

# Лабораторная работа номер 14

Именованные каналы

Мальков Роман

# Содержание

Цель работы	3
Задание	4
Ход работы	5
Выводы	14

## Цель работы

Приобретение практических навыков работы с именованными каналами.

## Задание

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

# Ход работы

Код и скрины приведены ниже ( Скриншоты 1 - 4 ):

## 1. 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
```

## 2. common.h

```
/*
```

```
* common.h - заголовочный файл со стандартными определениями
```

```
*/
```

```
#ifndef __COMMON_H__
```

```
#define __COMMON_H__
```

```
#include <time.h>
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
```

```
#define FIFO_NAME "/tmp/fifo"
#define MAX_BUFF 80
```

```
#endif /* __COMMON_H__ */
```

### 3. client.c

```
/*
 * 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;
```

```

int i;
long int ttime;

for(i = 0; i < 5; i++)
{
sleep(5);
ttime = time(NULL);

/* баннер */
printf("FIFO Client...\n");

/* 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);
}

/* 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);
}

/* закроем доступ к FIFO */
close(writefd);
}

```

```
exit(0);
```

```
}
```

#### 4. server.c

```
#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 cl = clock();
while (cl/CLOCKS_PER_SEC < 30)
{
    cl = clock();
    /* читаем данные из FIFO и выводим на экран */

    while((n = read(readfd, buff, MAX_BUFF)) > 0)
    {
        printf("sec = %d\n", cl/CLOCKS_PER_SEC);
        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);
```

