

Лабораторная работа №15

Матвеева Анастасия Сергеевна НПМбд-02-20¹

10 июня, 2021, Москва, Россия

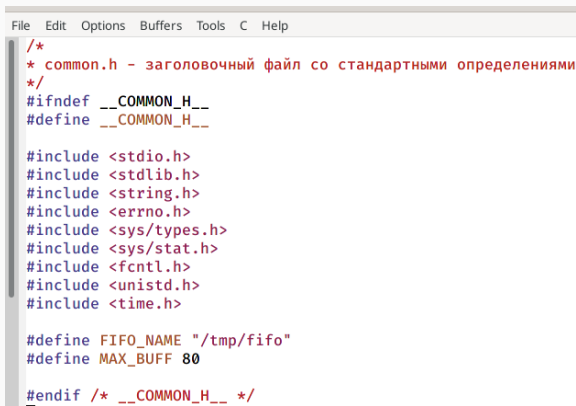
¹RUDN University, Moscow, Russian Federation

Перед нами стоит цель - приобретение практических навыков работы с именованными каналами, а также следующие задачи:

1. Познакомиться с механизмом именованных каналов и принципом FIFO.
2. Изучить данные в задании файлы.
3. В ходе работы изменить программы таким образом, чтобы выполнялись требуемые условия.
4. Сделать выводы и выполнить отчет.

Выполнение лабораторной работы

В ходе работы мы создали необходимые файлы. В файл `common.h` добавили стандартные заголовочные файлы `unistd.h` и `time.h`, необходимые для работы кодов других файлов. (рис. 1)

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 is for `common.h` and includes standard headers, defines `FIFO_NAME` and `MAX_BUFF`, and uses a conditional compilation guard.

```
/*
 * 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 1: Файл `common.h`

Выполнение лабораторной работы

В файл `server.c` добавили цикл `while` для контроля за временем работы сервера. Разница между текущим временем `time(NULL)` и временем начала работы `clock_t start=time(NULL)` не должна превышать 30 секунд. (рис. 2)

```
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);
    }

    /* откроем FIFO на чтение */
    if((readfd=open(FIFO_NAME, O_RDONLY))<0)
    {
        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
```

Выполнение лабораторной работы

В файл client.c добавила цикл, который отвечает за количество сообщений о текущем времени, и команду sleep(5) для приостановки работы клиента на 5 секунд. (рис. 3)

```
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"

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;
        }

        /* текущее время */
        long int ttime=time(NULL);
```

Далее проверили работу написанного кода. Отырли 3 консоли терминала и запустили: в первом терминале – «./server», в остальных двух – «./client». В результате каждый терминал-клиент вывел по 4 сообщения. Спустя 30 секунд работа сервера была прекращена. (рис. 4)

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