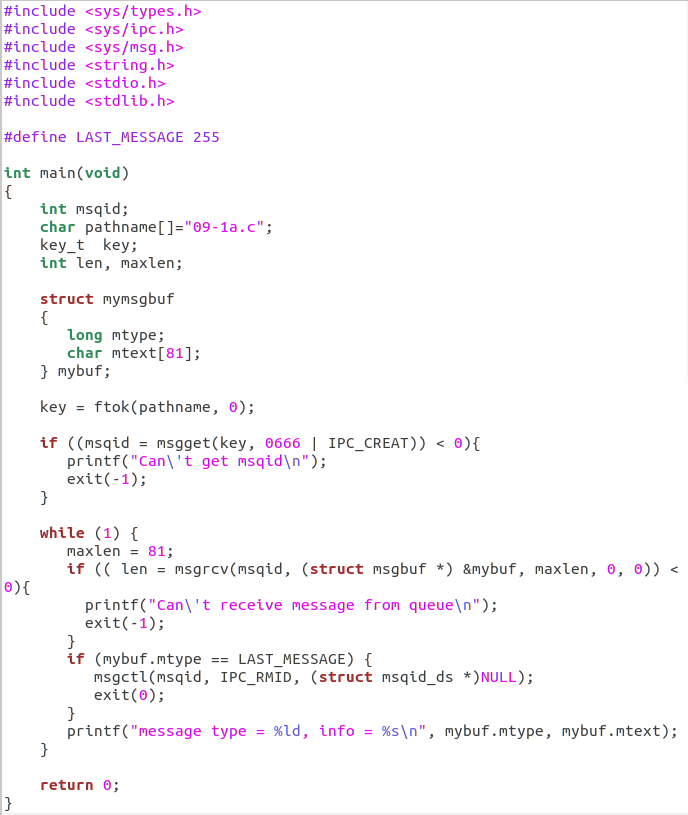
Текст программы:

09-1a.c





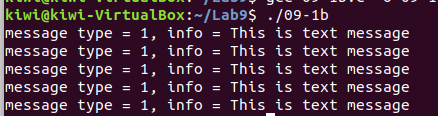
09-1b.c



Формат вызова:



Результат работы программы:



Объяснение:   
Программа 09-1a.c:  
Программа получает доступ к очереди сообщений, отправляет в нее 5 текстовых сообщений с типом 1 и одно пустое сообщение с типом 255 (объявляется в define), которое будет служить для программы 09-1b.c сигналом прекращения работы.  
Изначально происходит генерация IPC-ключа из имени файла 09-1a.c в текущей директории и номера экземпляра очереди сообщений 0. Далее мы пытаемся получить доступ по ключу к очереди сообщений при помощи функции msgget, если она (очередь) существует, или создать ее, с правами доступа read & write для всех пользователей. Посылаем в цикле пять сообщений с типом 1 в очередь сообщений, идентифицируемую msqid, для этого заполняем структуру, созданную для сообщения, и определяем длину сообщения. После чего происходит отправка сообщения. Если происходит ошибка отправки, то удаляем очередь сообщений из системы. После выхода из цикла отправляем сообщения типа LAST\_MESSAGE. И помещаем сообщения в очередь.  
Программа 09-1b.c:  
Программа получает доступ к очереди сообщений и читает из нее сообщения с любым типом в порядке FIFO до тех пор, пока не получит сообщение с типом 255, которое будет служить сигналом прекращения работы.  
Изначально происходит генерация IPC-ключа из имени файла 09-1a.c в текущей директории и номера экземпляра очереди сообщений 0. Далее мы пытаемся получить доступ по ключу к очереди сообщений при помощи функции msgget, если она (очередь) существует, или создать ее, с правами доступа read & write для всех пользователей. В бесконечном цикле принимаем сообщения любого типа в порядке FIFO с максимальной длиной информативной части - 81 символ, до тех пор, пока не поступит сообщение с типом LAST\_MESSAGE. При получении сообщения данного типа происходит удаление очереди сообщений из системы, иначе – печатаем текст сообщения.

