

Лабораторная работа №14.

Именованные каналы

Козлов Всеволод Павлович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Контрольные вопросы	11
4	Выводы	13
	Список литературы	14

Список иллюстраций

2.1	client.c	6
2.2	client2.c	7
2.3	common.h	7
2.4	server.c	8
2.5	Makefile	9
2.6	Запуск команды make	9
2.7	Запуск исполняемых файлов	10

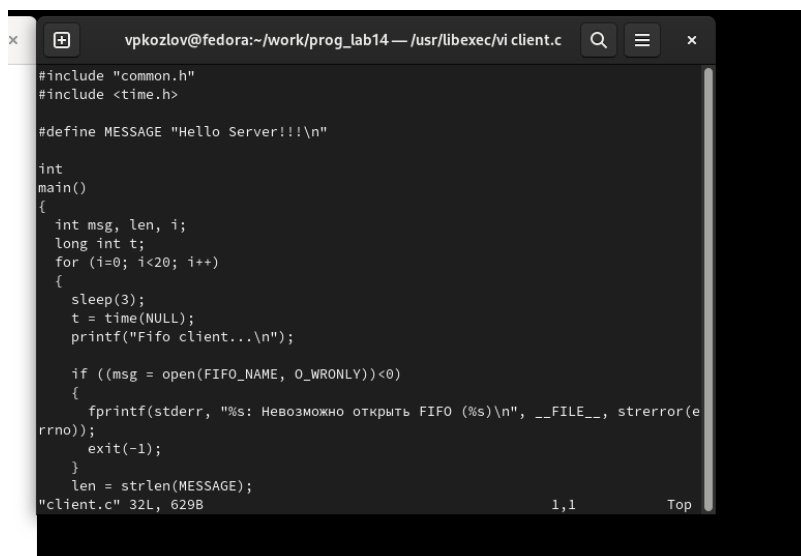
Список таблиц

1 Цель работы

Приобретение практических навыков работы с именованными каналами.

2 Выполнение лабораторной работы

Создал файл client.c (рис. [2.1])

A screenshot of a terminal window with a dark background. The window title is 'vpkozlov@fedora:~/work/prog_lab14 — /usr/libexec/vi client.c'. The code displayed is a C program named client.c. It includes 'common.h' and 'time.h', and defines a message 'Hello Server!!!\n'. The main function contains a loop that runs 20 times, sleeping for 3 seconds each iteration, and attempts to open a FIFO named 'FIFO_NAME'. If the opening fails, it prints an error message and exits with -1. Otherwise, it prints the length of the message. The status bar at the bottom shows 'client.c' 32L, 629B, 1,1, and a 'Top' button.

```
#include "common.h"
#include <time.h>

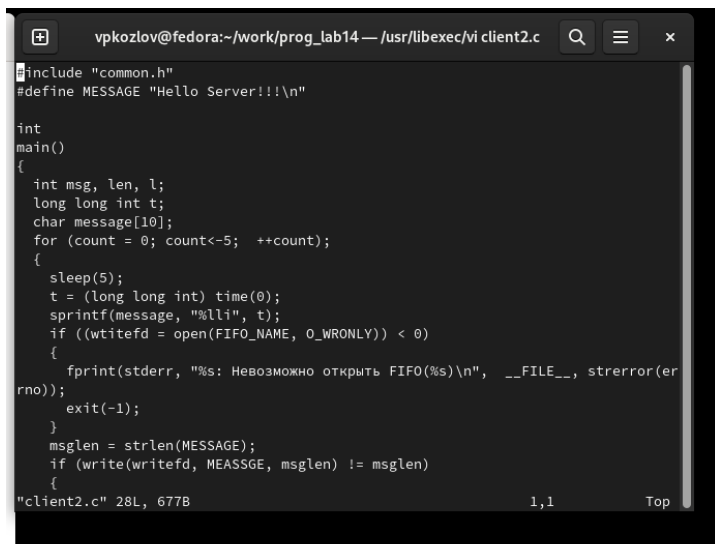
#define MESSAGE "Hello Server!!!\n"

int
main()
{
    int msg, len, i;
    long int t;
    for (i=0; i<20; i++)
    {
        sleep(3);
        t = time(NULL);
        printf("Fifo client...\n");

        if ((msg = open(FIFO_NAME, O_WRONLY))<0)
        {
            fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n", __FILE__, strerror(errno));
            exit(-1);
        }
        len = strlen(MESSAGE);
    }
}
```

Рис. 2.1: client.c

Создал client2.c (рис. [2.2])

