**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 15

**Именованные каналы**

Студент: Тасыбаева Наталья Сергеевна

Группа: НПИбд-02-20

**МОСКВА 2021г.**

**Цель работы:** Приобретение практических навыков работы с именованными каналами.

**Ход работы:**

1. Я изучила приведённые в тексте программы server.c и client.c. Взяв данные

примеры за образец, я написала аналогичные программы, внеся некоторые изменения.

- Я создала необходимые файлы и разрешила юзеру исполнение (рис.1)

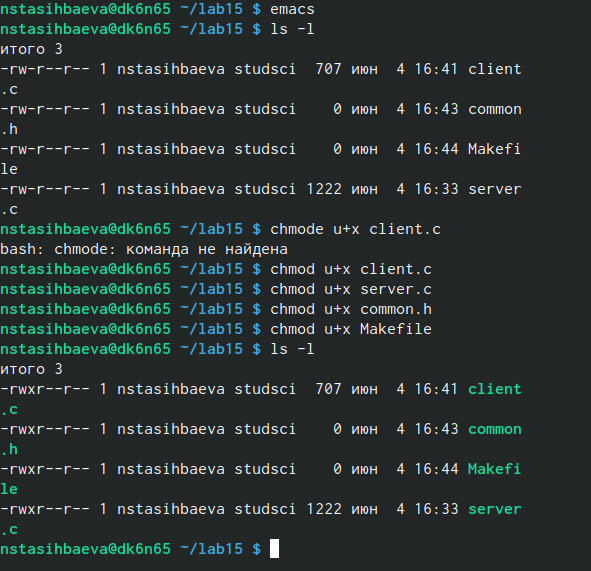


Рис.1

- Я написала файл client.с (рис.2)

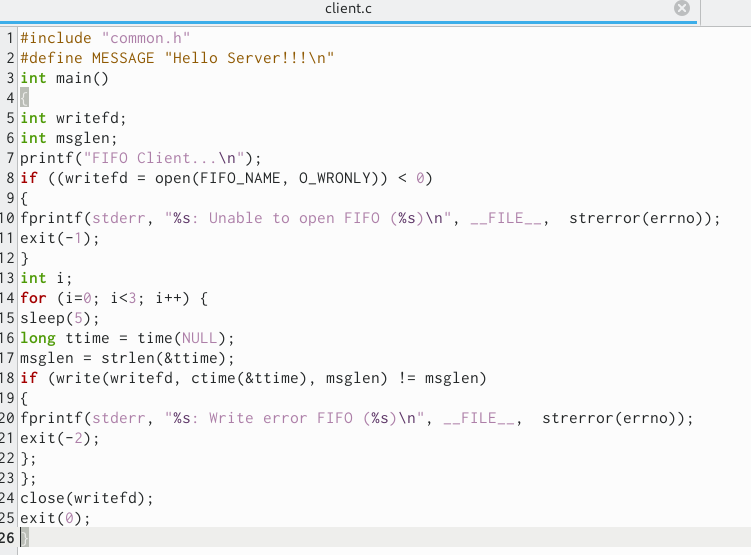


Рис.2

- Я написала файл server.c (рис.3)

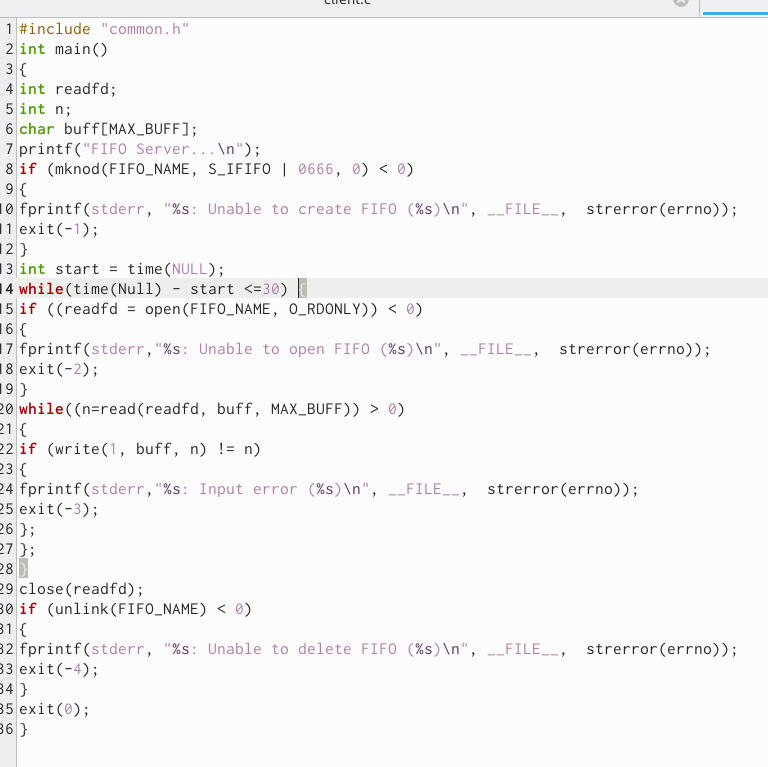


Рис.3

- Я написала файл common.h (рис.4)