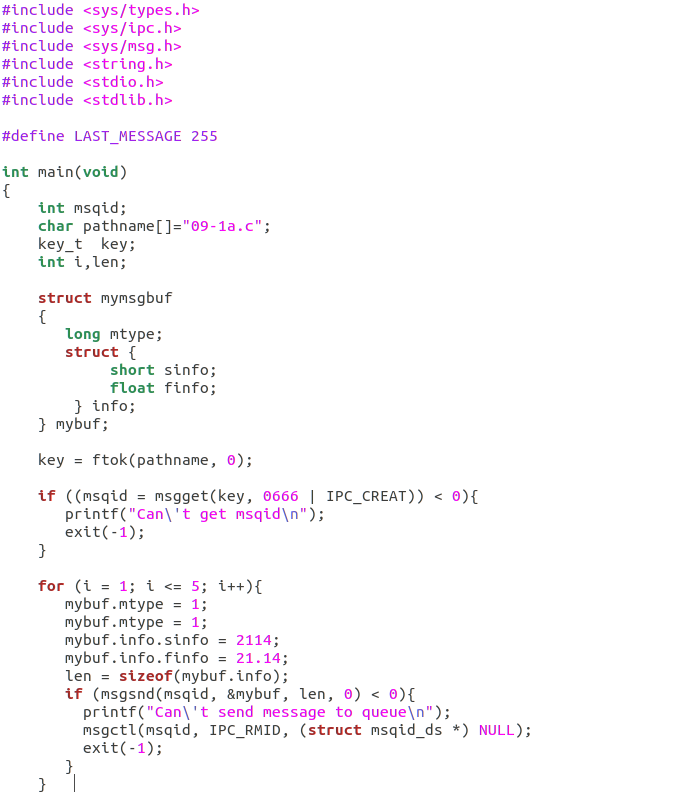
# Задание 2

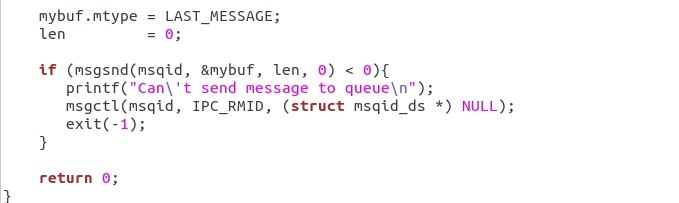
Модифицируйте предыдущие программы 09-1a.c и 09-1b.c из раздела «Прогон примера с однонаправленной передачей текстовой информации» для передачи **нетекстовых** сообщений.

(3 балла)

Текст программы:

09-2a.c





09-2b.c



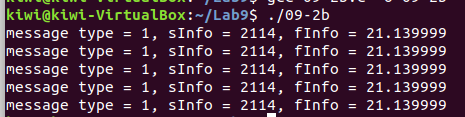


Формат вызова:





Результат работы программы:



Объяснение:  
В описании системных вызовов msgsnd() и msgrcv() говорится о том, что передаваемая информации не обязательно должна представлять собой текст, поэтому при передаче разнородной информации целесообразно информативную часть объединять внутри сообщения в отдельную структуру. Для полной передачи информативной части сообщения в качестве длины нужно указывать не сумму длин полей, а полную длину структуры.

