

# Laboratory №15

---

Krupennikova V.

MAY-2021

RUDN University, Moscow, Russian Federation

Приобретение практических навыков работы с именованными каналами.

1. Ознакомиться с теоретическим материалом.
2. Изучить основы программирования в оболочке ОС UNIX/Linux.
3. Выполнить упражнения.
4. Ответить на контрольные вопросы.

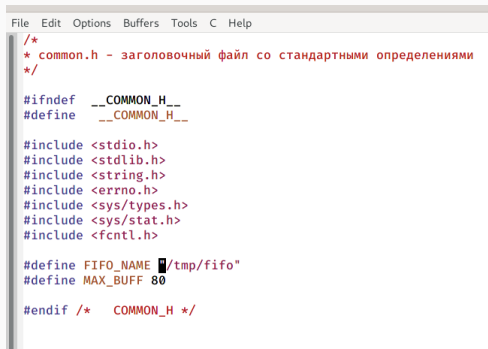
- 1) Для начала я создала необходимые файлы с помощью команды «touch common.h server.c client.c Makefile» и открыла редактор emacs для их редактирования.

```
vakrupennikova@dk6n66 ~ $ touch common.h server.c client.c Makefile  
vakrupennikova@dk6n66 ~ $ emacs &
```

Figure 1: Создание файлов

2) Далее я изменила коды программ, представленных в тексте лабораторной работы.

В файл common.h добавила стандартные заголовочные файлыunistd.h и time.h, необходимые для работы кодов других файлов. Common.h предназначен для заголовочных файлов, чтобы в остальных программах их не прописывать каждый раз.

A screenshot of a code editor window with a menu bar (File, Edit, Options, Buffers, Tools, C, Help) and a text area containing C header file code. The code defines \_\_COMMON\_H\_\_, includes standard headers, defines FIFO\_NAME and MAX\_BUFF, and ends with an ifdef block.

```
File Edit Options Buffers Tools C Help
/*
 * common.h - заголовочный файл со стандартными определениями
 */

#ifdef __COMMON_H__
#define __COMMON_H__

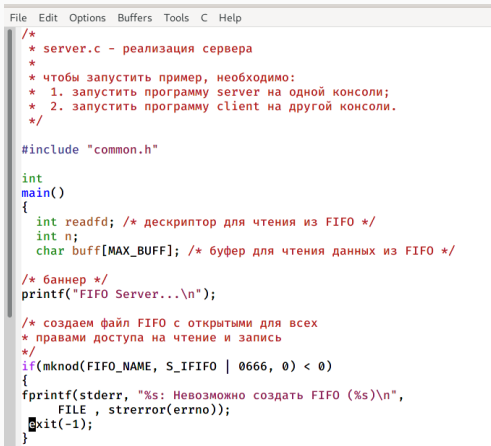
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

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

#endif /* COMMON_H */
```

Figure 2: Скрипт common.h

В файл server.c добавила цикл while для контроля за временем работы сервера. Разница между текущим временем time(NULL) и временем начала работы clock\_t start=time(NULL) (инициализация до цикла) не должна превышать 30 секунд.



```
File Edit Options Buffers Tools C Help
/*
 * 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)
    {
        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
            FILE , strerror(errno));
        exit(-1);
    }
}
```

Figure 3: Скрипт server.c

```

/* 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);
}

/*начало отсчета времени */
clock_t start=time(NULL);
/* цикл работает, пока с момента начала отсчета времени прошло менее 30 сек */
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);
        }
    }
    close(readfd); /* закроем FIFO */

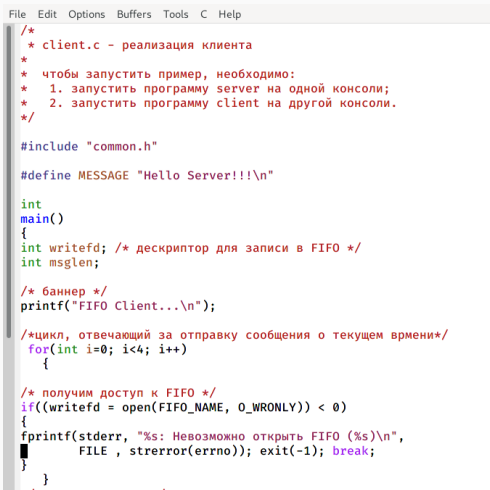
    /* удалим FIFO из системы */
    if(unlink(FIFO_NAME) < 0)
    {
        fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
            FILE , strerror(errno)); exit(-4);
    }
}

exit(0);
}

