Отчет

Лабораторная работа №14

Павлова Варвара Юрьевна НПМбд-02-21

Содержание

# Цель работы

Приобретение практических навыков работы с именованными каналами.

# 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 (напри- мер, 30 сек). Используйте функцию clock() для определения времени работы сервера. Что будет в случае, если сервер завершит работу, не закрыв канал?

# 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.

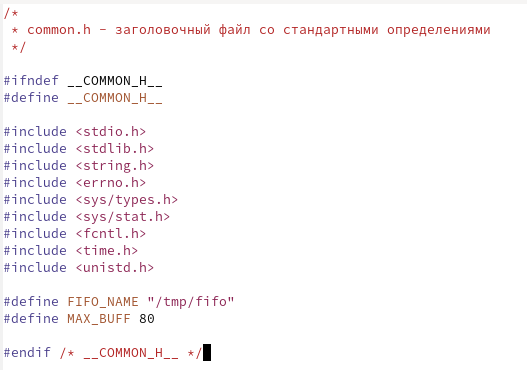
# Выполнение лабораторной работы

1. Создаю необходимые файлы с помощью команды *touch*.(рис. [-@fig:001])

создание файлов

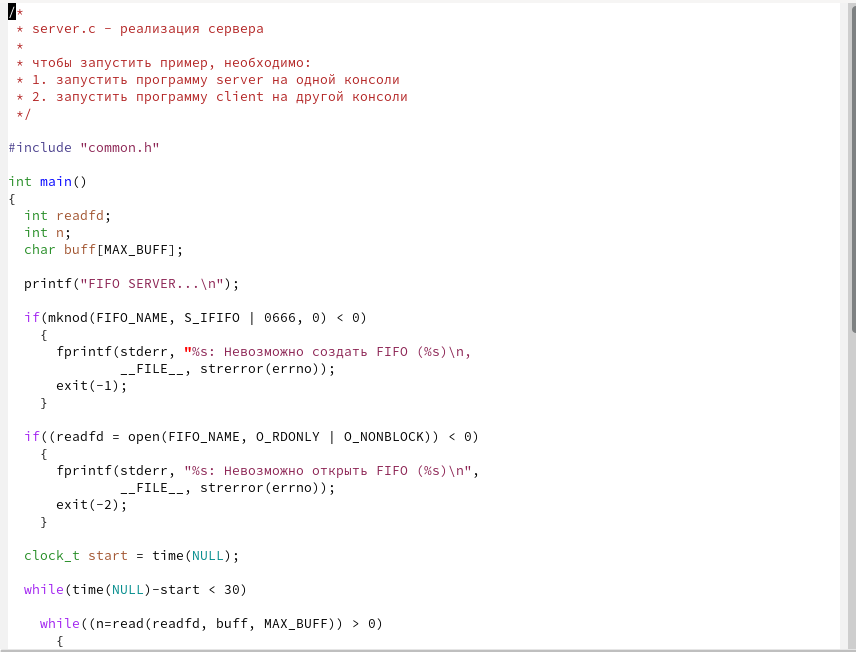
создание файлов

1. В файл *common.h* добавляю стандартные заголовочные файлы *time.h* и *unistd.h* (рис. [-@fig:002])



common.h

1. В файл *server.c* добавляю цикл while для контроля за временем работы сервера. Разница между текущим временем и временем начала работы не должна превышать 30 секунд. (рис. [-@fig:003]) (рис. [-@fig:004])

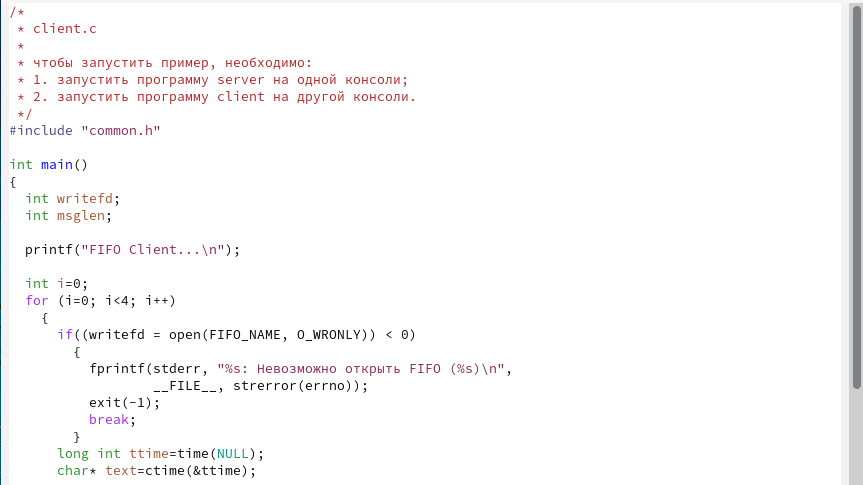


server.c

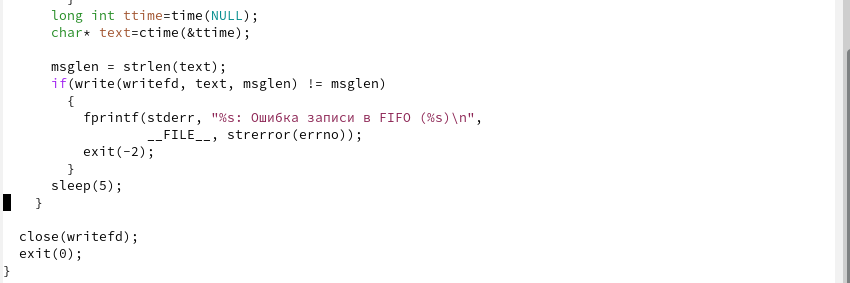


server.c

1. В файл *client.c* добавляю цикл, который отвечает за количество сообщений о текущем времени и команду sleep(5) для приостановки работы клиента на 5 секунд. (рис. [-@fig:005]) (рис. [-@fig:006])



client.c



client.c

1. Файл для сборки - Makefile - не изменяю. (рис. [-@fig:007])



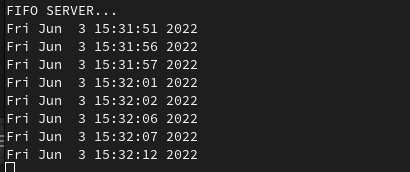
Makefile

1. Выполяню команду make all и компилирую необходимые файлы. (рис. [-@fig:008])

компиляция

компиляция

1. Проверяю работу написанного кода. Открываю 3 консоли и запускаю в одном из них *./server*(рис. [-@fig:009]) , а в двух других - *./client*. (рис. [-@fig:010]) (рис. [-@fig:011])



./server

./client

./client

./client

./client

1. Если сервер завершит работу, не закрыв канал, то при следующей попытке запуска будет выдана ошибка “Невозможно создать FIFO”.

# Выводы

Выполняя данную лабораторную работу я приобрела практические навыки работы с именованными каналами.

# Список литературы