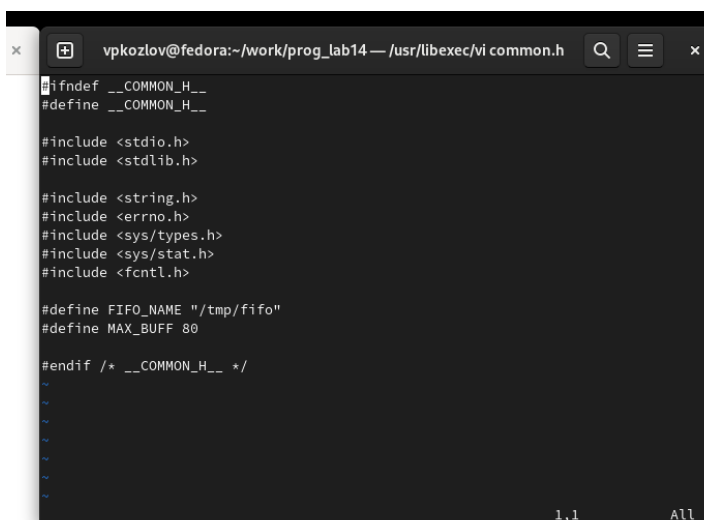


```
vpkozlov@fedora:~/work/prog_lab14 — /usr/libexec/vi client2.c
#include "common.h"
#define MESSAGE "Hello Server!!!\n"

int
main()
{
    int msg, len, l;
    long long int t;
    char message[10];
    for (count = 0; count<5; ++count);
    {
        sleep(5);
        t = (long long int) time(0);
        sprintf(message, "%lli", t);
        if ((wtfid = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr, "%s: Невозможно открыть FIFO(%s)\n", __FILE__, strerror(errno));
            exit(-1);
        }
        msglen = strlen(MESSAGE);
        if (write(wtfid, MESSAGE, msglen) != msglen)
        {
            fprintf(stderr, "%s: Невозможно записать в FIFO(%s)\n", __FILE__, strerror(errno));
            exit(-1);
        }
    }
}
```

Рис. 2.2: client2.c

Создал common.h (рис. [2.3])



```
vpkozlov@fedora:~/work/prog_lab14 — /usr/libexec/vi common.h
#ifndef __COMMON_H__
#define __COMMON_H__

#include <stdio.h>
#include <stdlib.h>

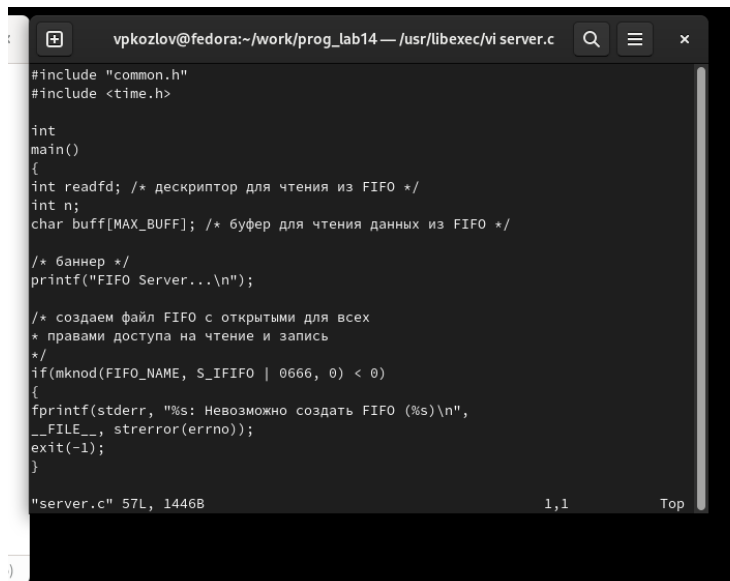
#include <string.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

#define FIFO_NAME "/tmp/fifo"
#define MAX_BUFF 80

#endif /* __COMMON_H__ */
```

Рис. 2.3: common.h

Создал server.c (рис. [2.4])



```
#include "common.h"
#include <time.h>

int
main()
{
    int readfd; /* дескриптор для чтения из FIFO */
    int n;
    char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */

    /* баннер */
    printf("FIFO Server...\n");

    /* создаем файл FIFO с открытыми для всех
     * правами доступа на чтение и запись
     */
    if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
    {
        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
            __FILE__, strerror(errno));
        exit(-1);
    }

    "server.c" 57L, 1446B 1,1 Top
```

Рис. 2.4: server.c

Создал Makefile (рис. [2.5])

Запустил исполняемые файлы (рис. [2.7])

[illegible]

Рис. 2.7: Запуск исполняемых файлов

3 Контрольные вопросы

1. Именованные каналы отличаются от неименованных наличием идентификатора канала, который представлен как специальный файл (соответственно имя именованного канала — это имя файла).
2. Создание неименованного канала из командной строки возможно командой `pipe`.
3. Создание именованного канала из командной строки возможно с помощью `mkfifo`.
4. Функция языка C, создающая 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. Функция языка C, создающая 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. При чтении меньшего числа байтов, возвращается требуемое число байтов, остаток сохраняется для следующих чтений. При чтении большего числа байтов, возвращается доступное число байтов 7. Запись числа байтов, меньшего емкости канала или FIFO, гарантированно атомарно. Это означает, что в случае, когда несколько процессов одновременно записывают в канал, пор-

ции 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=EP1PE`) (если процесс не установил обработки сигнала SIGPIPE, производится обработка по умолчанию – процесс завершается).

7. Два и более процессов могут читать и записывать в канал.
8. Функция `write` записывает `length` байтов из буфера `buffer` в файл, определенный дескриптором файла `fd`. Эта операция чисто ‘двоичная’ и без буферизации. При единице возвращает действительное число байтов. Функция `write` возвращает число действительно записанных в файл байтов или -1 при ошибке, устанавливая при этом `errno`.
9. Строковая функция `strerror` - функция языков C/C++, транслирующая код ошибки, который обычно хранится в глобальной переменной `errno`, в сообщении об ошибке, понятном человеку.

4 Выводы

Приобрел практические навыки работы с именованными каналами.

Список литературы