

Лабораторная работа

Именованные каналы

ОЗЬЯС Стив Икнэль Дани

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Client1.c	7
4	Client2.c	9
5	Server.c	10
6	Выводы	11
7	Контрольные вопросы	12
	Список литературы	13

Список иллюстраций

2.1	Создание файлов common.h, server.c, client1.c, client2.c	6
3.1	common.h	7
3.2	client1.c	8
4.1	client2.c	9
5.1	server.c	10

Список таблиц

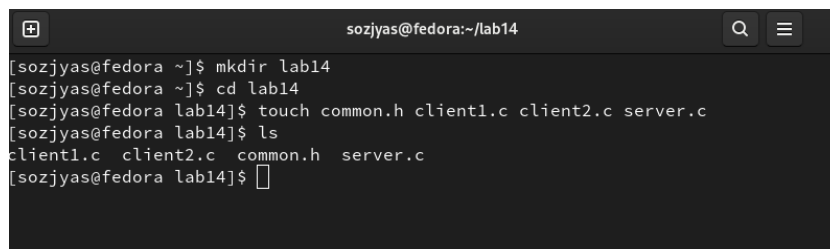
1 Цель работы

Цель данной работы — приобретение практических навыков работы с именovanными каналами.

2 Выполнение лабораторной работы

1. Изучил приведённые в тексте программы server.c и client.c. Взяв данные примеры за образец, напишите аналогичные программы, внося следующие изменения:

- Работает не 1 клиент, а несколько (например, два).(рис. 2.1)

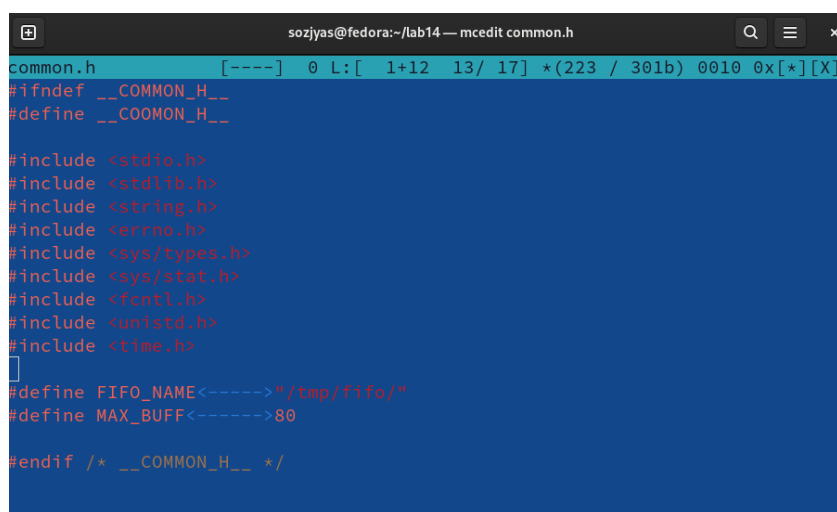


```
sozjyas@fedora:~/lab14
[sozjyas@fedora ~]$ mkdir lab14
[sozjyas@fedora ~]$ cd lab14
[sozjyas@fedora lab14]$ touch common.h client1.c client2.c server.c
[sozjyas@fedora lab14]$ ls
client1.c client2.c common.h server.c
[sozjyas@fedora lab14]$
```

Рис. 2.1: Создание файлов common.h, server.c, client1.c, client2.c

3 Client1.c

- common.h (рис. 3.1)

A screenshot of a terminal window with a dark background. The title bar at the top reads "sozjyas@fedora:~/lab14 — mcedit common.h". The terminal displays the content of a file named "common.h". The code is as follows:

```
common.h [-----] 0 L: [ 1+12 13/ 17] *(223 / 301b) 0010 0x[*][X]
#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__ */
```

Рис. 3.1: common.h

- Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). (рис. 3.2)

