

## Лабораторная работа №15. Именованные каналы.

---

Гекишева Анастасия Дмитриевна, НБИ-01-20,

30 апреля, 2021

<sup>1</sup>RUDN University, Moscow, Russian Federation

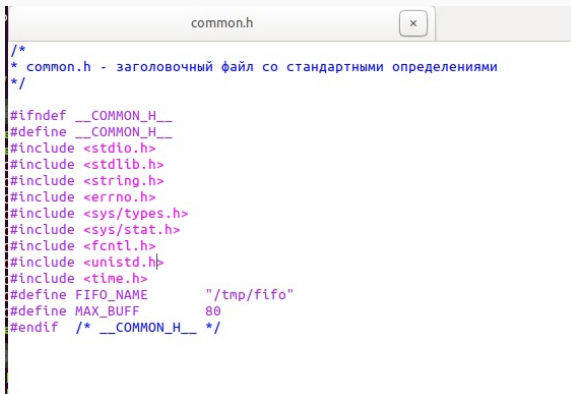
Приобретение практических навыков работы с именованными каналами.

Изучить приведённые в тексте программы `server.c` и `client.c` и напишите аналогичные программы, внося следующие изменения:

- Работает не 1 клиент, а несколько (например, два).
- Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Используйте функцию `sleep()` для приостановки работы клиента.
- 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?

## Выполнение лабораторной работы

---



```
common.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 /* __COMMON_H__ */
```

Рис. 1: Файл common.h

# Файл server.c: время работы 30 сек

```
Открыть  server.c  
~/work01 Сохранить
```

```
#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 */

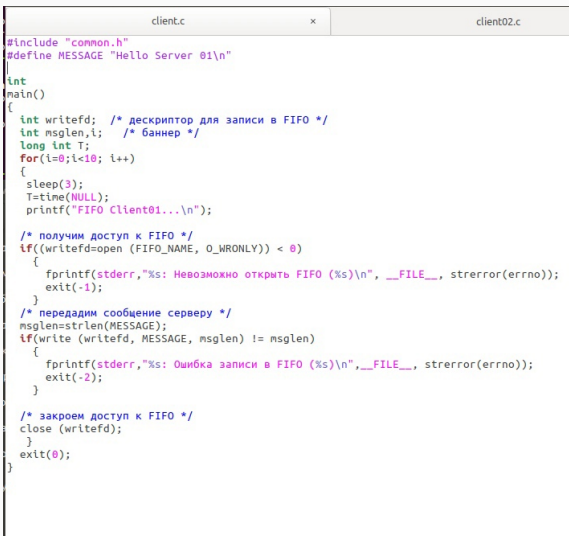
    printf("FIFO Server...\n");

    /* создаем файл FIFO с открытыми для всех
     * правами доступа на чтение и запись
     */

    if(mknod (FIFO_NAME, S_IFIFO | 0666, 0) < 0)
    {
        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 now=time(NULL), start=time(NULL);
    while((now-start)<30)
    {
        /* читаем данные из FIFO и выводим на экран */
        if ((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);
            }
        }
    }
    now=time(NULL);
}
printf("\n-----\nСервер закончил работу, прошло 30 секунд\n-----\n");
close(readfd);

/* закроем FIFO */ /* удалим FIFO из системы */
if(unlink(FIFO_NAME) < 0)
{
    fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n", __FILE__, strerror(errno));
    exit(-4);
}
exit(0);
}
```



```
client.c      x      client02.c

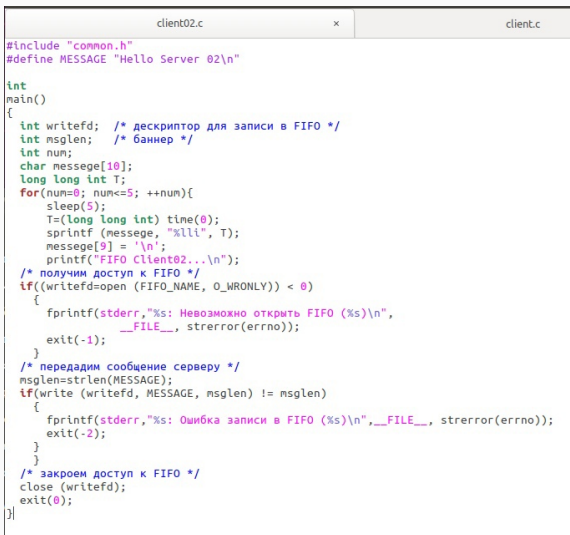
#include "common.h"
#define MESSAGE "Hello Server 01\n"

int
main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen,i; /* баннер */
    long int T;
    for(i=0;i<10; i++)
    {
        sleep(3);
        T=time(NULL);
        printf("FIFO Client01...\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);
        }

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

Рис. 3: Файл client.c



