

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 15**

дисциплина: Операционные системы

Студент: Джунусова Рузель

Группа: НПИбд-01-20

МОСКВА

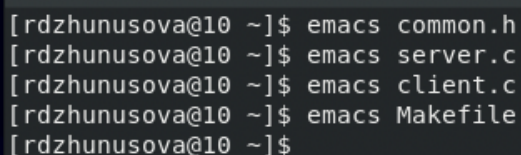
2021

Цель работы:

В данной лабораторной работе мне будет необходимо приобрести практические навыки работы с именованными каналами.

Выполнение лабораторной работы:

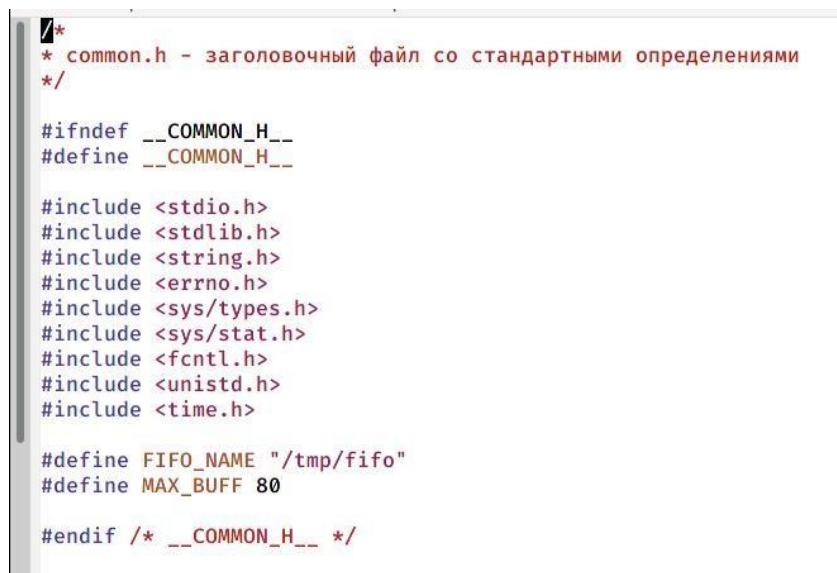
1. В домашнем каталоге создала необходимые файлы для работы. Использовала команду emacs. (рис. 4.1)



```
[rdzhunusova@10 ~]$ emacs common.h
[rdzhunusova@10 ~]$ emacs server.c
[rdzhunusova@10 ~]$ emacs client.c
[rdzhunusova@10 ~]$ emacs Makefile
[rdzhunusova@10 ~]$
```

Figure 4.1: Создание файлов

2. Отредактировала файл common.h - заголовочный файл со стандартными определениями. Добавила заголовочные файлы unistd.h и time.h. (рис. 4.2)



```
/*
 * 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__ */
```

Figure 4.2: Файл common.h

3. Добавила в файл server.c (реализация сервера) контроль за временем работы.

(рис. 4.3) (рис. 4.4)

```

/*
 * 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(-2);
    }

    clock_t start = time(NULL);
    while (time(NULL)-start<5)
    {
        /* 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 4.3: Файл server.c

```

        __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 4.4: Файл server.c

4. Добафилa в файл client.c (реализация клиента) цикл для вывода времени.

(рис. 4.5)

```

/*
 * 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");
    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;
        }
        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);
        }
        sleep(5);
    }
    /* закроем доступ к FIFO */
    close(writefd);
    exit(0);
}

```

Figure 4.5: Файл client.c

5. Файл Makefile не изменял. (рис. 4.6)

```
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 4.6: Файл Makefile

6. Запустила код в трёх консолях. В двух ./client и в одной ./server. Через 30 секунд работа программы была прекращена.
7. Если только в одном терминале запускать ./server, то появится ожидаемая ошибка.

Вывод:

В данной лабораторной работе мне успешно удалось приобрести практические навыки работы с именованными каналами.