Операционные системы

Лабораторная работа №14

Матюшкин Денис Владимирович (НПИбд-02-21)

Содержание

# 1 Цель работы

* Приобретение практических навыков работы с именованными каналами.

# 2 Ход работы

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

* Работает не 1 клиент, а несколько (например, два).
* Клиенты передают текущее время с некоторой периодичностью (раз в пять секунд). Используем функцию sleep() для приостановки работы клиента.
* Сервер работает не бесконечно, а прекращает работу через некоторое время (30 сек). Используем не функцию clock(), а взятие unixTime (time(NULL)). Дело в том, что clock() не засчитывает такты sleep. С методом time() работает все хорошо. Если сервер завершит свою работу, не закрыв канал, то при повторном запуске сервера будет выводить ошибку: “Невозможно создать FIFO”.
* Измененный файл server.c (рис. 1 и рис. 2).  
  Измененный файл client.c (рис. 3 и рис. 4).  
  Файл common.h (рис. 5).  
  Файл Makefile (рис. 6).

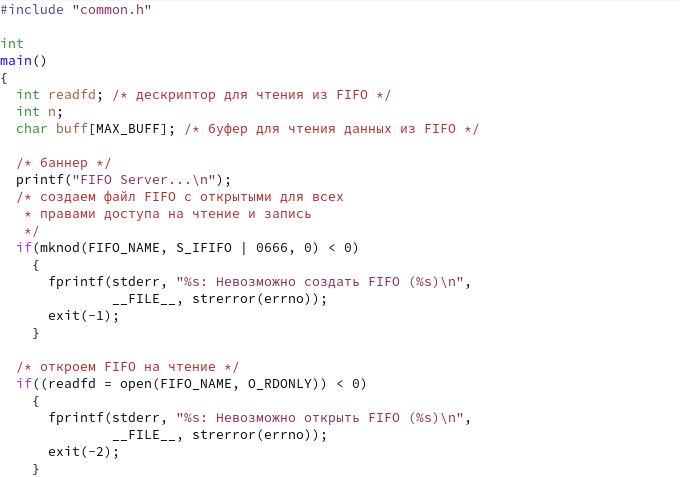


Рис. 1: Измененный server.c файл ч.1

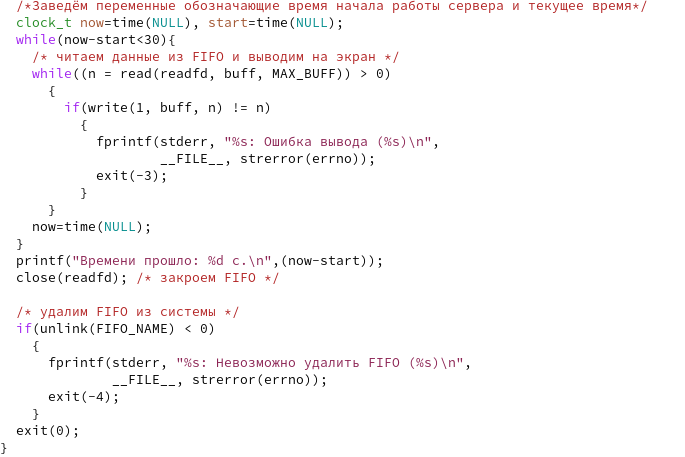


Рис. 2: Измененный server.c файл ч.2

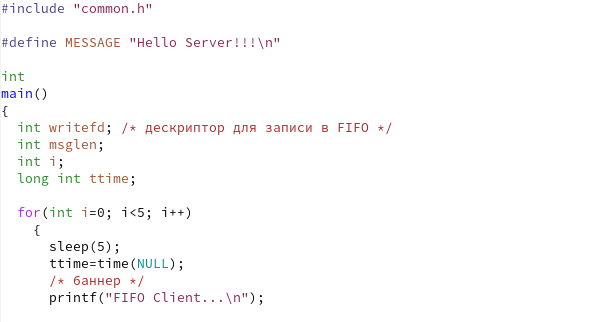


Рис. 3: Измененный client.c файл ч.1

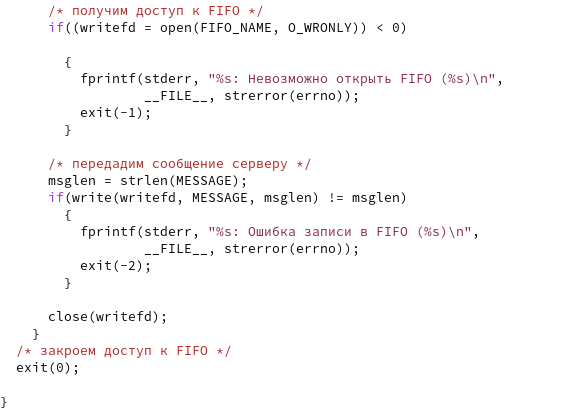


Рис. 4: Измененный client.c файл ч.2

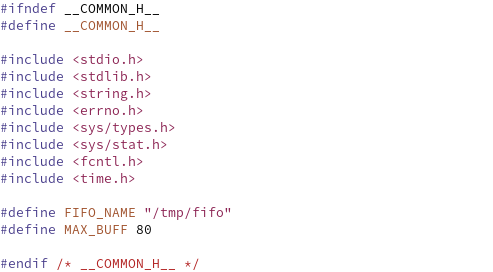


Рис. 5: Файл common.h

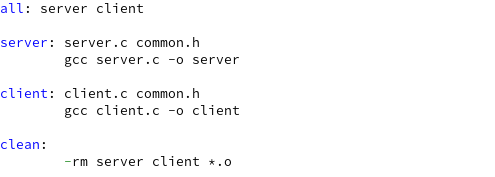


Рис. 6: Файл Makefile

1. Пропишем make в консоль, в итоге у нас должны создаться исполняемые файлы (рис. 7).

Рис. 7: Использование Makefile

Рис. 7: Использование Makefile

1. Проверим работу файлов (рис. 8 и рис. 9).

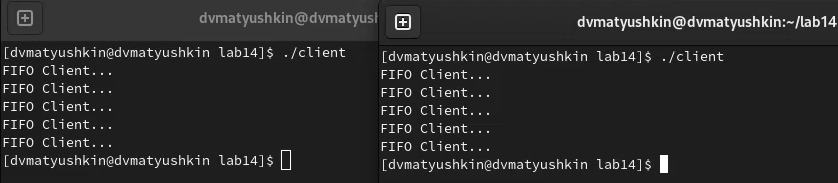


Рис. 8: Проверка client.c файла

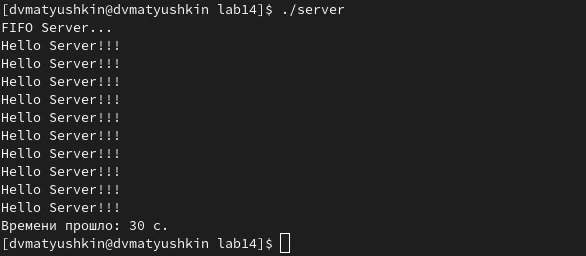


Рис. 9: Проверка server.c файла

# 3 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.

# 4 Вывод

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