```

Figure 4: Скрипт server.c продолжение

В файл client.c добавила цикл, который отвечает за количество сообщений о текущем времени (4 сообщения), которое получается в результате выполнения команд (`/* текущее время */`) и команду `sleep(5)` для приостановки работы клиента на 5 секунд.



```
File Edit Options Buffers Tools C Help

/*
 * client.c - реализация клиента
 *
 * чтобы запустить пример, необходимо:
 * 1. запустить программу server на одной консоли;
 * 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");

    /*цикл, отвечающий за отправку сообщения о текущем времени*/
    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); break;
        }
    }
}
```

Figure 5: Скрипт client.c



```

/* текущее время */
long int ttime=time(NULL);
char* text=ctime(&ttime);

/* 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);
}
/* приостановка работы клиента на 5 сек*/
sleep(5);

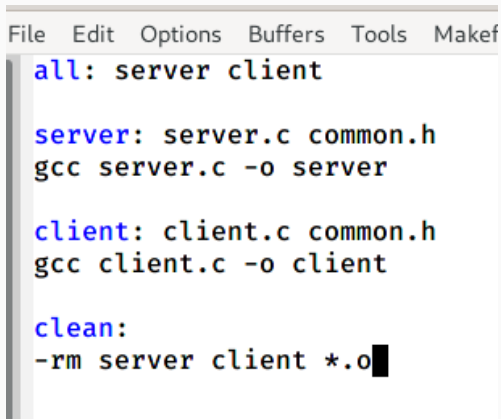
/* закроем доступ к FIFO */
close(writefd);

exit(0);
}

```

Figure 6: Скрипт client.c продолжение

Makefile (файл для сборки) не изменяла.

A screenshot of a text editor window with a menu bar containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', and 'Makef'. The editor displays a Makefile script 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:', and '-rm server client \*.o'. The text is color-coded: targets are in blue, prerequisites and commands are in black. A black cursor is positioned at the end of the last line.

```
File Edit Options Buffers Tools Makef
all: server client

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

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

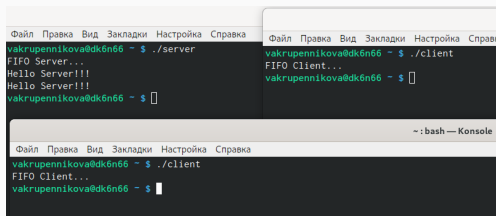
clean:
-rm server client *.o
```

Figure 7: Скрипт Makefile

3) После написания кодов я, используя команду «make all», скомпилировала необходимые файлы.

Далее я проверила работу написанного кода. Открыла 3 консоли (терминала) и запустила: в первом терминале – «./server», в остальных двух – «./client».

В результате каждый терминал - клиент вывел по 4 сообщения. Спустя 30 секунд работа сервера была прекращена. Программа работает корректно.



The image shows three terminal windows. The top-left window shows the execution of the server program, which outputs 'FIFO Server...', 'Hello Server!!!', and 'Hello Server!!!' before returning to the prompt. The top-right window shows the execution of the client program, which outputs 'FIFO Client...' and returns to the prompt. The bottom window shows the execution of the client program, which outputs 'FIFO Client...' and returns to the prompt.

```
Файл  Правка  Вид  Закладки  Настройка  Справка
vkrupennikova@dk6n66 ~ $ ./server
FIFO Server...
Hello Server!!!
Hello Server!!!
vkrupennikova@dk6n66 ~ $

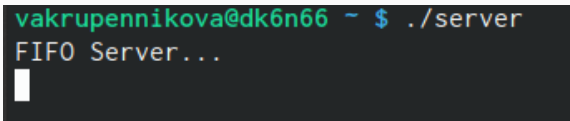
Файл  Правка  Вид  Закладки  Настройка  Справка
vkrupennikova@dk6n66 ~ $ ./client
FIFO Client...
vkrupennikova@dk6n66 ~ $

~ : bash — Konsole

Файл  Правка  Вид  Закладки  Настройка  Справка
vkrupennikova@dk6n66 ~ $ ./client
FIFO Client...
vkrupennikova@dk6n66 ~ $
```

Figure 8: Консоли

Также я отдельно проверила длительность работы сервера, введя команду «./server» в одном терминале. Он завершил свою работу через 30 секунд.

A terminal window with a dark background. The prompt is 'vakrupennikova@dk6n66 ~ \$' in green. The command './server' is entered in white. The output 'FIFO Server...' is shown in white. A white cursor is visible on the line below the output.

```
vakrupennikova@dk6n66 ~ $ ./server
FIFO Server...
█
```

Figure 9: Команда «./server»

Если сервер завершит свою работу, не закрыв канал, то, когда мы будем запускать этот сервер снова, появится ошибка «Невозможно создать FIFO», так как у нас уже есть один канал.

В ходе выполнения данной лабораторной работы я приобрела практические навыки работы с именованными каналами.

The end.