

# **Лабораторная работа №14**

**Операционные системы**

Серёгина Ирина Андреевна

# Содержание

|          |                                       |           |
|----------|---------------------------------------|-----------|
| <b>1</b> | <b>Цель работы</b>                    | <b>3</b>  |
| <b>2</b> | <b>Задание</b>                        | <b>4</b>  |
| <b>3</b> | <b>Теоретическое введение</b>         | <b>5</b>  |
| <b>4</b> | <b>Выполнение лабораторной работы</b> | <b>6</b>  |
| <b>5</b> | <b>Выводы</b>                         | <b>11</b> |
|          | <b>Список литературы</b>              | <b>12</b> |

# **1 Цель работы**

Приобретение практических навыков работы с именованными каналами.

## 2 Задание

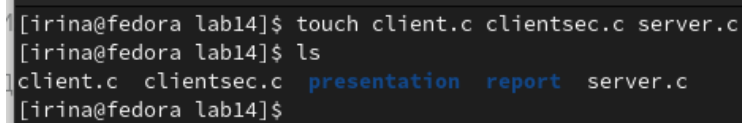
Изучите приведённые в тексте программы `server.c` и `client.c`. Взяв данные примеры за образец, напишите аналогичные программы, внося следующие изменения: 1. Работает не 1 клиент, а несколько (например, два). 2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). 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

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: общепонимание (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.

## 4 Выполнение лабораторной работы

1. Я открываю терминал и создаю файлы, которые понадобятся мне при выполнении лабораторной работы (рис. fig. 4.1).



```
[irina@fedora lab14]$ touch client.c clientsec.c server.c
[irina@fedora lab14]$ ls
client.c clientsec.c presentation report server.c
[irina@fedora lab14]$
```

Рис. 4.1: создание файлов

2. Программа 1 (рис. fig. 4.2).

```

* 2. запустить программу client на другой консоли.
*/

#include "common.h"

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

int
main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;

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

    for (int i; i<4; i++){

        /* получим доступ к FIFO */
        if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-1);
        }

        /* передадим сообщение серверу */
        msglen = strlen(MESSAGE);
        if(write(writefd, MESSAGE, msglen) != msglen)
        {
            fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-2);
        }

        sleep(5);

        /* закроем доступ к FIFO */
        close(writefd);

        exit(0);
    }
}

```

Рис. 4.2: код программы 1

### 3. Программа 2 (рис. fig. 4.3).

```

#include "common.h"
#include <time.h>
#define MESSAGE "Hello Server!!!\n"

int
main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;
    long int ttime;

    for (int i=0; i<15; i++){
        ttime=time(NULL);
        printf(ctime(&ttime));

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

        /* получим доступ к FIFO */
        if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-1);
        }
        /* передадим сообщение серверу */
        msglen = strlen(MESSAGE);
        if(write(writefd, MESSAGE, msglen) != msglen)
        {
            fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-2);
        }
        sleep(4)
    }
    /* закроем доступ к FIFO */
    close(writefd);

    exit(0);
}

```

Рис. 4.3: код программы 2

#### 4. Программа 3 (рис. fig. 4.4).



```

client.c

/* 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);
}

/* откроем FIFO на чтение */
if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
{
    fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
    __FILE__, strerror(errno));
    exit(-2);
}

clock_t beginning=time(NULL), clock_t now=time(NULL);
while (beginning-now<30){

    /* читаем данные из FIFO и выводим на экран */
    while((n = read(readfd, buff, MAX_BUFF)) > 0)
    {
        if(write(1, buff, n) != n)
        {
            fprintf(stderr, "%s: Ошибка вывода (%s)\n",
            __FILE__, strerror(errno));
            exit(-3);
        }
    }
    close(readfd); /* закроем FIFO */

    /* удалим FIFO из системы */
    if(unlink(FIFO_NAME) < 0)
    {
        fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
        __FILE__, strerror(errno));
        ...
    }
}

```

Рис. 4.4: код программы 3

5. После этого я создаю Makefile и заполняю его, все компилируется удачно (рис. fig. 4.5).

```
all: server client

server: server.c common.h
    gcc server.c -o server

client: client.c common.h
    gcc client.c -o client

clean:
    -rm server client *.o
```

Рис. 4.5: код Makefile

## **5 Выводы**

Я приобрела практические навыки работы с именованными каналами.

## **Список литературы**