

Лабораторная работа No 14.

Именованные каналы

Акопян Сатеник

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	10
	Список литературы	11

Список иллюстраций

2.1	рисунок 1	6
2.2	рисунок 2	7
2.3	рисунок 3	8
2.4	рисунок 4	8
2.5	рисунок 5	9

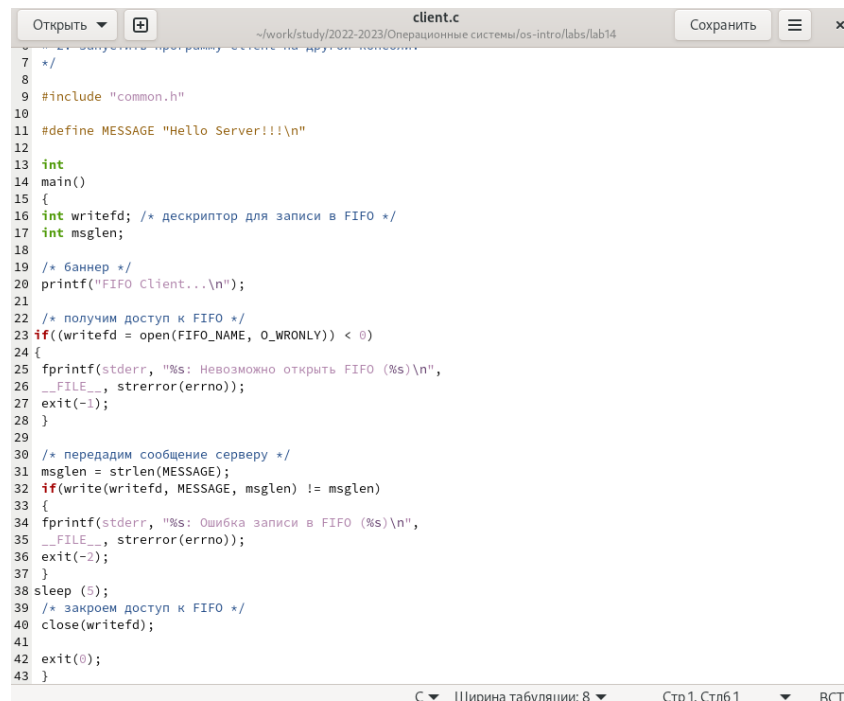
Список таблиц

1 Цель работы

Приобретение практических навыков работы с именованными каналами.

2 Выполнение лабораторной работы

Изучите приведённые в тексте программы server.c и client.c. Взяв данные примеры за образец, напишите аналогичные программы, внося следующие изменения: 1. Работает не 1 клиент, а несколько (например, два).

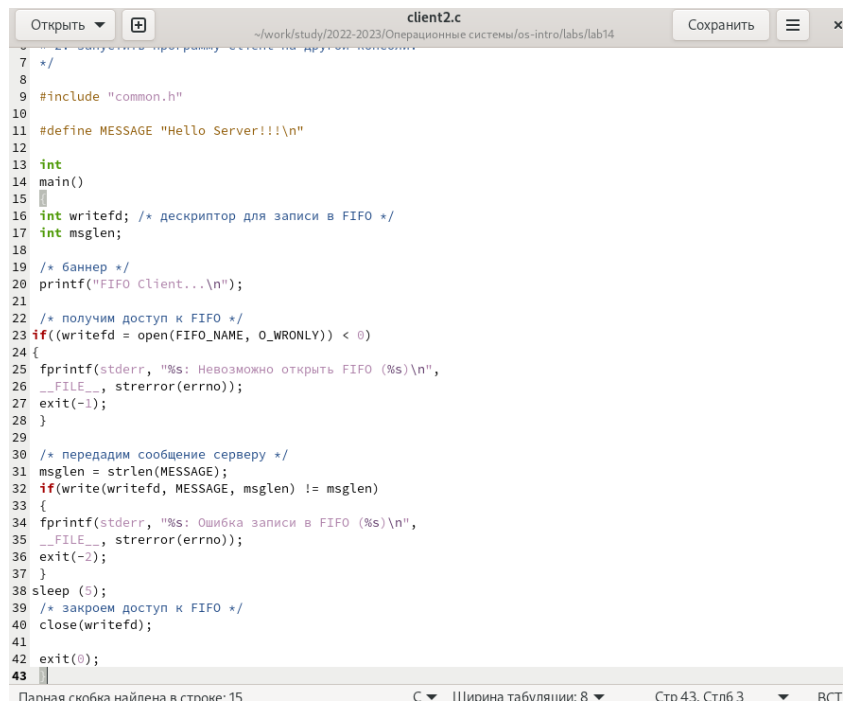


```
client.c
~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14
Сохранить

7  */
8
9  #include "common.h"
10
11 #define MESSAGE "Hello Server!!!\n"
12
13 int
14 main()
15 {
16     int writefd; /* дескриптор для записи в FIFO */
17     int msglen;
18
19     /* баннер */
20     printf("FIFO Client...\n");
21
22     /* получим доступ к FIFO */
23     if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
24     {
25         fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
26             __FILE__, strerror(errno));
27         exit(-1);
28     }
29
30     /* передадим сообщение серверу */
31     msglen = strlen(MESSAGE);
32     if(write(writefd, MESSAGE, msglen) != msglen)
33     {
34         fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
35             __FILE__, strerror(errno));
36         exit(-2);
37     }
38     sleep (5);
39     /* закроем доступ к FIFO */
40     close(writefd);
41
42     exit(0);
43 }
```

