Отчёт по лабораторной работе №14

Именованные каналы

Надежда Александровна Рогожина

Содержание

# 1 Цель работы

Приобретение практических навыков работы с именованными каналами.

# 2 Задание

Изучите приведённые в тексте программы server.c и client.c. Взяв данные примеры за образец, напишите аналогичные программы, внеся следующие изменения:

1. Работает не 1 клиент, а несколько (например, два).
2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Используйте функцию sleep() для приостановки работы клиента.
3. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию clock() для определения времени работы сервера.

# 3 Теоретическое введение

Одним из видов взаимодействия между процессами в операционных системах является обмен сообщениями. Под сообщением понимается последовательность байтов, передаваемая от одного процесса другому. В операционных системах типа UNIX есть 3 вида межпроцессорных взаимодействий:

* общеюниксные (именованные каналы, сигналы),
* System V Interface Definition (SVID — разделяемая память, очередь сообщений, семафоры) и
* BSD (сокеты)

Для передачи данных между неродственными процессами можно использовать механизм именованных каналов (named pipes). Данные передаются по принципу FIFO (First In First Out) (первым записан — первым прочитан), поэтому они называются также FIFO pipes или просто FIFO. Именованные каналы отличаются от неименованных наличием идентификатора канала, который представлен как специальный файл (соответственно имя именованного канала — это имя файла). Поскольку файл находится на локальной файловой системе, данное IPC используется внутри одной системы.

После создания файла канала процессы, участвующие в обмене данными, должны открыть этот файл либо для записи, либо для чтения. При закрытии файла сам канал продолжает существовать. Для того чтобы закрыть сам канал, нужно удалить его файл, например с помощью вызова unlink(2).

Каналы представляют собой простое и удобное средство передачи данных, которое, однако, подходит не во всех ситуациях. Например, с помощью каналов довольно трудно организовать обмен асинхронными сообщениями между процессами.

# 4 Выполнение лабораторной работы

Изучите приведённые в тексте программы server.c и client.c. Взяв данные примеры за образец, напишите аналогичные программы, внеся следующие изменения:

1. Работает не 1 клиент, а несколько (например, два).
2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Используйте функцию sleep() для приостановки работы клиента.
3. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию clock() для определения времени работы сервера.

Первым делом составим нужные нам файлы:

* client.c для первого клиента (рис. [1](#fig:001)).

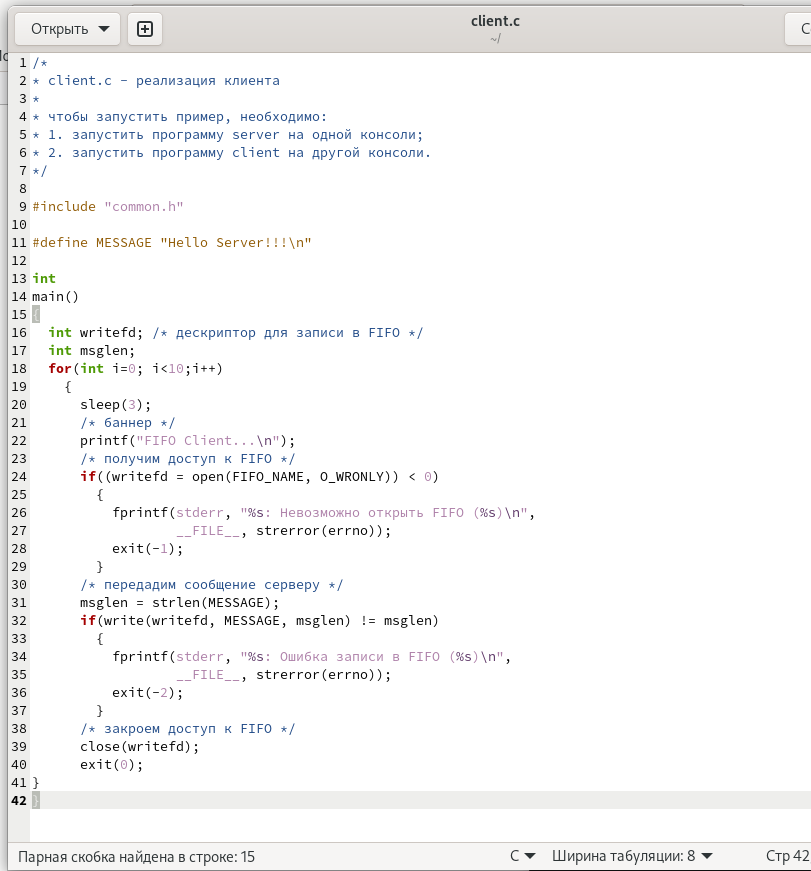


Figure 1: client.c

* client2.c для второго клиента (рис. [2](#fig:002)).

