

Лабораторная работа №15. Именованные каналы

2021 г.

Выполнил: Маметкадыров Ынтымак

ФГАОУ ВО «Российский университет дружбы народов», Москва, Россия

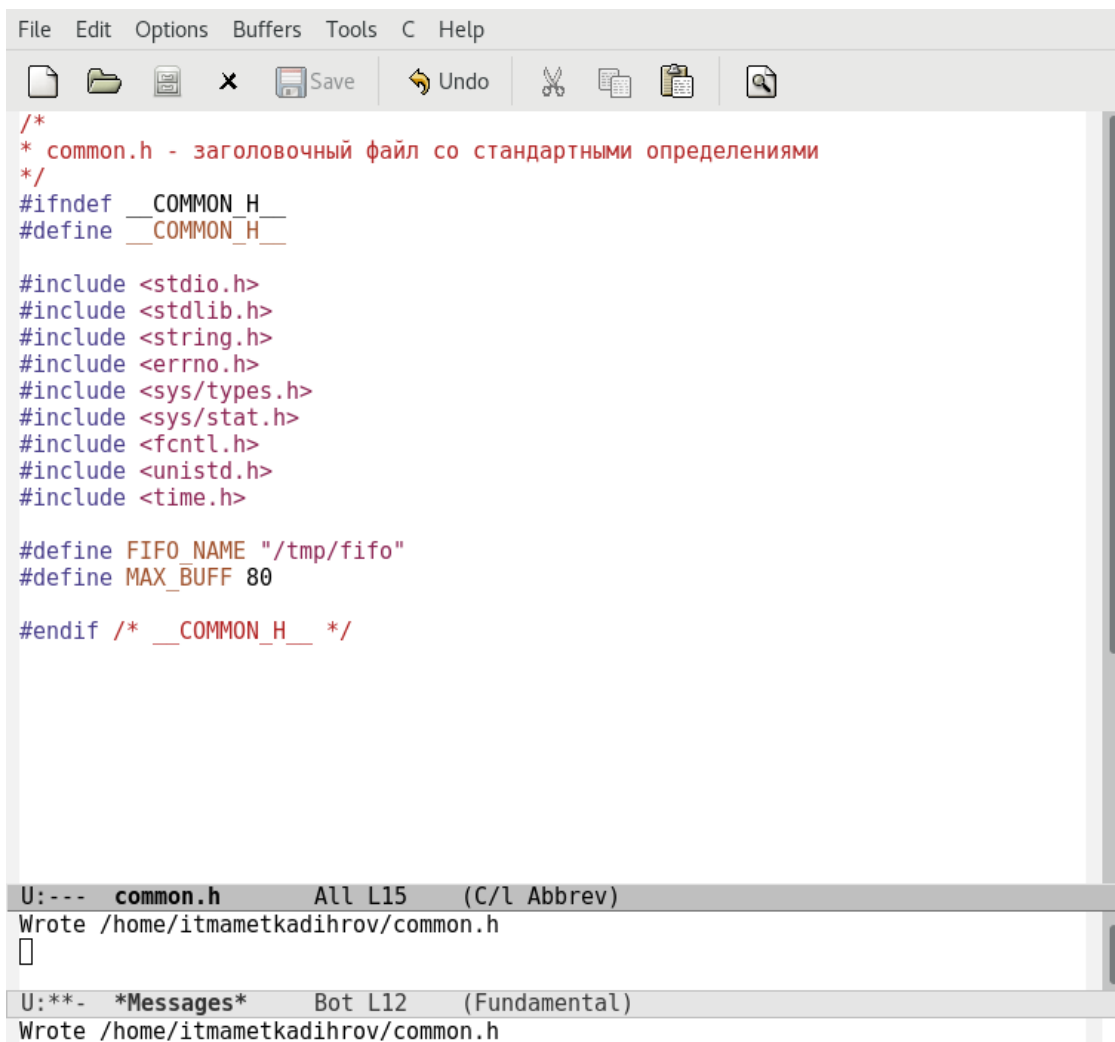
Цель работы

Приобретение практических навыков работы с именованными каналами.

Создали необходимые файлы

```
[itmametskadihrov@itmametskadihrov ~]$ touch common.h server.c client.c Makefile
[itmametskadihrov@itmametskadihrov ~]$ emacs &
[1] 3490
[itmametskadihrov@itmametskadihrov ~]$ █
```

Добавили заголовочный файлы unistd.h и time.h в файл common.h



```
File Edit Options Buffers Tools C Help
[Icons: New, Open, Save, Print, Undo, Redo, Copy, Paste, Find]
/*
 * 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__ */

U:--- common.h All L15 (C/l Abbrev)
Wrote /home/itmametskadihrov/common.h
[ ]

U:**- *Messages* Bot L12 (Fundamental)
Wrote /home/itmametskadihrov/common.h
```

В файл server.c добавили цикл while для контроля за временем работы сервера

```
/*
 * 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);
    }
}
```

```
/*Начало отсчета времени*/
clock_t start = time(NULL);

/*Цикл работает, пока с момента начала отсчета времени прошло меньше 30 секунд*/
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);
}
```

В файл client.c добавили цикл, который отвечает за количество сообщений о текущем времени

```
/*
 * 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);
        char* text=ctime(&ttime);
        █
        /* передадим сообщение серверу */
    }
}
```

```
msglen = strlen(text);  
if(write(writefd, text, msglen) != msglen)  
{  
    fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",  
        __FILE__, strerror(errno));  
    exit(-2);  
}  
/* Приостановка работы клиента на 5 секунд */  
sleep(5);  
}  
  
/* закроем доступ к FIFO */  
close(writefd);  
exit(0);  
}
```

Создали Makefile

```
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
```


Скомпилировали программу

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

Проверили работу программы

```
[itmametskadihrov@itmametskadihrov ~]$ ./server
FIFO Server...
Sat Jun 12 18:55:18 2021
Sat Jun 12 18:55:23 2021
Sat Jun 12 18:55:27 2021
Sat Jun 12 18:55:28 2021
Sat Jun 12 18:55:32 2021
Sat Jun 12 18:55:33 2021
Sat Jun 12 18:55:37 2021
Sat Jun 12 18:55:42 2021
[itmametskadihrov@itmametskadihrov ~]$
```

itmametskadihrov@itmam

Файл Правка Вид Поиск Терминал Справка

```
[itmametskadihrov@itmametskadihrov ~]$ ./client
FIFO Client...
[itmametskadihrov@itmametskadihrov ~]$
```

itmametskadihrov@itmam

Файл Правка Вид Поиск Терминал Справка

```
[itmametskadihrov@itmametskadihrov ~]$ ./client
FIFO Client...
[itmametskadihrov@itmametskadihrov ~]$
```

Вывод

Приобрели практические навыки работы с именованными каналами.