```
}
```

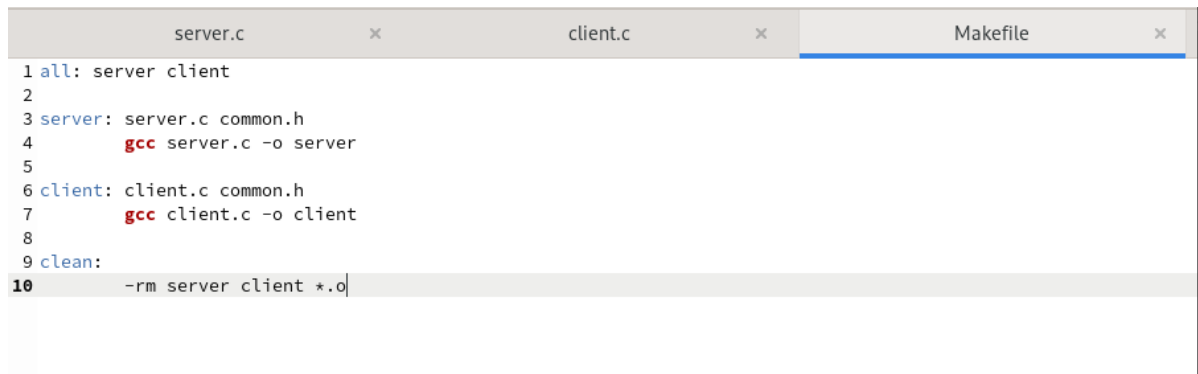


Рис. 1: Screenshot\_1

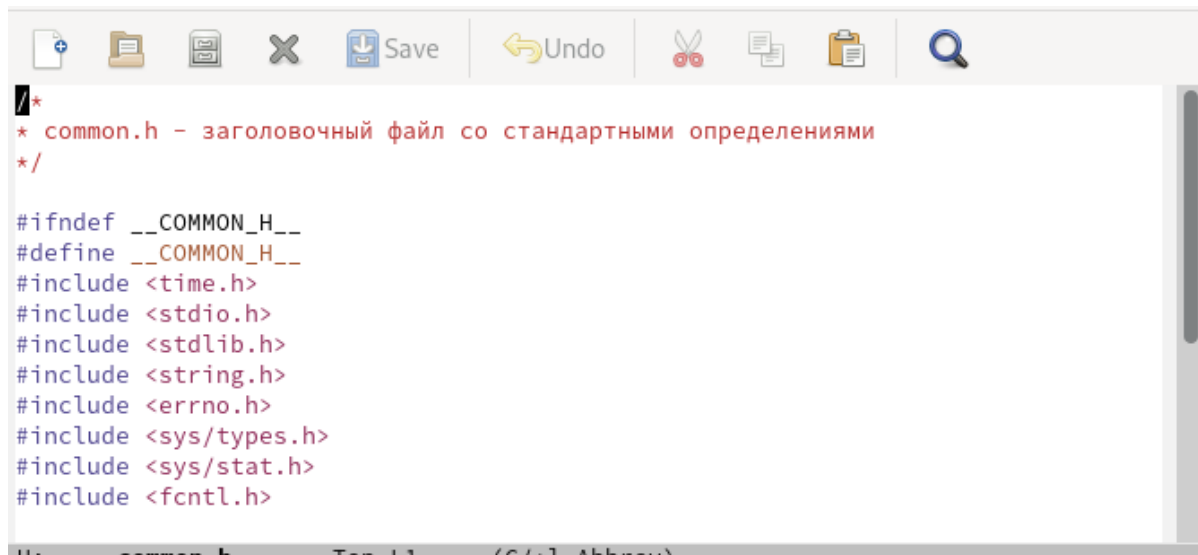


Рис. 2: Screenshot\_2

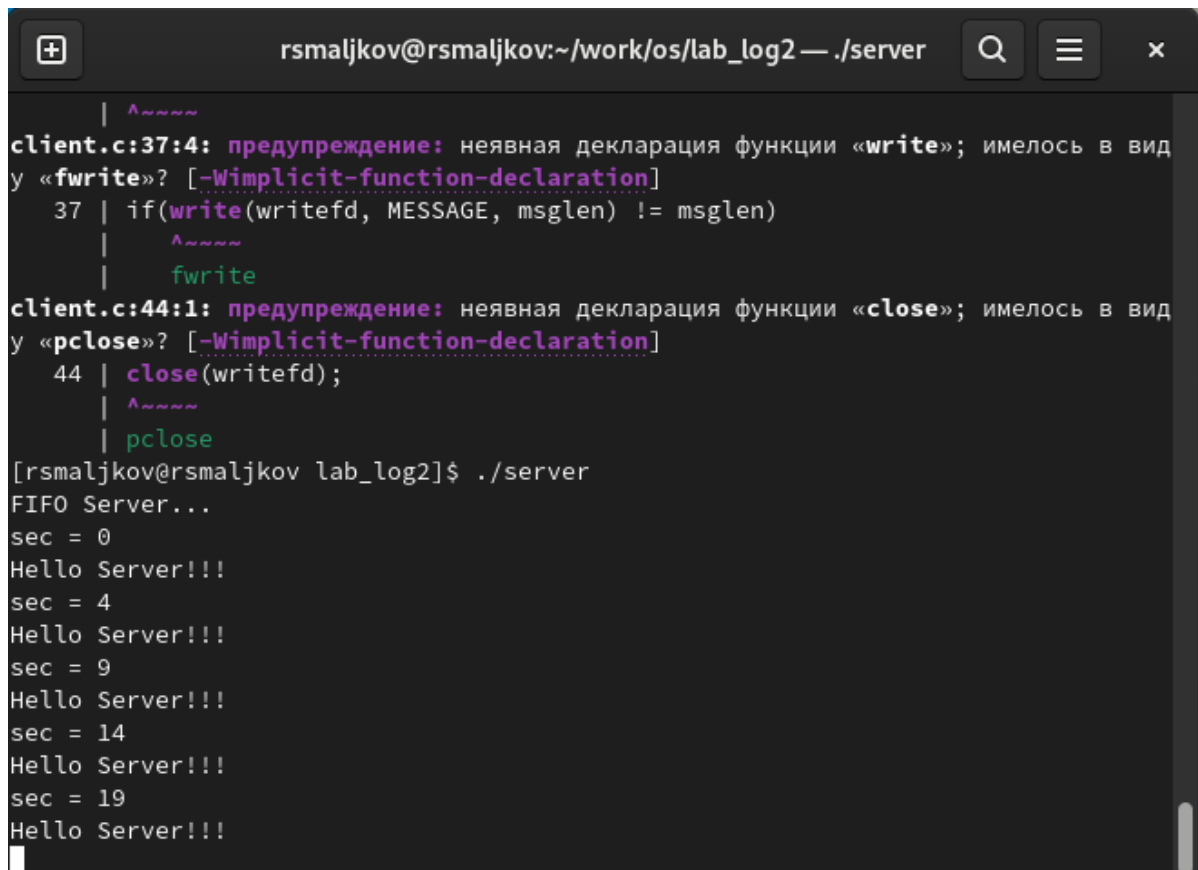
```
server.c  x  client.c  x  Makefile  x
1 /*
2 * client.c - реализация клиента
3 *
4 * чтобы запустить пример, необходимо:
5 * 1. запустить программу server на одной консоли;
6 * 2. запустить программу client на другой консоли.
7 */
8
9 #include "common.h"
10
11 #define MESSAGE "Hello Server!!!\n"
12
13 int main()
14 {
15     int writefd; /* дескриптор для записи в FIFO */
16     int msglen;
17     int i;
18     long int ttime;
19
20     for(i = 0; i < 5; i++)
21     {
22         sleep(5);
```

Рис. 3: Screenshot\_3