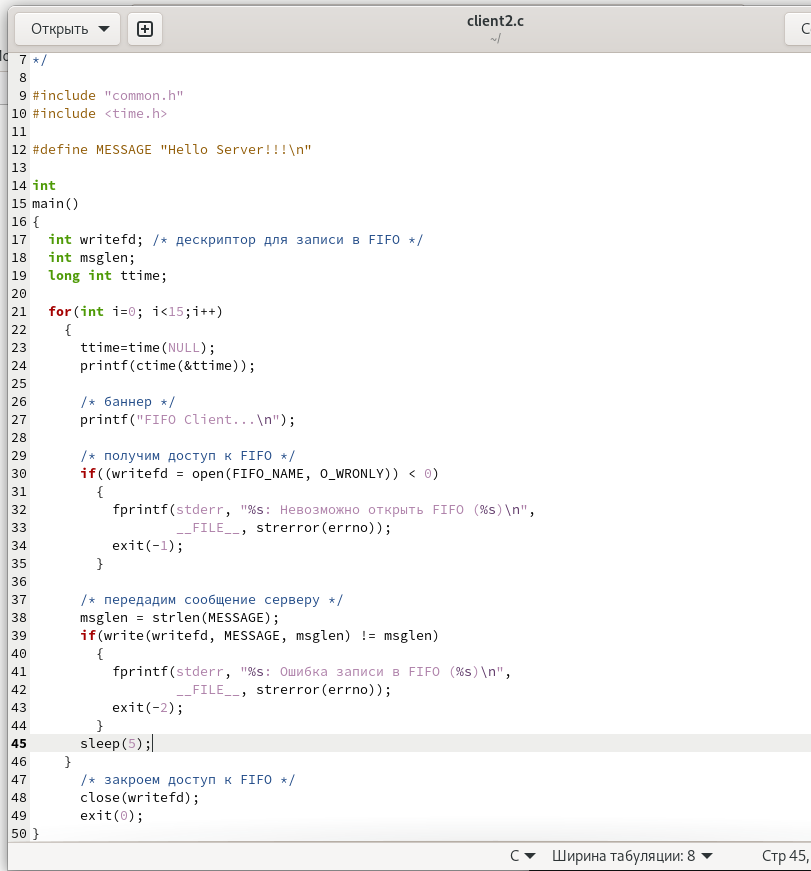


Figure 2: client2.c

* Файл самого сервера (рис. [3](#fig:003)).

