

# Отчет по лабораторной работе №14

---

Ничипорова Елена

26-05-22

РУДН, Москва

## Отчет

---

Приобретение практических навыков работы с именованными каналами.

- В начале я создала необходимые файлы и открыла их в редакторе emacs(рис. 1)

```
ednichiporova@dk3n53 ~$ touch common.h server.c client.c Makefile
ednichiporova@dk3n53 ~$ emacs &
[1] 15676
ednichiporova@dk3n53 ~$ emacs &
[2] 15630
[3] Завершён      emacs
ednichiporova@dk3n53 ~$ emacs common.h &
[3] 15668
[2] Завершён      emacs
ednichiporova@dk3n53 ~$ emacs server.c &
[4] 16537
ednichiporova@dk3n53 ~$ emacs client.c &
[5] 19218
[3] Завершён      emacs common.h
ednichiporova@dk3n53 ~$ emacs Makefile &
[6] 21631
[4] Завершён      emacs server.c
ednichiporova@dk3n53 ~$
```

Figure 1: Создание файлов

- 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 других файлов. Этот файл предназначен для заголовочных файлов, чтобы не прописывать их в других программах каждый раз. (рис. 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/pipe"  
#define MAX_BUFFER 80  
  
#endif /* __COMMON_H__ */
```

Figure 2: Программа в файле common.h

- В файле client.c добавила цикл, который отвечает за количество сообщений 0 текущем времени (4 сообщения), которое получается в результате выполнения команда Рисунке 7 и команду sleep(5) для приостановки работы клиента на 5 секунд (рис. 3)(рис. 4)

```
/*
 * client.c - реализация клиента
 *
 * чтобы запустить программу, необходимо:
 * 1. запустить серверную часть на одной консоли;
 * 2. запустить программу client на другой консоли;
 */

#include "common.h"
int
main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int readfd;

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

    for(int i=0; i<4; i++)
    {
        if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            perror("error: %s: Не удалось открыть FIFO (%s)",
                  _FILE_, strerror(errno));
            exit(-1);
        }

        long int t1=time(NULL);
        char *text = "time1time1";

        msglen = strlen(text);
        if(write(writefd, text, msglen) < msglen)
        {
            perror("error: %s: Не удалось записать в FIFO (%s)",
                  _FILE_, strerror(errno));
            exit(-1);
        }

        close(writefd);
    }
}
```

Figure 3: Программа в файле client.c

```
    sleep(5);
}
close (writefd);
exit(0);
}
```

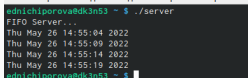
Figure 4: Программа в файле client.c

- Makefile, то есть файл для сборки, я оставила без изменения. После написания кодов, я использую команду “make all” для компиляции необходимых файлов(рис. 5)

```
make -f Makefile -> make all
cc client.c -o client
client.c:8: warning: implicit declaration of function 'atoi' is before its use
18 |     char host[atoi(atoi)];
   |     ^
client.c:18:18: warning: variable 'host' is used uninitialized
client.c:12:21: warning: 'atoi' is undeclared (first use in this function)
22 |     return atoi(host);
   |            ^
client.c:12:21: warning: 'atoi' is undeclared (first use in this function)
note: *** This file was generated by the 'make' command.
make: *** [client.o] Error 1
```

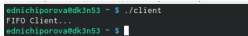
Figure 5: Команда make all

- Откроем три терминала. В первом окне запустим команду `./server.c`, во втором и третьем - `./client.c`. В результате каждый терминал-клиент вывел по четыре сообщения о текущем времени. Спустя 30 секунд работа сервера была прекращена. Если клиент завершит свою работу, не закрыв канал, то при повторном запуске сервера, появится ошибка “Невозможно создать FIFO”, так как уже существует один канал.(рис. 6)(рис. 7)

A terminal window with a black background and green text. The prompt is 'ednichiporova@dk3n53 ~ \$'. The command './server' has been executed. The output shows 'FIFO Server...' followed by four timestamps: 'Thu May 26 14:55:04 2022', 'Thu May 26 14:55:09 2022', 'Thu May 26 14:55:14 2022', and 'Thu May 26 14:55:19 2022'. The prompt is now 'ednichiporova@dk3n53 ~ \$' with a cursor.

```
ednichiporova@dk3n53 ~ $ ./server
FIFO Server...
Thu May 26 14:55:04 2022
Thu May 26 14:55:09 2022
Thu May 26 14:55:14 2022
Thu May 26 14:55:19 2022
ednichiporova@dk3n53 ~ $
```

Figure 6: Проверка работы

A terminal window with a black background and green text. The prompt is 'ednichiporova@dk3n53 ~ \$'. The command './client' has been executed. The output shows 'FIFO Client...' followed by a cursor.

```
ednichiporova@dk3n53 ~ $ ./client
FIFO Client...
ednichiporova@dk3n53 ~ $
```

Figure 7: Проверка работы



Я приобрела практические навыки работы с именованными каналами.