```
1
2 #include "common.h"
3
4 int
5 main()
6 {
7     int readfd; /* дескриптор для чтения из FIFO */
8     int n;
9     char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */
10
11     /* баннер */
12     printf("FIFO Server...\n");
13
14     /* создаем файл FIFO с открытыми для всех
15      * правами доступа на чтение и запись
16      */
17     if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
18     {
19         fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
20             __FILE__, strerror(errno));
21         exit(-1);
22     }
23
24     /* откроем FIFO на чтение */
25     if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
26     {
```

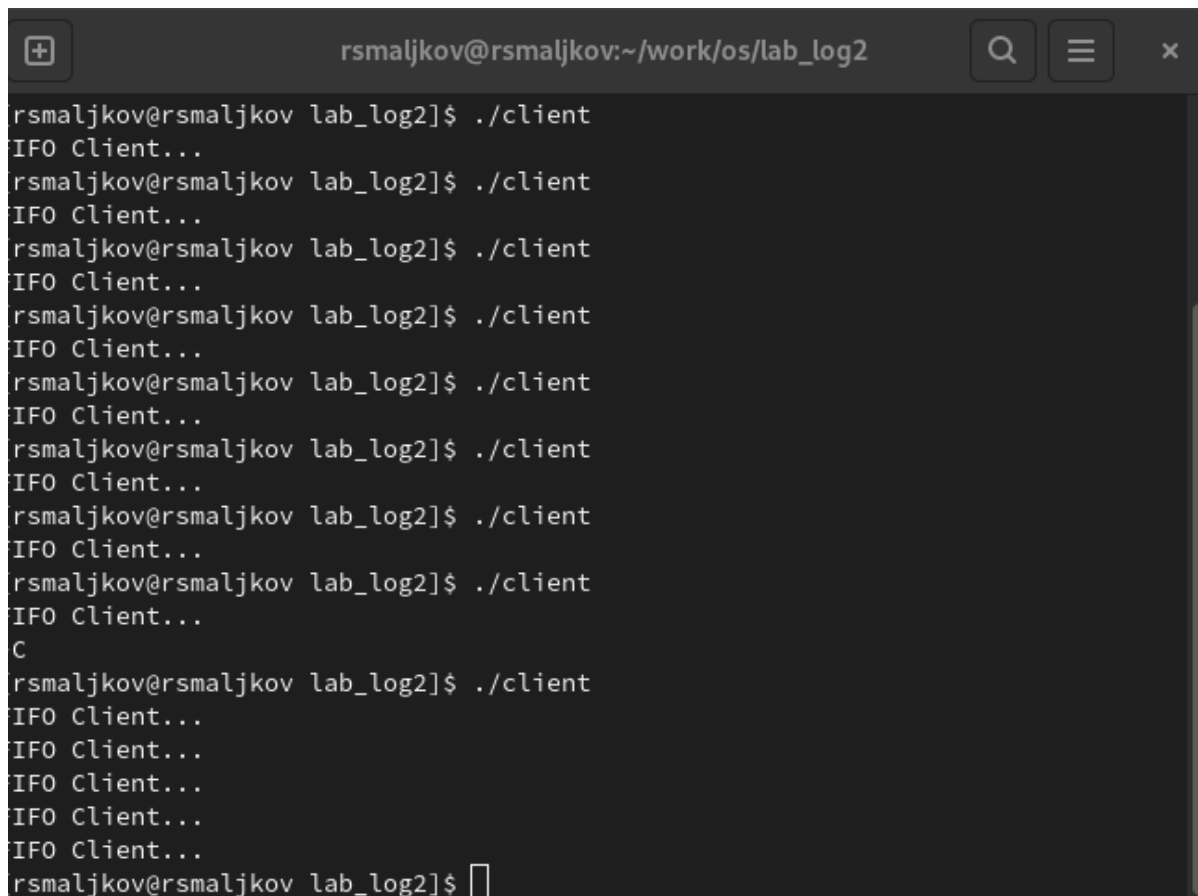
Рис. 4: Screenshot\_4

Пример выполнения программы ( Скриншоты 5 - 6)