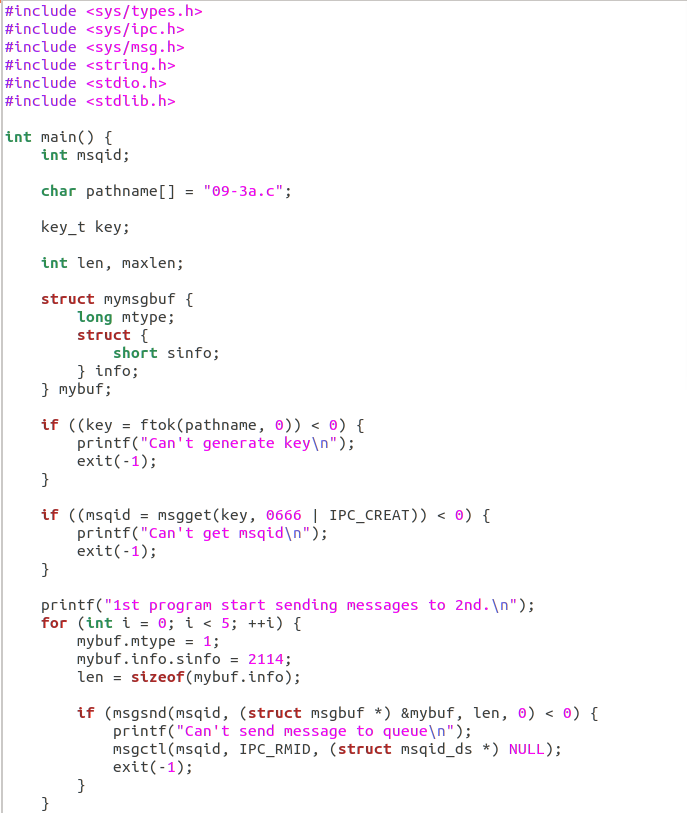
# Задание 3

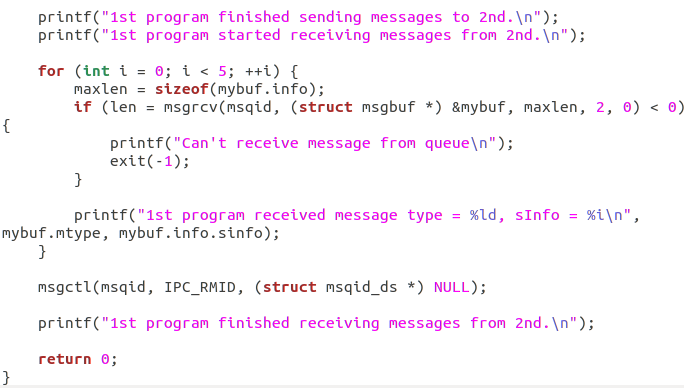
Напишите, откомпилируйте и прогоните программы, осуществляющие **двустороннюю** связь через **одну очередь** сообщений

(5 баллов)

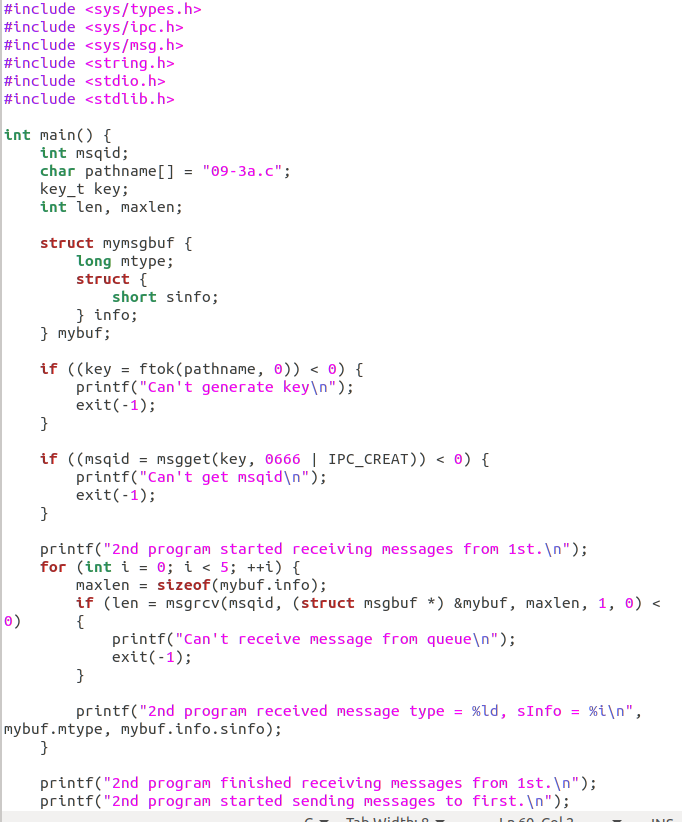
Текст программы:

