

# Лабораторная работа №14

Именованные каналы

---

Захаренко А.В.

10 апреля 2023

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Захаренко Анастасия Викторовна
- студентка первого курса бакалавриата
- направление бизнес-информатика, факультет физико-математических и естественных наук
- Российский университет дружбы народов

## Элементы презентации

---

Приобретение практических навыков работы с именованными каналами.

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?

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: общедоступные (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



1. Я создала необходимые для работы файлы(common.h, server.c, client.c, Makefile)

```
[avzakharenko@fedora lab14]$ touch common.h server.c client.c Makefile  
[avzakharenko@fedora lab14]$ touch Makefile
```

Рис. 1: команды



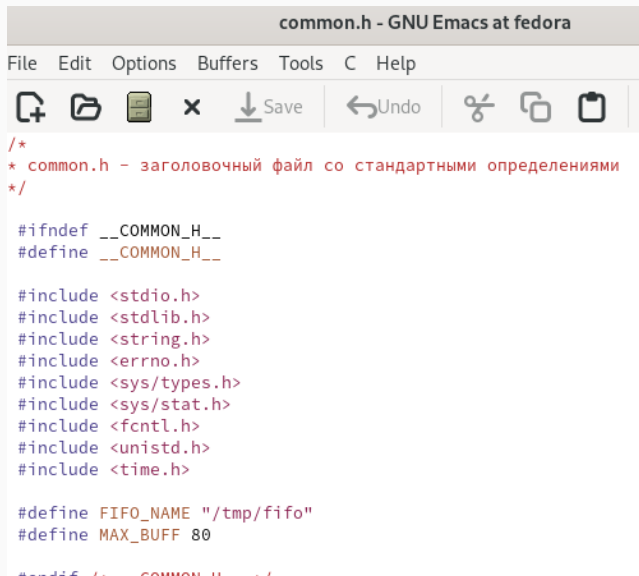
	client	25,5 кБ	21:19	☆
	client.c	1,2 кБ	21:13	☆
	common.h	426 байт	20:40	☆
	Makefile	156 байт	21:19	☆
	presentation	3 объекта	24 фев	☆
	report	5 объектов	24 фев	☆
	server	25,5 кБ	21:19	☆
	server.c	1,6 кБ	20:49	☆

Рис. 2: файлы

2. Далее я скопировала коды из лабораторной и дополнила их в соответствии с заданием.
3. common.h: добавила стандартные заголовочные файлы unistd.h, time.h.



```
/*
 * common.h - заголовочный файл со стандартными определениями
 */

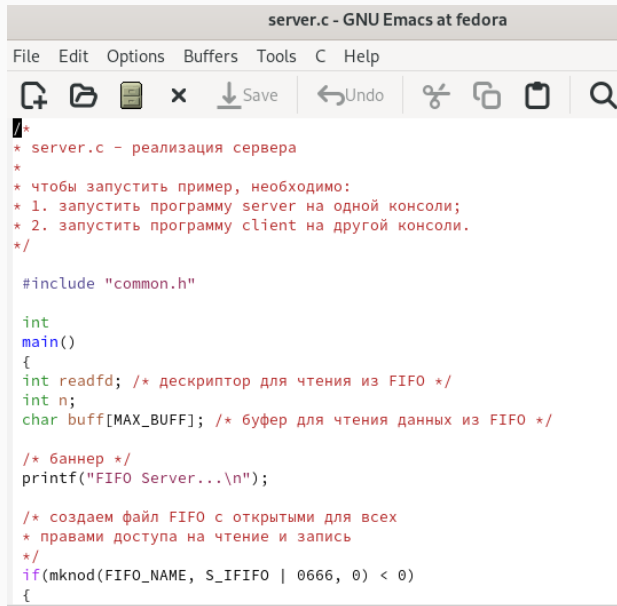
#ifndef __COMMON_H__
#define __COMMON_H__

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <time.h>

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

#endif
```

4. server.c: добавила цикл для контроля за временем работы сервера.



```
server.c - GNU Emacs at fedora
File Edit Options Buffers Tools C Help

[Icons: New, Open, Save, Close, Undo, Redo, Cut, Copy, Paste, Find]

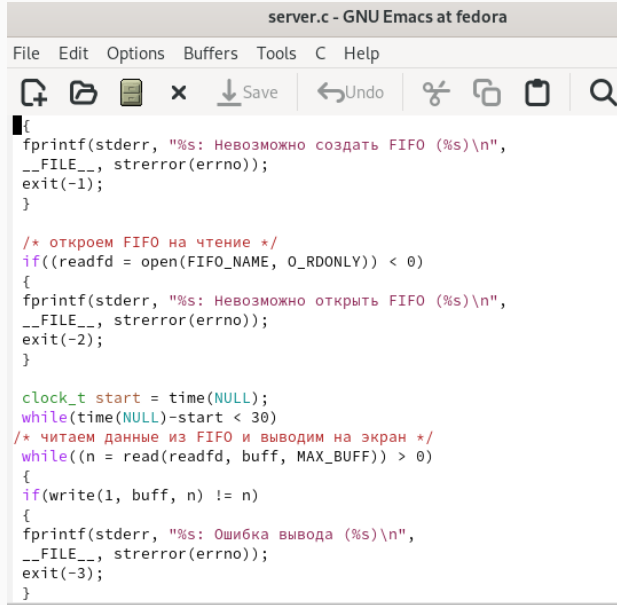
/*
 * server.c - реализация сервера
 *
 * чтобы запустить пример, необходимо:
 * 1. запустить программу server на одной консоли;
 * 2. запустить программу client на другой консоли.
 */

#include "common.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)
    {
```



```
server.c - GNU Emacs at fedora
File Edit Options Buffers Tools C Help

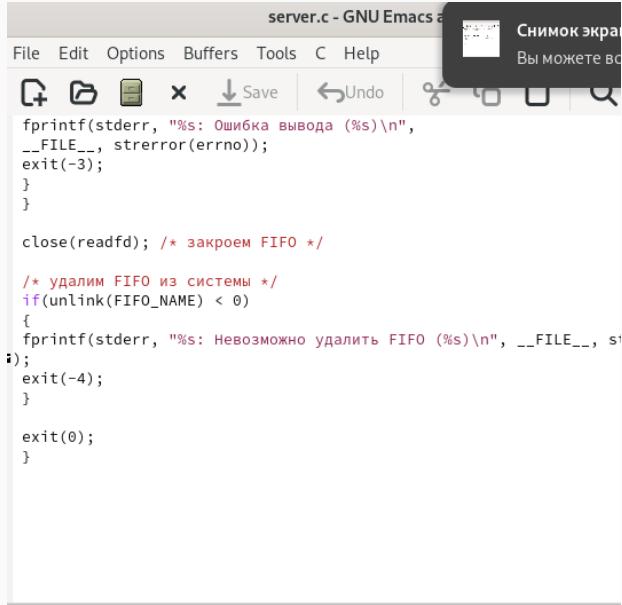
[Save] [Undo] [Redo] [Search]

{
    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 start = time(NULL);
while(time(NULL)-start < 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);
    }
}
```