Рис. 2.1: рисунок 1

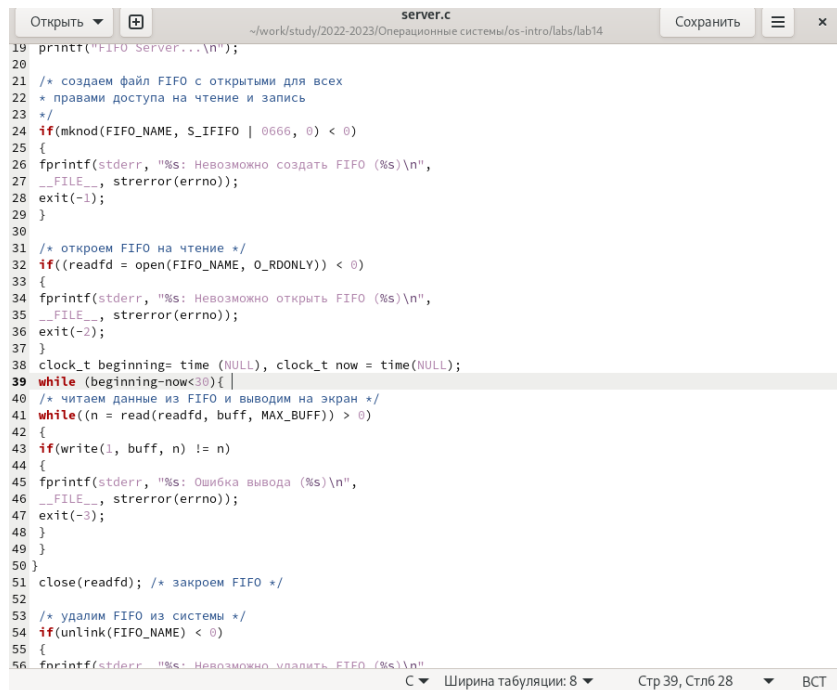
2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Используйте функцию `sleep()` для приостановки работы клиента.



```
7 /*
8
9 #include "common.h"
10
11 #define MESSAGE "Hello Server!!!\n"
12
13 int
14 main()
15 {
16     int writefd; /* дескриптор для записи в FIFO */
17     int msglen;
18
19     /* баннер */
20     printf("FIFO Client...\n");
21
22     /* получим доступ к FIFO */
23     if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
24     {
25         fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
26             __FILE__, strerror(errno));
27         exit(-1);
28     }
29
30     /* передадим сообщение серверу */
31     msglen = strlen(MESSAGE);
32     if(write(writefd, MESSAGE, msglen) != msglen)
33     {
34         fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
35             __FILE__, strerror(errno));
36         exit(-2);
37     }
38     sleep (5);
39     /* закроем доступ к FIFO */
40     close(writefd);
41
42     exit(0);
43 }
```