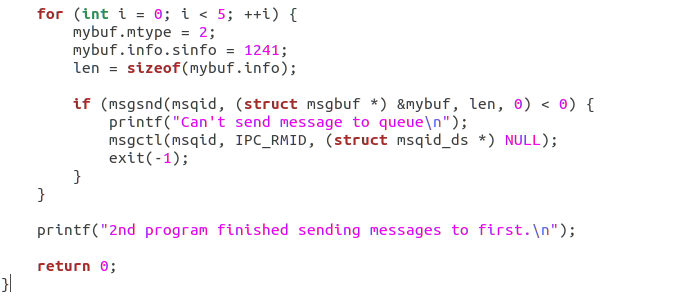
09-3a.c





09-3b.c



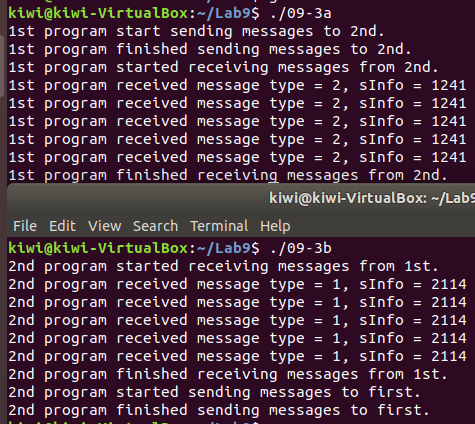


Формат вызова программы:





Результат работы программы:



Объяснение:  
Очереди сообщений, как и семафоры, и разделяемая память, являются средством связи с непрямой адресацией, требуют инициализации для организации взаимодействия процессов и специальных действий для освобождения системных ресурсов по окончании взаимодействия. Пространством имен очередей сообщений является то же самое множество значений ключа, генерируемых с помощью функции ftok().Очереди сообщений располагаются в адресном пространстве ядра операционной системы в виде однонаправленных списков и имеют ограничение по объему информации, хранящейся в каждой очереди. Каждый элемент списка представляет собой отдельное сообщение. Сообщения имеют атрибут, называемый типом сообщения.  
Реализация примитивов send и receive обеспечивает скрытое от пользователя взаимоисключение во время помещения сообщения в очередь или его получения из очереди. Также она обеспечивает блокировку процесса при попытке выполнить примитив receive над пустой очередью или очередью, в которой отсутствуют сообщения запрошенного типа, или при попытке выполнить примитив send для очереди, в которой нет свободного места.  
Очереди сообщений, как и другие средства System V IPC, позволяют организовать взаимодействие процессов, не находящихся одновременно в вычислительной системе.  
Наличие у сообщений типов позволяет организовать двустороннюю связь между процессами через одну и ту же очередь сообщений. Процесс 1 может посылать процессу 2 сообщения с типом 1, а получать от него сообщения с типом 2.