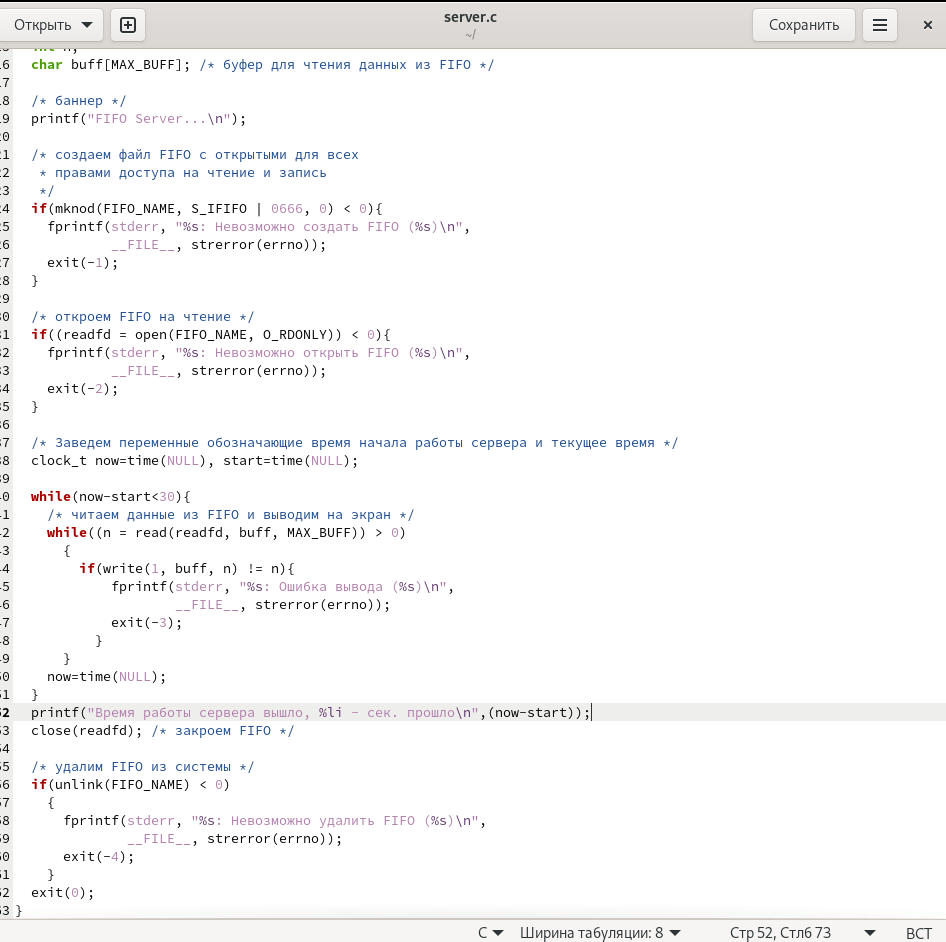


Figure 3: server.c

* Заголовочный файл со стандартными определениями (рис. [4](#fig:004)).



Figure 4: common.h

* Makefile (рис. [5](#fig:005)).

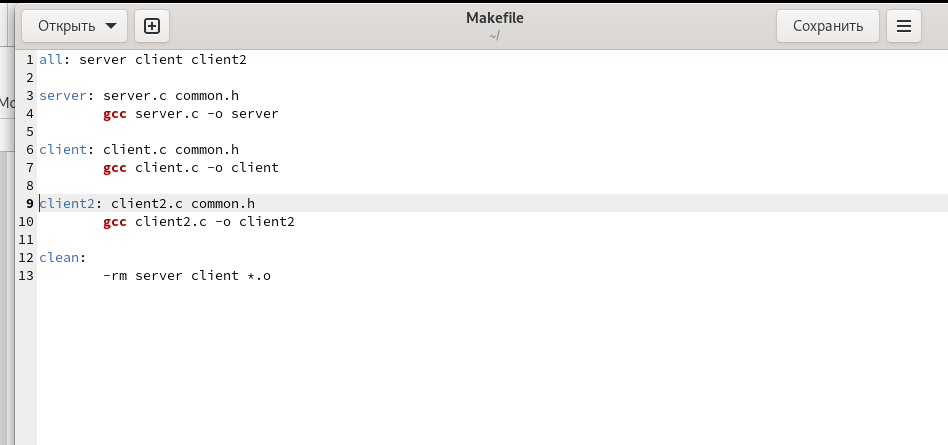


Figure 5: Makefile

После создания всех файлов запустим команду make (рис. [6](#fig:006)).

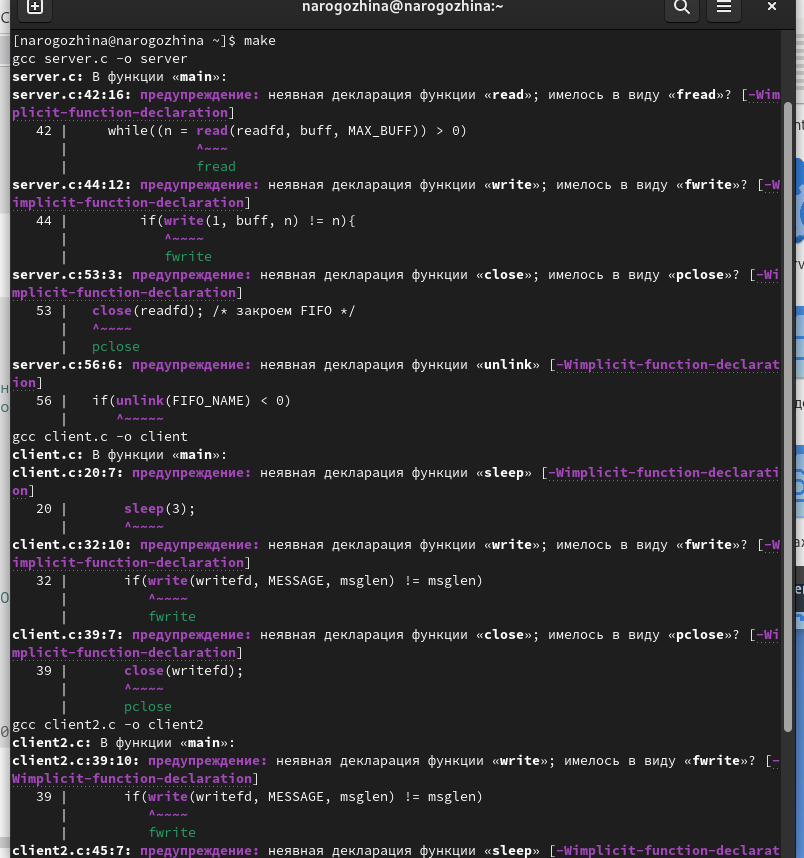


Figure 6: make

Команда выполнена успешно, и затем я открыла 3 терминала и запустила сервер и двух клиентов (рис. [7](#fig:007)).

