Отчёт по лабораторной работе

Именованные каналы

Грузинова Елизавета Константиновна

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. Что будет в случае, если сервер завершит работу, не закрыв канал?

# 3 Теоретическое введение

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. [1]

# 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). Также привести необходимые для работы тексты будущих командных файлов. (рис. 1-4)

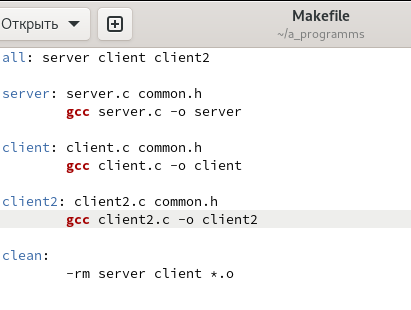


Рис. 1: Текст Makefile

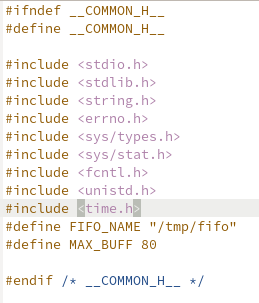


Рис. 2: Текст common.h

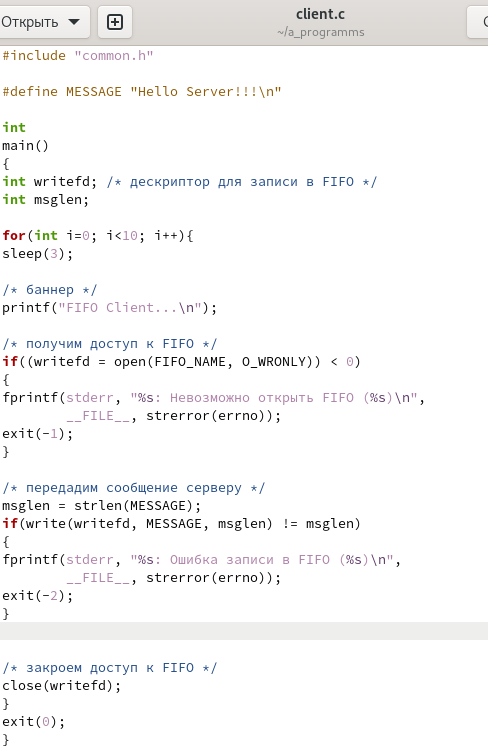


Рис. 3: Текст будущей программы client

Рис. 4: Работа программы client при запущенном сервере

Рис. 4: Работа программы client при запущенном сервере

1. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Используйте функцию sleep() для приостановки работы клиента. (рис. 5, 6)

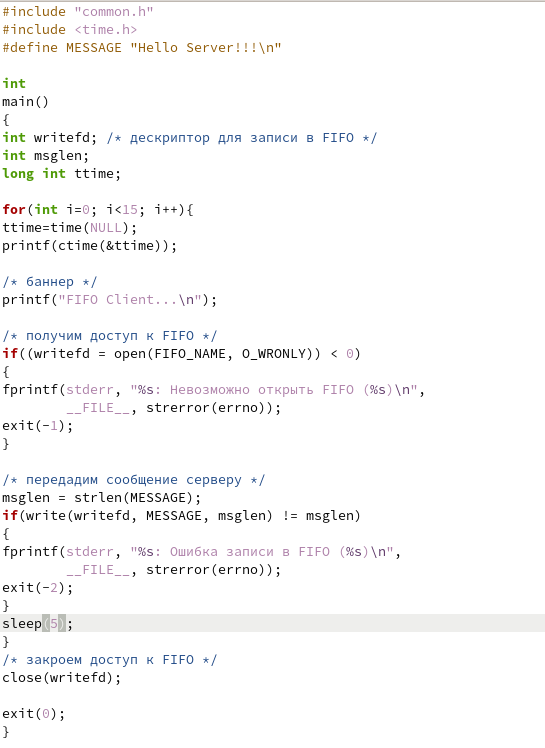


Рис. 5: Текст будущей программы client2

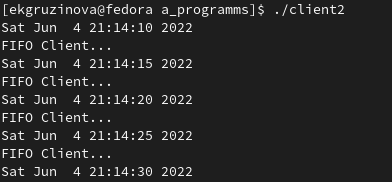


Рис. 6: Работа программы client2 при запущенном сервере

1. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию clock() для определения времени работы сервера. (рис. 7-9)



Рис. 7: Изменения в файле server.c(1)

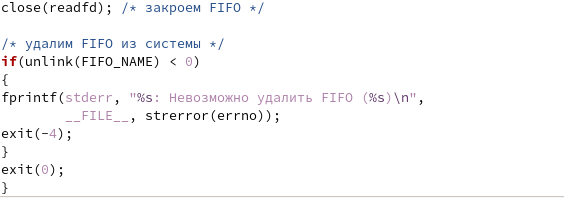


Рис. 8: Изменения в файле server.c(1)

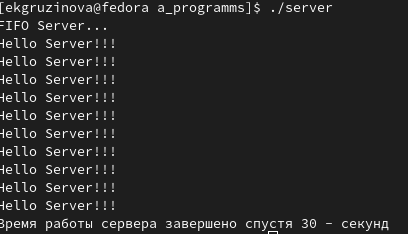


Рис. 9: Работа программы server

Если сервер завершит работу, не закрыв канал, то файл FIFO не исчезнет, вследствие чего его нельзя будет запустить во второй раз, что приведет к недееспособности сервера.

# 5 Выводы

В процессе выполнения лабораторной работы я приобрела практические навыки работы с именованными каналами.

# 6 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).

1. 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

1. Возможно ли создание именованного канала из командной строки?

Для создания файла FIFO можно использовать более общую функцию mknod(2), пред- назначенную для создания специальных файлов различных типов (FIFO, сокеты, файлы устройств и обычные файлы для хранения данных). 1 #include <sys/types.h> 2 #include <sys/stat.h> 3 #include <fcntl.h> 4 #include <unistd.h> 5 6 int mknod(const char \*pathname, mode\_t mode, dev\_t dev);

1. 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).

1. 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);

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

При чтении меньшего числа байтов, чем находится в канале, возвращается требуемое число байтов, остаток сохраняется для последующих чтений. При чтении большего числа байтов, чем находится в канале или FIFO возвращается доступное число байтов.

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

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. 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

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

В общем случае возможна многонаправленная работа процессов с каналом, т.е. возможна ситуация, когда с одним и тем же каналом взаимодействуют два и более процесса, и каждый из взаимодействующих каналов пишет и читает информацию в канал. Но традиционной схемой организации работы с каналом является однонаправленная организация, когда канал связывает два, в большинстве случаев, или несколько взаимодействующих процесса, каждый из которых может либо читать, либо писать в канал.

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

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

1. 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. Ошибки эти возникают при вызове функций стандартных Си-библиотек.

# Список литературы

1. Лабораторная работа No 14. Именованные каналы [Электронный ресурс]. URL: <https://esystem.rudn.ru/pluginfile.php/1383193/mod_resource/content/3/014-ipc-fifo.pdf>.