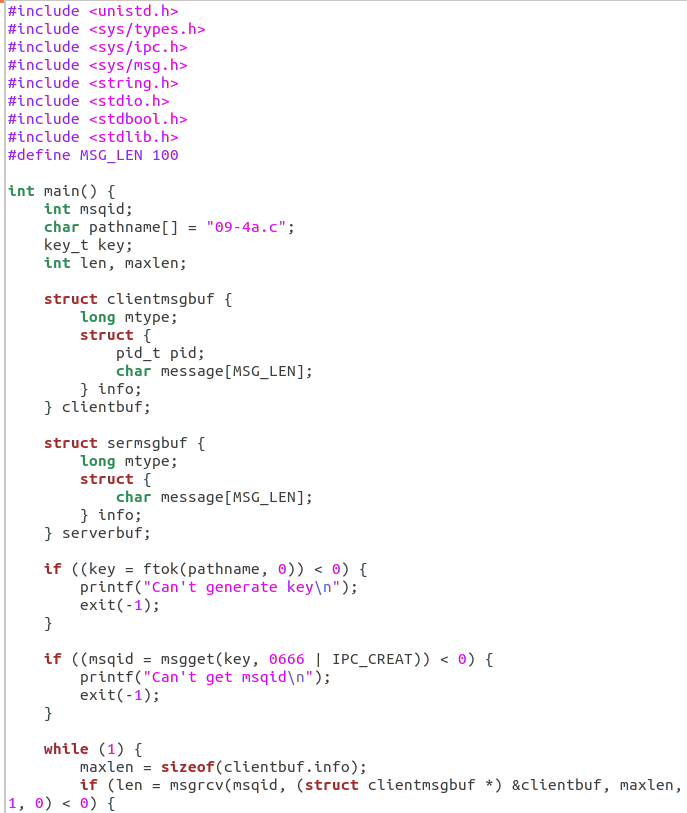
# Задание 4

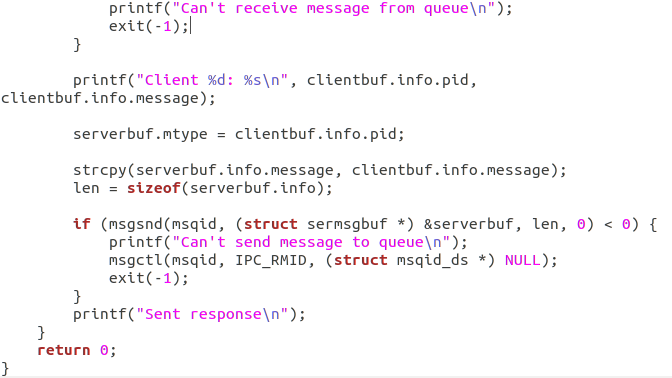
Напишите, откомпилируйте и прогоните программы сервера и клиентов для предложенной в семинаре 9 схемы **мультиплексирования** сообщений.

(10 баллов)

Текст программы:

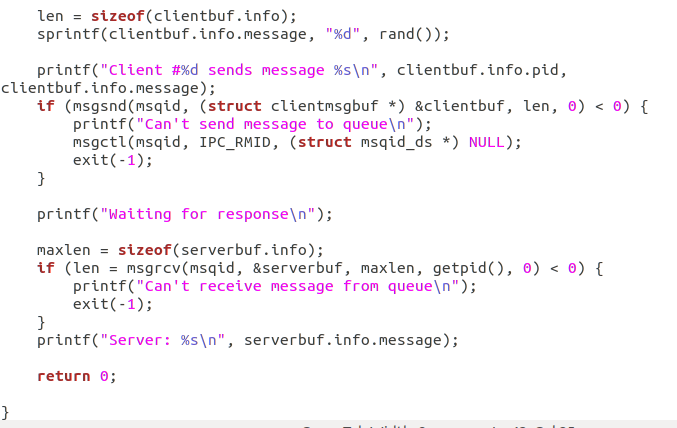
09-4a.c - сервер





09-4b.c - клиент



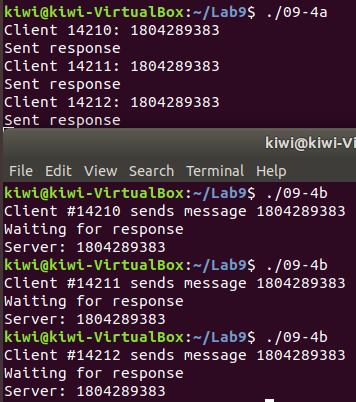


Формат вызова:





Результат работы:



Объяснение:   
Мультиплексирование сообщений - возможность одновременного обмена информацией с несколькими партнерами. Метод мультиплексирования широко применяется в модели взаимодействия процессов клиент–сервер.  
В этой модели один из процессов является сервером. Сервер получает запросы от других процессов – клиентов – на выполнение некоторых действий и отправляет им результаты обработки запросов.  
Сервер получает из очереди сообщений только сообщения с типом 1. В состав сообщений с типом 1, посылаемых серверу, процессы-клиенты включают значения своих идентификаторов процесса. Приняв сообщение с типом 1, сервер анализирует его содержание, выявляет идентификатор процесса, пославшего запрос, и отвечает клиенту, посылая сообщение с типом, равным идентификатору запрашивавшего процесса. Процесс-клиент после отправления запроса ожидает ответа в виде сообщения с типом, равным своему идентификатору. Поскольку идентификаторы процессов в системе различны, и ни один пользовательский процесс не может иметь PID равный 1, все сообщения могут быть прочитаны только теми процессами, которым они адресованы. Если обработка запроса занимает продолжительное время, сервер может организовывать параллельную обработку запросов, порождая для каждого запроса новый процесс-ребенок или новую нить исполнения.