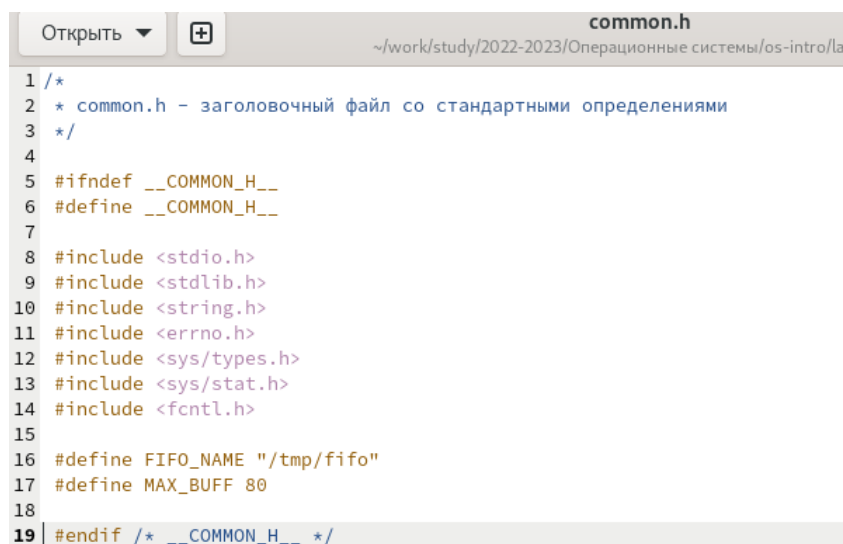
Рис. 2.2: рисунок 2

3. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию `clock()` для определения времени работы сервера. Что будет в случае, если сервер завершит работу, не закрыв канал?



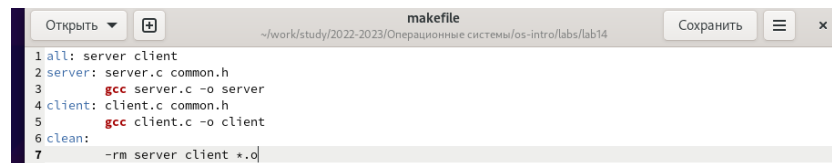
```
19 printf("FIFO Server...\n");
20
21 /* создаем файл FIFO с открытыми для всех
22 * правами доступа на чтение и запись
23 */
24 if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
25 {
26     fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
27         __FILE__, strerror(errno));
28     exit(-1);
29 }
30
31 /* откроем FIFO на чтение */
32 if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
33 {
34     fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
35         __FILE__, strerror(errno));
36     exit(-2);
37 }
38 clock_t beginning= time(NULL), clock_t now = time(NULL);
39 while (beginning-now<30){ |
40 /* читаем данные из FIFO и выводим на экран */
41 while((n = read(readfd, buff, MAX_BUFF)) > 0)
42 {
43     if(write(1, buff, n) != n)
44     {
45         fprintf(stderr, "%s: Ошибка вывода (%s)\n",
46             __FILE__, strerror(errno));
47         exit(-3);
48     }
49 }
50 }
51 close(readfd); /* закроем FIFO */
52
53 /* удалим FIFO из системы */
54 if(unlink(FIFO_NAME) < 0)
55 {
56     fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
57         __FILE__, strerror(errno));
58     exit(-4);
59 }
```

Рис. 2.3: рисунок 3



```
1 /*
2 * common.h - заголовочный файл со стандартными определениями
3 */
4
5 #ifndef __COMMON_H__
6 #define __COMMON_H__
7
8 #include <stdio.h>
9 #include <stdlib.h>
10 #include <string.h>
11 #include <errno.h>
12 #include <sys/types.h>
13 #include <sys/stat.h>
14 #include <fcntl.h>
15
16 #define FIFO_NAME "/tmp/fifo"
17 #define MAX_BUFF 80
18
19 #endif /* __COMMON_H__ */
```

Рис. 2.4: рисунок 4



The image shows a text editor window titled 'makefile' with a path of '~/.work/study/2022-2023/Операционные системы/os-intro/labs/lab14'. The editor contains a Makefile with the following content:

```
1 all: server client
2 server: server.c common.h
3       gcc server.c -o server
4 client: client.c common.h
5       gcc client.c -o client
6 clean:
7       -rm server client *.o
```

Рис. 2.5: рисунок 5

3 Выводы

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

Список литературы