

Лабораторная работа №15

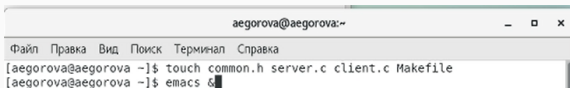
Егорова Александра¹

11 июня, 2021, Москва, Россия

¹RUDN University, Moscow, Russian Federation

Приобретение практических навыков работы с именованными каналами.

- 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 (рис. -fig. 1)

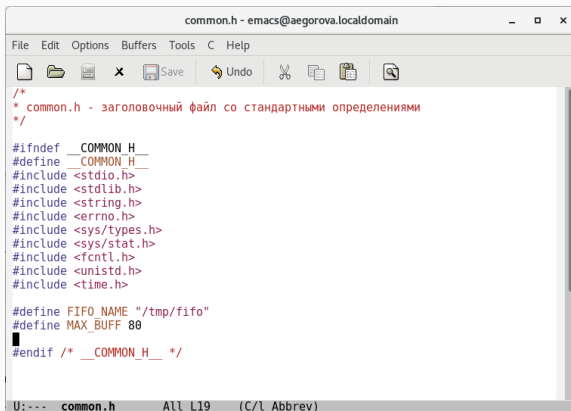


```
aegorova@aegorova:~  
Файл Правка Вид Поиск Терминал Справка  
[aegorova@aegorova ~]$ touch common.h server.c client.c Makefile  
[aegorova@aegorova ~]$ emacs &
```

Figure 1: Создание файлов

Выполнение лабораторной работы

- 2) Далее я изменила коды программ, представленных в тексте лабораторной работы. Common.h предназначен для заголовочных файлов, чтобы в остальных программах их не прописывать каждый раз. (рис. -fig. 2)



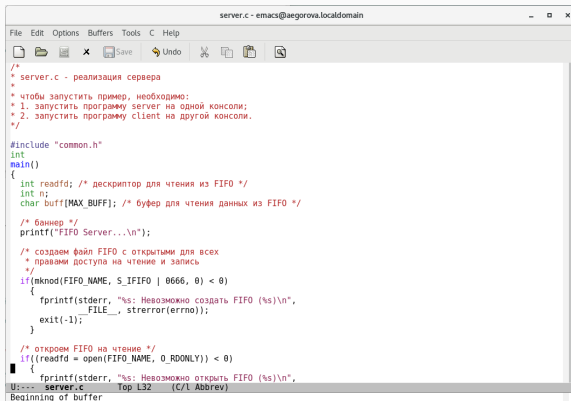
```
/*
 * common.h - заголовочный файл со стандартными определениями
 */

#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__ */
```

Выполнение лабораторной работы

В файл server.c добавила цикл while для контроля за временем работы сервера. (рис. -fig. 3)



```
server.c - emacs@aegerova.localdomain
File Edit Options Buffers Tools C Help
[Icons] Save Undo [Icons]

/*
 * 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);
    }

    /* откроем FIFO на чтение */
    if(!readfd = open(FIFO_NAME, O_RDONLY)) < 0)
    {
        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
                __FILE__, strerror(errno));
        exit(-1);
    }

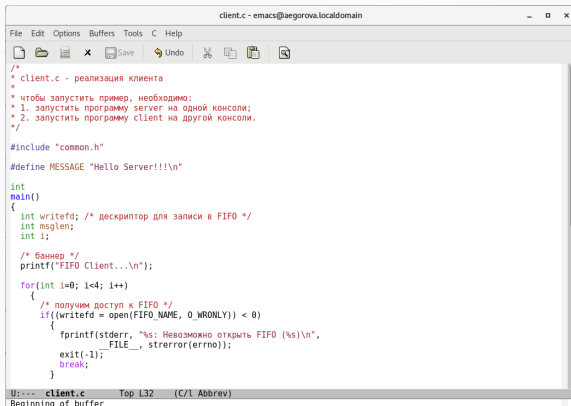
    while(1)
    {
        n = read(readfd, buff, MAX_BUFF);
        if(n < 0)
        {
            fprintf(stderr, "%s: Невозможно прочитать из FIFO (%s)\n",
                    __FILE__, strerror(errno));
            exit(-1);
        }
        buff[n] = '\0';
        printf("FIFO Server: %s\n", buff);
    }
}
```

U:--- server.c Top L32 (C/L Abbrev)
Beginning of buffer

Figure 3: server.c

Выполнение лабораторной работы

В файл client.c добавила цикл, который отвечает за количество сообщений о текущем времени. (рис. -fig. 4)



```
client.c - emacs@aegorova.localdomain

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;
    int i;

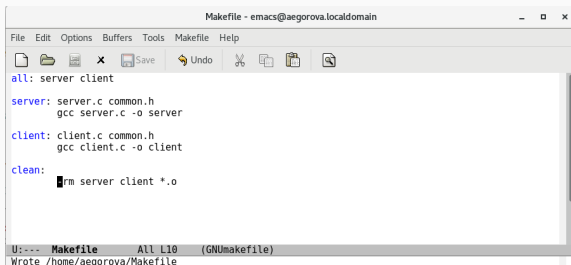
    /* баннер */
    printf("FIFO Client...\n");

    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);
            break;
        }
    }

    U:--- client.c      Top L32      (C/l Abbrev)
    Beginning of buffer
```

Figure 4: client.c

Makefile не изменяла. (рис. -fig. 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
```

U:--- **Makefile** All L10 (GNUmakefile)
Wrote /home/aegorova/Makefile

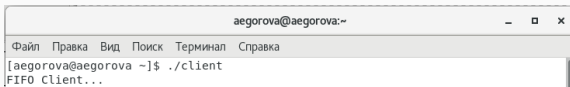
Figure 5: Makefile

- 3) После написания кодов, используя команду «make all», скомпилировала необходимые файлы. (рис. -fig. 6)

```
[aëgorova@aëgorova ~]$ make all  
gcc server.c -o server  
gcc client.c -o client
```

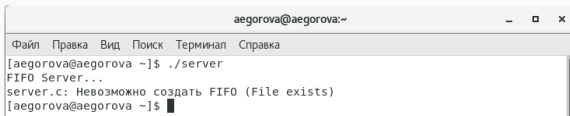
Figure 6: Компиляция файлов

Далее я проверила работу написанного кода. (рис. -fig. 7) (рис. -fig. 8)



```
aegorova@aegorova:~  
Файл Правка Вид Поиск Терминал Справка  
[aegorova@aegorova ~]$ ./client  
FIFO Client...
```

Figure 7: ./client



```
aegorova@aegorova:~  
Файл Правка Вид Поиск Терминал Справка  
[aegorova@aegorova ~]$ ./server  
FIFO Server...  
server.c: Невозможно создать FIFO (File exists)  
[aegorova@aegorova ~]$
```

Figure 8: Проверка кода

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