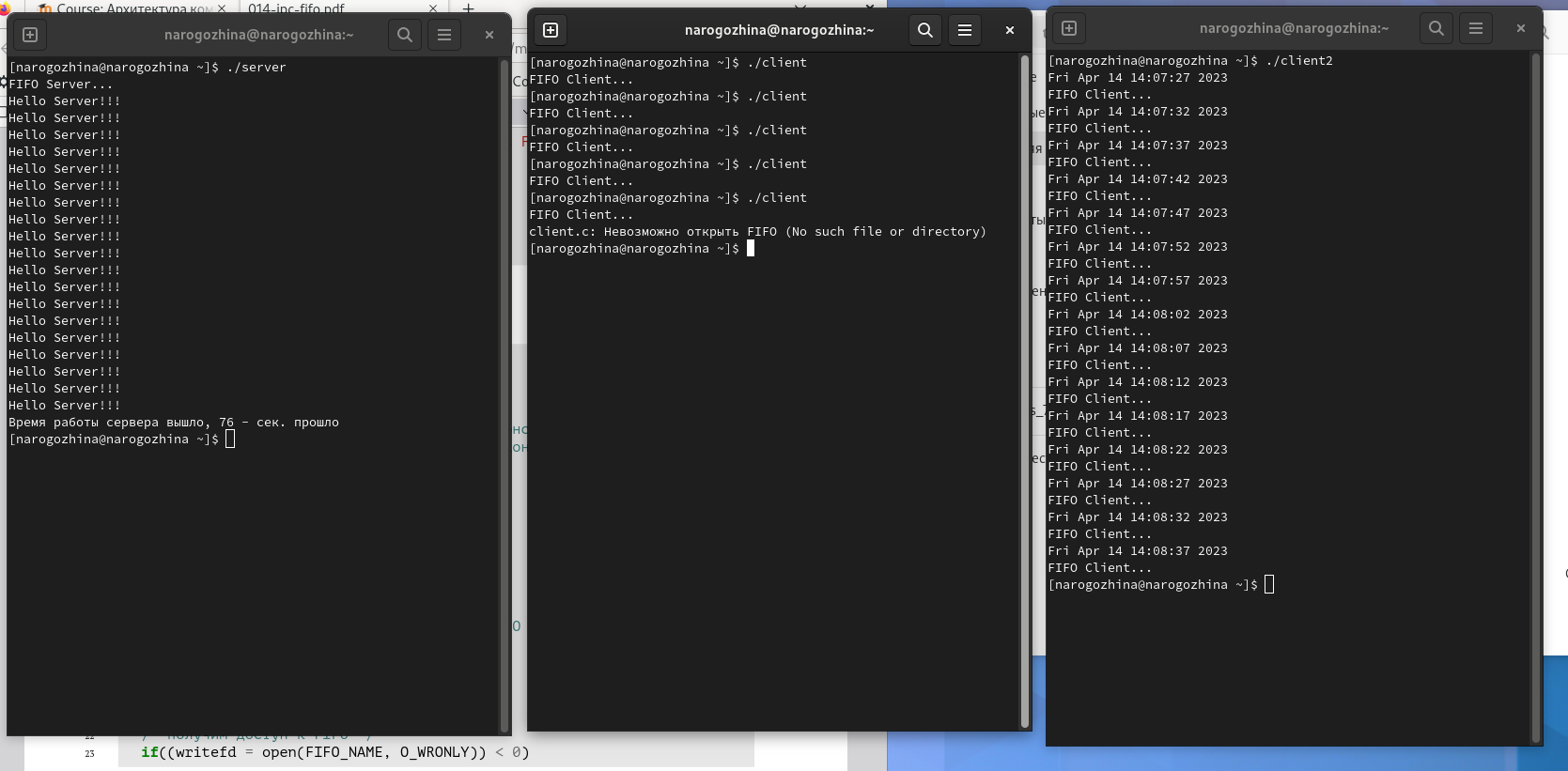


Figure 7: Запуск сервера с клиентами

Через некоторое время (-76 секунд) сервер завершил работу, и в тот же момент, я попыталась запустить первого клиента - именно из-за этого выдало “ошибку подключения”.

# 5 Выводы

В ходе лабораторной работы мы приобрели практические навыки работы с именованными каналами.

# Список литературы

1. [Руководство по выполнению лабораторной работы №14](https://esystem.rudn.ru/pluginfile.php/1975787/mod_resource/content/3/014-ipc-fifo.pdf)