```
client.c:37:4: предупреждение: неявная декларация функции «write»; имелось в вид
y «fwrite»? [-Wimplicit-function-declaration]
  37 | if(write(writefd, MESSAGE, msglen) != msglen)
      |      ^~~~~
      |      fwrite
client.c:44:1: предупреждение: неявная декларация функции «close»; имелось в вид
y «pclose»? [-Wimplicit-function-declaration]
  44 | close(writefd);
      |      ^~~~~
      |      pclose
[rsmaljkov@rsmaljkov lab_log2]$ ./server
FIFO Server...
sec = 0
Hello Server!!!
sec = 4
Hello Server!!!
sec = 9
Hello Server!!!
sec = 14
Hello Server!!!
sec = 19
Hello Server!!!
```

Рис. 5: Screenshot\_5



```
rsmaljkov@rsmaljkov:~/work/os/lab_log2
rsmaljkov@rsmaljkov lab_log2]$ ./client
IFO Client...
rsmaljkov@rsmaljkov lab_log2]$ ./client
IFO Client...
rsmaljkov@rsmaljkov lab_log2]$ ./client
IFO Client...
rsmaljkov@rsmaljkov lab_log2]$ ./client
IFO Client...
rsmaljkov@rsmaljkov lab_log2]$ ./client
IFO Client...
rsmaljkov@rsmaljkov lab_log2]$ ./client
IFO Client...
rsmaljkov@rsmaljkov lab_log2]$ ./client
IFO Client...
rsmaljkov@rsmaljkov lab_log2]$ ./client
IFO Client...
rsmaljkov@rsmaljkov lab_log2]$ ./client
IFO Client...
rsmaljkov@rsmaljkov lab_log2]$ C
rsmaljkov@rsmaljkov lab_log2]$ ./client
IFO Client...
IFO Client...
IFO Client...
IFO Client...
IFO Client...
rsmaljkov@rsmaljkov lab_log2]$
```

Рис. 6: Screenshot\_6

## Выводы

Мы приобрели практические навыки работы с именованными каналами.