```
client02.c      x      client.c

#include "common.h"
#define MESSAGE "Hello Server 02\n"

int
main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;  /* баннер */
    int num;
    char messege[10];
    long long int T;
    for(num=0; num<=5; ++num){
        sleep(5);
        T=(long long int) time(0);
        sprintf (messege, "%lli", T);
        messege[9] = '\n';
        printf("FIFO client02...\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);
        }
        /* закроем доступ к FIFO */
        close (writefd);
        exit(0);
    }
}
```

Рис. 4: Файл client02.c



## Makefile и компиляция файлов

```
o
r
o
r
n
o
p
r
w
Makefile
all: server client02 client
server: server.c common.h
    gcc server.c -o server
client: client.c client02.c common.h
    gcc client.c -o client
    gcc client02.c -o client02
clean:
    -rm server client *.o

adkekisheva@adkekisheva-VirtualBox:~/work01$ make
gcc server.c -o server
gcc client.c -o client
gcc client02.c -o client02
adkekisheva@adkekisheva-VirtualBox:~/work01$ ./server
5150 Server
```

## Результат работы программы 1

```
adkekisheva@adkekisheva-VirtualBox:~/work01$ ./server
FIFO Server...
Hello Server 02
Hello Server 02
Hello Server 02
Hello Server 02
Hello Server 01
Hello Server 01
Hello Server 02
Hello Server 01
Hello Server 02
Hello Server 01
Hello Server 01
Hello Server 01
Hello Server 01
.....
Сервер закончил работу, прошло 30 секунд
.....
adkekisheva@adkekisheva-VirtualBox:~/work01$
```

Рис. 5: Результат работы программы 1

# Файл server.c: время работы 20 сек

```
client02.c  x  server.c  x  common.h  x  cl

#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");

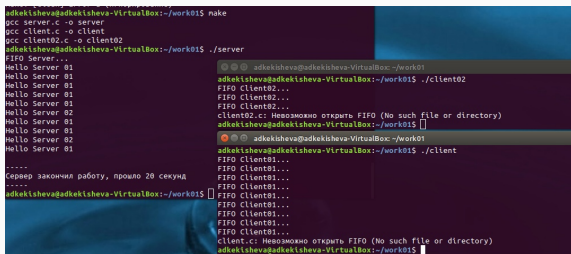
    /* создаем файл FIFO с открытыми для всех
     * правами доступа на чтение и запись
     */

    if(mknod (FIFO_NAME, S_IFIFO | 0666, 0) < 0)
    {
        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 now=time(NULL), start=time(NULL);
    while((now-start)<20)
    {
        /* читаем данные из FIFO и выводим на экран */
        if ((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);
            }
        }
    }
    now=time(NULL);
    printf("\n-----\nСервер закончил работу, прошло 20 секунд\n-----\n");
    close(readfd);

    /* закроем FIFO */ /* удалим FIFO из системы */
    if(unlink(FIFO_NAME) < 0)
    {
        fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n", __FILE__, strerror(errno));
        exit(-4);
    }
    exit(0);
}
```

## Результат работы программы 2



```
adkeki sheva@adkeki sheva-VirtualBox:~/work01$ make
gcc server.c -o server
gcc client.c -o client
gcc client02.c -o client02
adkeki sheva@adkeki sheva-VirtualBox:~/work01$ ./server
FIFO Server...
Hello Server 01
Hello Server 01
Hello Server 01
Hello Server 01
Hello Server 01
Hello Server 01
Hello Server 02
Hello Server 01
-----
Сервер закончил работу, прошло 20 секунд
-----
adkeki sheva@adkeki sheva-VirtualBox:~/work01$
adkeki sheva@adkeki sheva-VirtualBox:~/work01$ ./client02
FIFO Client02...
FIFO Client02...
FIFO Client02...
client02.c: Невозможно открыть FIFO (No such file or directory)
adkeki sheva@adkeki sheva-VirtualBox:~/work01$
adkeki sheva@adkeki sheva-VirtualBox:~/work01$ ./client
FIFO Client01...
FIFO Client01...
FIFO Client01...
FIFO Client01...
FIFO Client01...
FIFO Client01...
FIFO Client01...
FIFO Client01...
client.c: Невозможно открыть FIFO (No such file or directory)
adkeki sheva@adkeki sheva-VirtualBox:~/work01$
```

Рис. 7: Результат работы программы 2

Я приобрела практические навыки работы с именованными каналами, написала программы, с требующимися изменениями.

1. Ссылка 1