# Задание 5

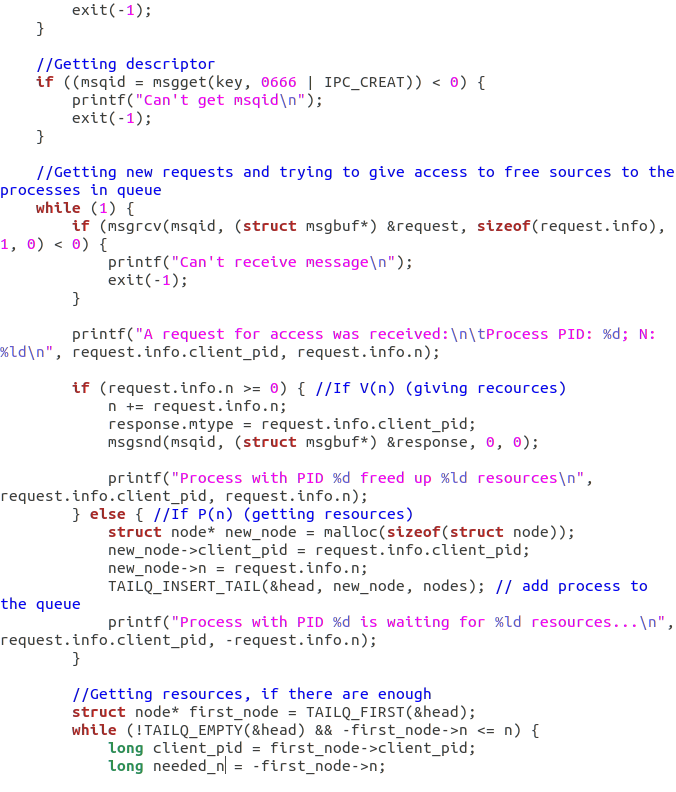
**Задача повышенной сложности:** реализуйте семафоры через очереди сообщений.

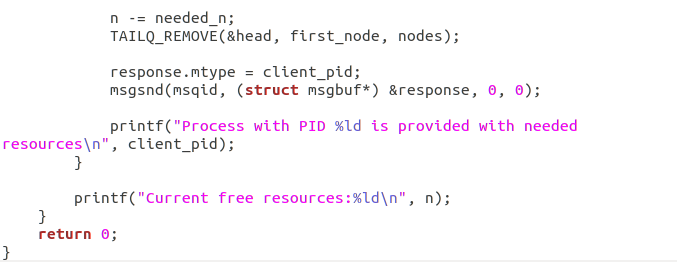
20 баллов за оригинальную реализацию  
8 баллов – объяснение известного варианта

Текст программы:

09-5a.c – server

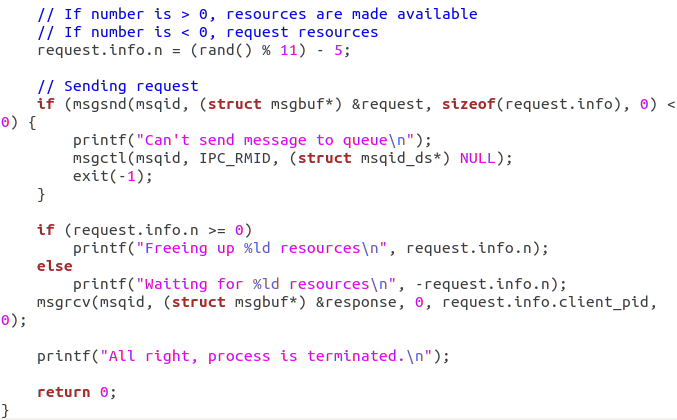




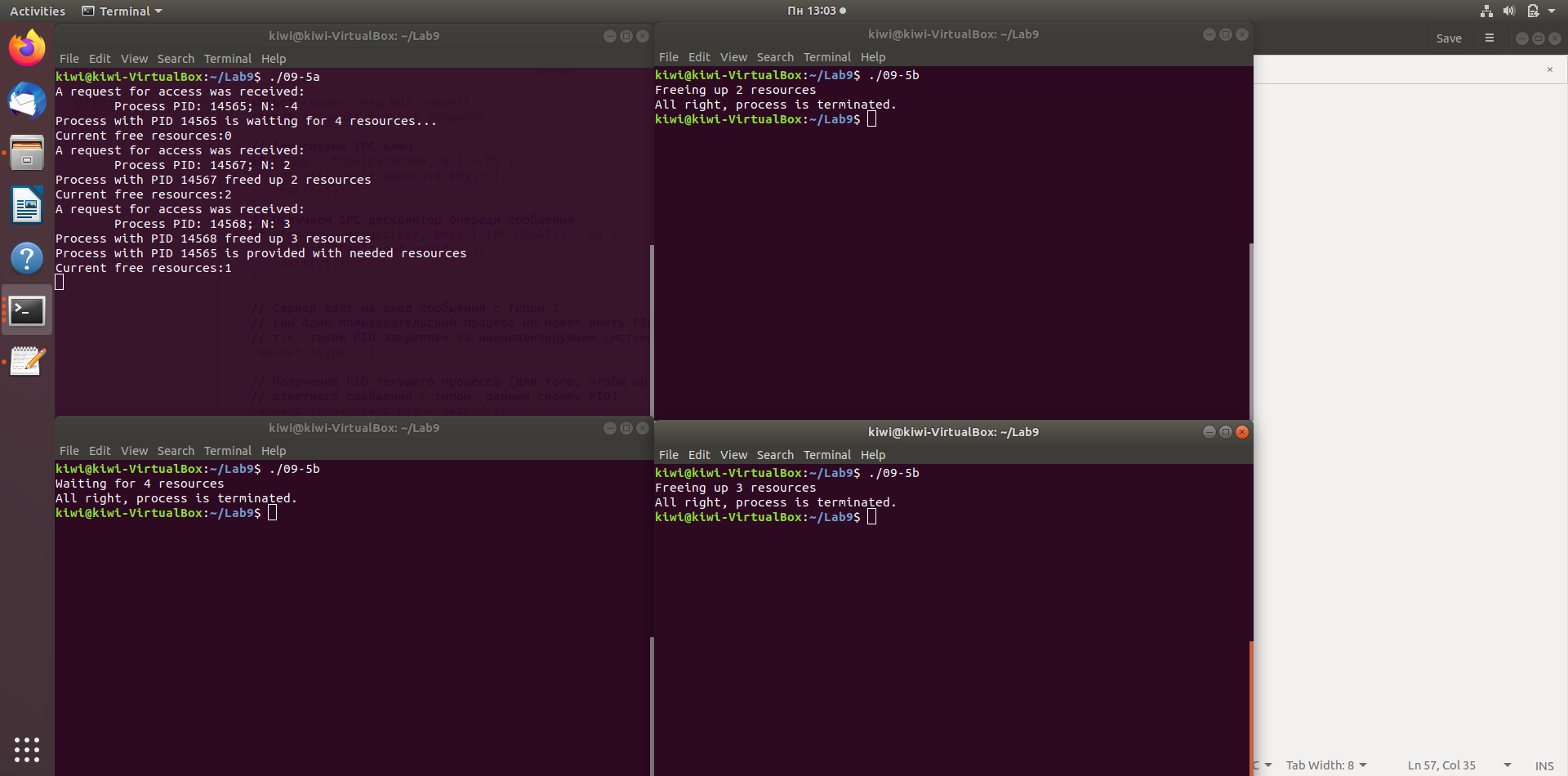


09-5b.c – client





Формат вызова:  


Результат работы программы:  


Объяснение:  
Для реализации семафоров с помощью очередей сообщений используется новый синхронизирующий процесс.   
Этот процесс имеет счетчик и очередь для процессов, ожидающих включения семафора. Для того чтобы выполнить операции P и V, процессы посылают синхронизирующему процессу сообщения, в которых указывают свои потребности, после чего ожидают получения подтверждения от синхронизирующего процесса.  
После получения сообщения синхронизирующий процесс проверяет значение счетчика, чтобы выяснить, можно ли совершить требуемую операцию. Операция V всегда может быть выполнена, в то время как операция P может потребовать блокирования процесса. Если операция может быть совершена, то она выполняется, и синхронизирующий процесс посылает подтверждающее сообщение. Если процесс должен быть блокирован, то его идентификатор заносится в очередь блокированных процессов, и подтверждение не посылается. Позднее, когда какой-либо из других процессов выполнит операцию V, один из блокированных процессов удаляется из очереди ожидания и получает соответствующее подтверждение.