

Лабораторная работа 14

Именованные каналы

Абдулфазов Мансур али оглы

Содержание

Цель работы	2
Выполнение лабораторной работы	2
Вывод	4
Контрольные вопросы	4

Цель работы

Приобретение практических навыков работы с именованными каналами.

Выполнение лабораторной работы

1. Файл client.c (рис. 1)

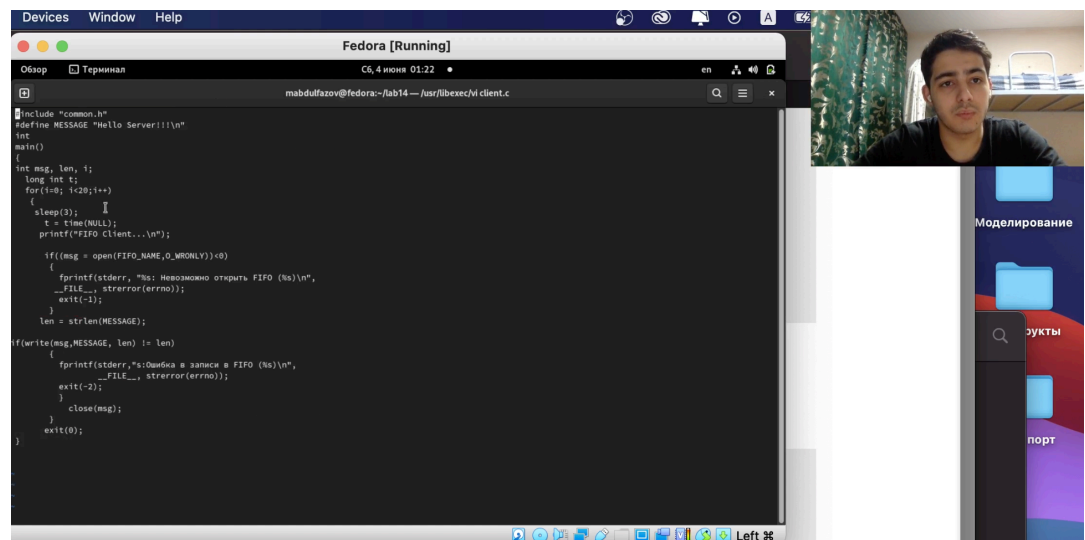


Рис. 1: client.c

2. Файл client1.c (рис. 2)

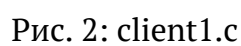


Рис. 3: server.c

3

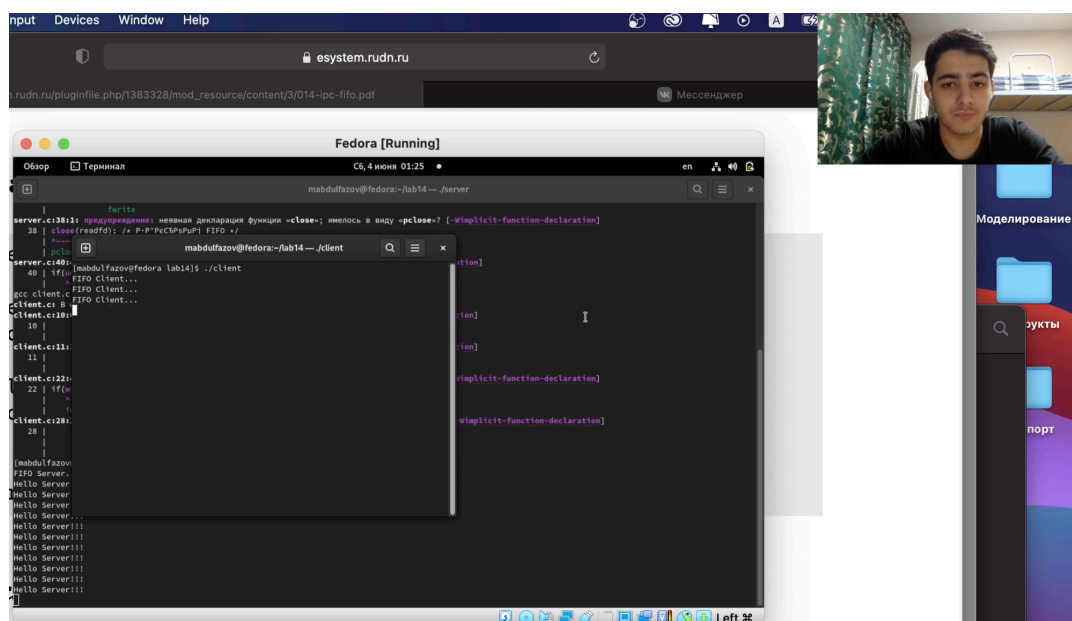


Рис. 4: Результат

Вывод

Приобрёл практические навыки работы с именованными каналами.

Контрольные вопросы

1: В чем ключевое отличие именованных каналов от неименованных?

Именованные каналы, в отличие от неименованных, могут использоваться неродственными процессами. Они дают вам, по сути, те же возможности, что и неименованные каналы, но с некоторыми преимуществами, присущими обычным файлам. Именованные каналы используют специальную запись в директории для управления правами доступа.

2: Возможно ли создание неименованного канала из командной строки?

Да, возможно

3: Возможно ли создание именованного канала из командной строки?

Да, возможно

4: Какой результат даст вычисление выражения $\$((10/3))$?

CreatePipe используется для создания неименованного канала.

5: Опишите функцию языка, создающую именованный канал.

Функции стандартной библиотеки Си, такие, как fopen, fread, fwrite и fclose позволяют обращаться к именованному каналу.

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

При чтении меньшего числа байтов, чем находится в канале или FIFO, возвращается требуемое число байтов, остаток сохраняется для последующих чтений. При чтении большего числа байтов, чем находится в канале или FIFO, возвращается доступное число байтов. Процесс, читающий из канала, должен соответствующим образом обработать ситуацию, когда прочитано меньше, чем заказано.

7: Что будет в случае записи из fifo меньшего числа байтов, чем находится в канале? Большого числа байтов?

Запись числа байтов, меньшего емкости канала или FIFO, гарантированно атомарно. Это означает, что в случае, когда несколько процессов одновременно записывают в канал, порции данных от этих процессов не перемешиваются. При записи большего числа байтов, чем это позволяет канал или FIFO, 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 SIGPIPE, а вызов `write(2)` возвращает 0 с установкой ошибки (`errno=ERRPIPE`) (если процесс не установил обработки сигнала SIGPIPE, производится обработка по умолчанию — процесс завершается).

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

Да, при этом один из процессов записывает данные в канал, а другой их считывает.

9: 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.