

# **Лабораторная работа №15**

**Именованные каналы.**

Сергей Михайлович Роман

# Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	10

## List of Tables

# List of Figures

3.1	Программы из примера . . . . .	7
3.2	Программы с изменениями . . . . .	8
3.3	Сервер-клиент-сервер . . . . .	8
3.4	с функцией sleep() . . . . .	9
3.5	Работа . . . . .	9

# **1 Цель работы**

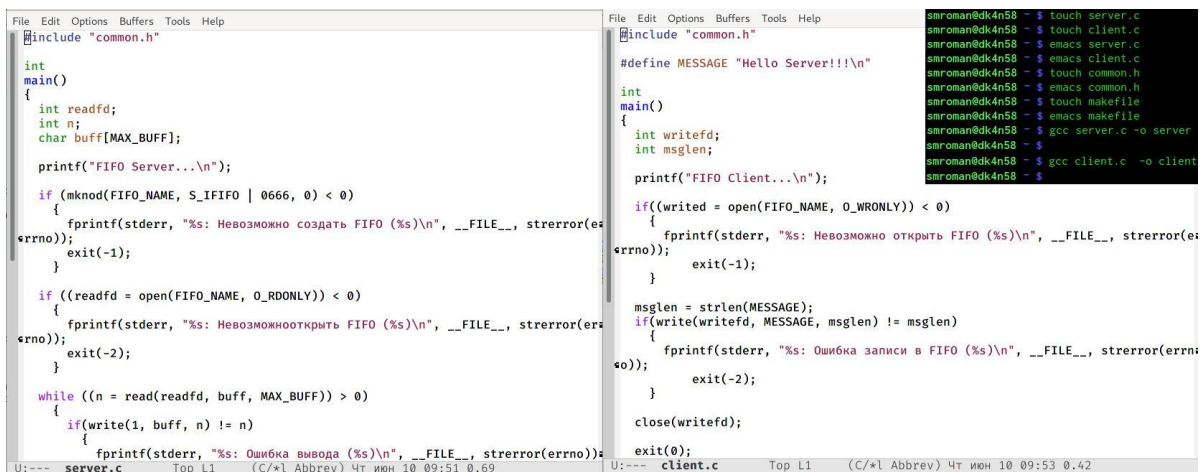
Приобретение практических навыков работы с именованными каналами.

## 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 Выполнение лабораторной работы

1. Я изучил и написал приведённые в тексте программы server.c и client.c.(рис. - fig. 3.1):



```
server.c
#include "common.h"

int
main()
{
    int readfd;
    int n;
    char buff[MAX_BUFF];

    printf("FIFO Server...\n");

    if (mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
    {
        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n", __FILE__, strerror(errno));
        exit(-1);
    }

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

    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);
        }
    }
}

client.c
#include "common.h"

#define MESSAGE "Hello Server!!\n"

int
main()
{
    int writefd;
    int msglen;

    printf("FIFO Client...\n");

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

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

    close(writefd);

    exit(0);
}
```

```
sairoman@dk4n58 ~$ touch server.c
sairoman@dk4n58 ~$ touch client.c
sairoman@dk4n58 ~$ emacs server.c
sairoman@dk4n58 ~$ emacs client.c
sairoman@dk4n58 ~$ touch common.h
sairoman@dk4n58 ~$ touch makefile
sairoman@dk4n58 ~$ gcc server.c -o server
sairoman@dk4n58 ~$ gcc client.c -o client
sairoman@dk4n58 ~$
```

Figure 3.1: Программы из примера

2. Пишу программы с изменениями. Работают 2 клиента.(client.c уже написан с изменениями, а client\_2.c написан без изменений и передаёт сообщение серверу).(рис. - fig. 3.2):



Figure 3.2: Программы с изменениями

- Работа сервер-клиент-сервер:(рис. - fig. 3.3):



Figure 3.3: Сервер-клиент-сервер

- Переписал программу, где клиент работает с некоторой периодичностью, с использованием функции sleep(), а сервер прекращает работу через некоторое время. Также клиент передаёт серверу время работы.(рис. - fig. 3.4):



```

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

int
main()
{
    int writefd;
    int msglen;
    char message[10];
    int c;
    long long int T;

    for(c = 0; c<=5; ++c)
    {
        sleep(5);
        T = (long long int) time(0);
        sprintf(message, "%lli", T);
        message[9] = '\n';
        printf("FIFO Client...\n");
    }

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

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

    close(writefd);
    exit(0);
}

int
main()
{
    int readfd;
    int n;
    char buff[MAX_BUFF];

    printf("FIFO Server...\n");

    if (mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
    {
        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n", __FILE__, strerror(errno));
        exit(-1);
    }

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

    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);
        if(unlink(FIFO_NAME) < 0)
        {
            fprintf(stderr, "%s: Ошибка вывода (%s)\n", __FILE__, strerror(errno));
            exit(-4);
        }
        exit(0);
    }
}

```

Figure 3.4: с функцией sleep()

На последнем скриншоте видно, что клиент передаёт информацию серверу несколько раз, спустя некоторое время.(рис. - fig. 3.5):

```

~: bash — Konsole
saroman@dk4n58 ~$ ./client
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
saroman@dk4n58 ~$

~: bash — Konsole
saroman@dk4n58 ~$ ./server
FIFO Server...
158.490.395
158.490.395
158.490.395
158.490.395
158.490.395
158.490.395

```

Figure 3.5: Работа

## **4 Выводы**

Сегодня я приобрёл практические навыки работы с именованными каналами.