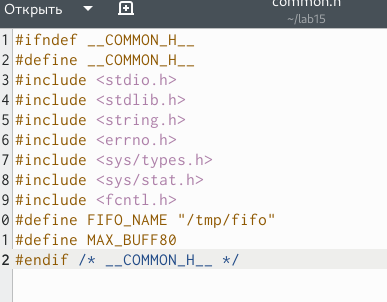


Рис.4

- Я написала Makefile (рис.5)

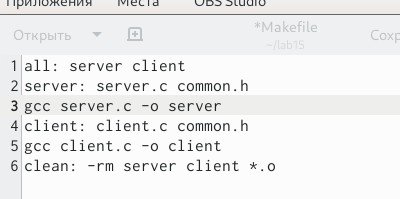


Рис.5

- В результате мы можем видеть, что открыты два клиента на одном сервере (рис.6)

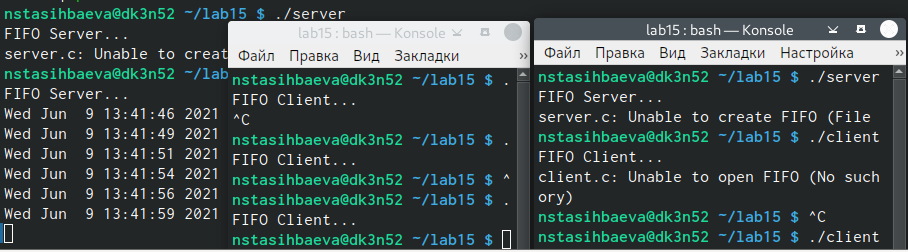


Рис.6

**Вывод:**

Я приобрела практические навыки работы с именованными каналами.

Ответы на контрольные вопросы:

1. В чем ключевое отличие именованных каналов от неименованных?

Именованные каналы отличаются от неименованных наличием идентификатора канала, который представлен как специальный файл (соответственно имя именованного канала — это имя файла).

2. Возможно ли создание неименованного канала из командной строки?

Для создания неименованного канала используется системный вызов pipe. Массив из двух целых чисел является выходным параметром этого системного вызова.

3. Возможно ли создание именованного канала из командной строки?

Да

4. Опишите функцию языка С, создающую неименованный канал.

int read(int pipe\_fd, void \*area, int cnt);

int write(int pipe\_fd, void \*area, int cnt);

Первый аргумент этих вызовов - дескриптор канала, второй - указатель на область памяти, с которой происходит обмен, третий - количество байт. Оба вызова возвращают число переданных байт (или -1 - при ошибке).

5. Опишите функцию языка С, создающую именованный канал.

int mkfifo (const char \*pathname, mode\_t mode);

Первый параметр — имя файла, идентифицирующего канал, второй параметр маска прав доступа к файлу. Вызов функции mkfifo() создаёт файл канала (с именем, заданным макросом FIFO\_NAME):

mkfifo(FIFO\_NAME, 0600);

6. Что будет в случае прочтения из fifo меньшего числа байтов, чем находится в канале? Большего числа байтов?

При чтении меньшего числа байтов, возвращается требуемое число байтов, остаток сохраняется для следующих чтений.

При чтении большего числа байтов, возвращается доступное число байтов

7. Аналогично, что будет в случае записи в fifo меньшего числа байтов, чем позволяет буфер? Большего числа байтов?

При записи большего числа байтов, вызов write(2) блокируется до освобождения требуемого места. При этом атомарность операции не гарантируется. Если процесс пытается записать данные в канал, не открытый ни одним процессом на чтение, процессу генерируется сигнал.

Запись числа байтов, меньшего емкости, гарантированно атомарно. Это означает, что в случае, когда несколько процессов одновременно записывают в канал, данные будут записывать отдельно и не будут "повреждать" структуру файла.

8. Могут ли два и более процессов читать или записывать в канал?

Могут

9. Опишите функцию write (тип возвращаемого значения, аргументы и логику работы). Что означает 1 (единица) в вызове этой функции в программе server.c (строка 42)?

Функция записывает length байтов из буфера buffer в файл, определенный дескриптором файла fd. Эта операция двоичная без буферизации. Реализуется как непосредственный вызов DOS. С помощью функции write мы посылаем сигнал клиенту или серверу

10. Опишите функцию strerror. Строковая функция strerror - функция языков C/C++, транслирующая код ошибки, который обычно хранится в глобальной переменной errno, в сообщение об ошибке, понятном человеку.