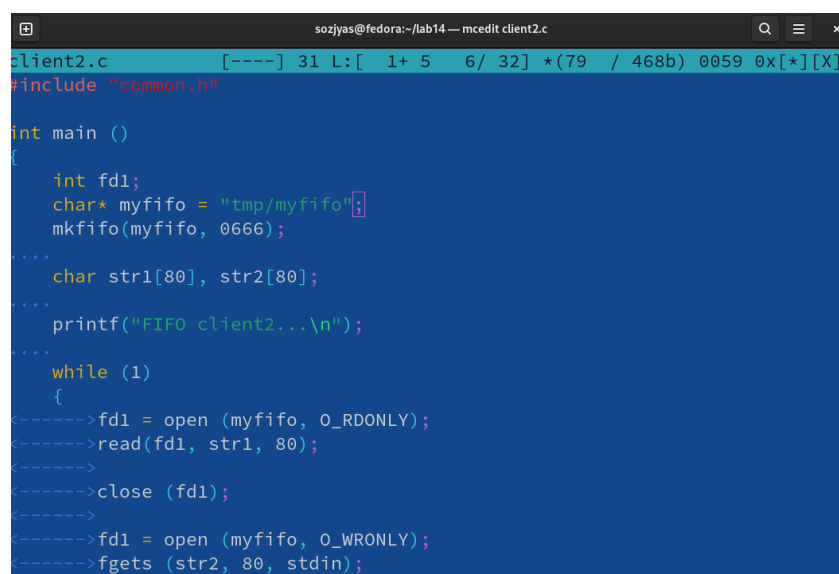
```
sozjyas@fedora:~/lab14 — mcedit client1.c
client1.c [----] 31 L: [ 1+ 5 6/ 33] *(78 / 494b) 0059 0x[*][X]
#include "common.h"

int main ()
{
    int fd;
    char* myfifo = "tmp/myfifo";
    mkfifo(myfifo, 0666);
    ...
    char arr1[80], arr2[80];
    ...
    printf("FIFO client1...\n");
    ...
    while (1)
    {
<----->fd = open(myfifo, O_WRONLY);
<----->fgets(arr2, 80, stdin);
<----->
<----->write(fd, arr2, strlen(arr2) + 1);
<----->
<----->close(fd);
<----->printf("Waiting for User2...\n");
<----->
    }
```

Рис. 3.2: client1.c

4 Client2.c

- Использовал функцию `sleep()` для приостановки работы клиента. (рис. 4.1)



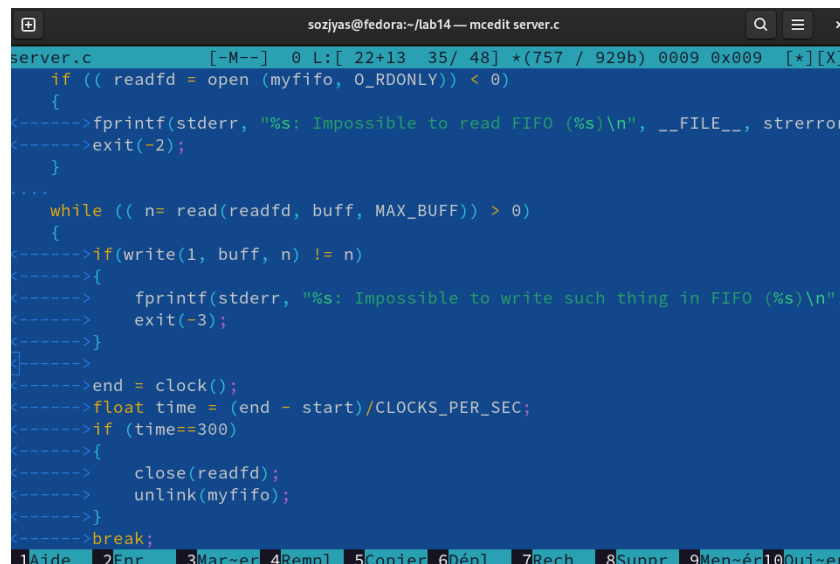
```
client2.c [-----] 31 L: [ 1+ 5 6/ 32] *(79 / 468b) 0059 0x[*][X]
#include "common.h"

int main ()
{
    int fd1;
    char* myfifo = "tmp/myfifo";
    mkfifo(myfifo, 0666);
    ...
    char str1[80], str2[80];
    ...
    printf("FIFO client2...\n");
    ...
    while (1)
    {
<----->fd1 = open (myfifo, O_RDONLY);
<----->read(fd1, str1, 80);
<----->
<----->close (fd1);
<----->
<----->fd1 = open (myfifo, O_WRONLY);
<----->fgets (str2, 80, stdin);
```

Рис. 4.1: client2.c

5 Server.c

- Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Использовал функцию `clock()` для определения времени работы сервера. (рис. 5.1)



```
server.c [-M--] 0 L: [ 22+13 35/ 48] *(757 / 929b) 0009 0x009 [*][X]
if (( readfd = open (myfifo, O_RDONLY)) < 0)
{
<----->fprintf(stderr, "%s: Impossible to read FIFO (%s)\n", __FILE__, strerror
<----->exit(-2);
}
...
while (( n= read(readfd, buff, MAX_BUFF)) > 0)
{
<----->if(write(1, buff, n) != n)
<----->{
<----->    fprintf(stderr, "%s: Impossible to write such thing in FIFO (%s)\n",
<----->    exit(-3);
<----->}
}
<----->end = clock();
<----->float time = (end - start)/CLOCKS_PER_SEC;
<----->if (time==300)
<----->{
<----->    close(readfd);
<----->    unlink(myfifo);
<----->}
<----->break;
```

Рис. 5.1: server.c

6 Выводы

Я приобрел практических навыков работы с именованными каналами.

7 Контрольные вопросы

Список литературы