Рис. 5: server.c



```
server.c - GNU Emacs a
File Edit Options Buffers Tools C Help
[Icons: Save, Undo, etc.]

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__, st
);
exit(-4);
}

exit(0);
}
```

Рис. 6: server.c

5. client.c: добавила цикл, отвечающий за кол-во сообщений о текущем времени.

```
#include "common.h"

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

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

    for(int i=0; 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);
        }

        long int ttime = time(NULL);
        char* text = ctime(&ttime);
        /* передадим сообщение серверу */
        msglen = strlen(text);
        if(write(writefd, text, msglen) != msglen)
```

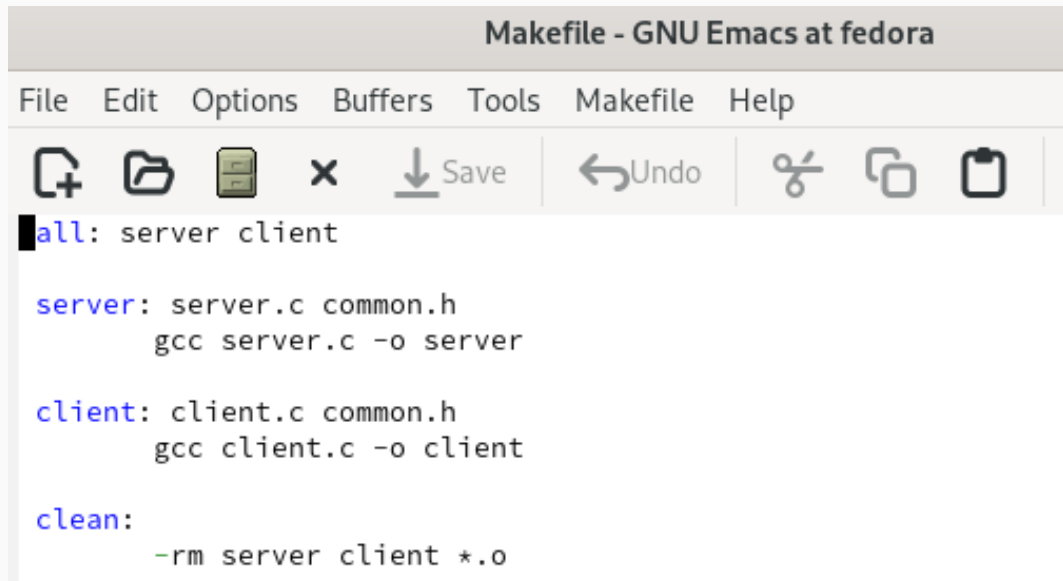
```
if(write(writefd, text, msglen) != msglen)
{
    fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n", __FILE__, strerror(errno));
    exit(-2);
}
sleep (5);
}

/* закроем доступ к FIFO */
close(writefd);
exit(0);
}
```

Рис. 8: client.c



6. Makefile: поставила таб вместо пробелов в 4 и тд. строке



The screenshot shows a GNU Emacs editor window titled "Makefile - GNU Emacs at fedora". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Makefile", and "Help". The toolbar contains icons for opening a file, saving, undo, redo, and other standard editing functions. The main text area displays a Makefile with the following content:

```
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
```

The cursor is positioned at the start of the first line, "all: server client".

7. Далее скомпилировала все файлы

```
[avzakharenko@fedora lab14]$ make all
Makefile:4: *** пропущен разделитель. Останов.
[avzakharenko@fedora lab14]$ make all
Makefile:4: *** пропущен разделитель (возможно нужен TAB вместо восьми пробелов?). Останов.
[avzakharenko@fedora lab14]$ make all
gcc server.c -o server
gcc client.c -o client
```

Рис. 10: make all

8. В конце я проверила работу сервера, используя команды: ./server, ./client

```
[avzakharenko@fedora lab14]$ ./server
FIFO Server...
Mon Apr 10 21:28:01 2023
Mon Apr 10 21:28:06 2023
Mon Apr 10 21:28:11 2023
Mon Apr 10 21:28:16 2023
[avzakharenko@fedora lab14]$
```

Рис. 11: ./server

```
[avzakharenko@fedora lab14]$ ./client  
FIFO Client...  
[avzakharenko@fedora lab14]$
```

Рис. 12: ./client

Я приобрела практические навыки работы с именованными каналами.