

# **Лабораторная работа №14**

**Именованные каналы**

Заболотная Кристина Александровна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>13</b>
	<b>Список литературы</b>	<b>14</b>

## Список иллюстраций

3.1	создаем файлы . . . . .	7
3.2	изменим common.h . . . . .	8
3.3	server.c . . . . .	9
3.4	client.c . . . . .	10
3.5	less . . . . .	11
3.6	файлы . . . . .	11
3.7	make all . . . . .	12
3.8	./server . . . . .	12

## Список таблиц

# **1 Цель работы**

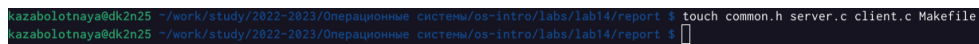
Приобретение практических навыков работы с именованными каналами.

## 2 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?

### 3 Выполнение лабораторной работы

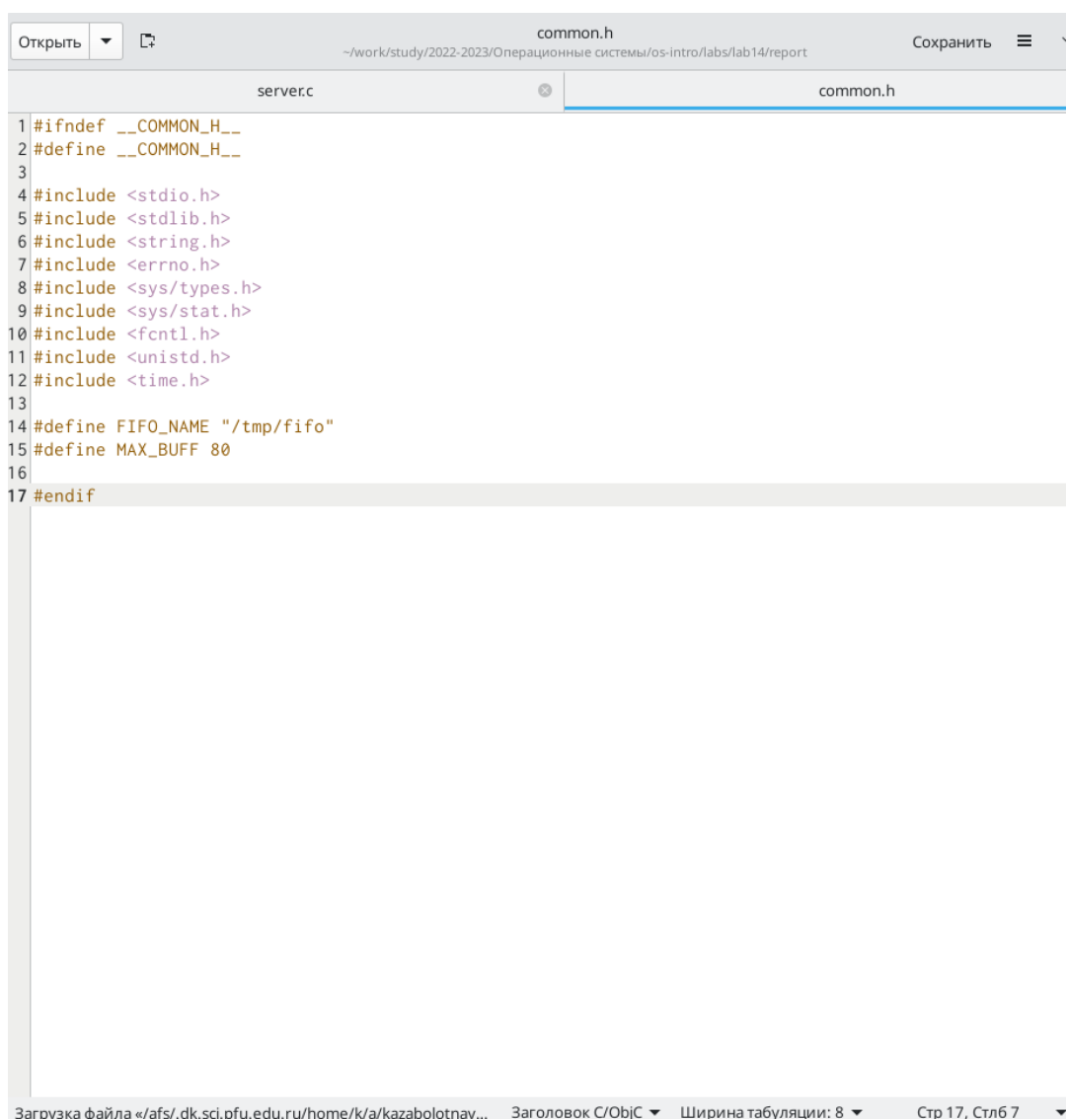
1. Создадим необходимые файлы с помощью команды touch. Откроем редактор emacs для их редактирования.



```
kazaboltnaya@dk2n25 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14/report $ touch common.h server.c client.c Makefile
kazaboltnaya@dk2n25 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14/report $
```

Рис. 3.1: создаем файлы

2. Изменим коды программ, представленных в задании лабораторной работы. В файле common.h добавим основные заголовочные файлы.



```
1 #ifndef __COMMON_H__
2 #define __COMMON_H__
3
4 #include <stdio.h>
5 #include <stdlib.h>
6 #include <string.h>
7 #include <errno.h>
8 #include <sys/types.h>
9 #include <sys/stat.h>
10 #include <fcntl.h>
11 #include <unistd.h>
12 #include <time.h>
13
14 #define FIFO_NAME "/tmp/fifo"
15 #define MAX_BUFF 80
16
17 #endif
```

Рис. 3.2: изменим common.h

3. В файле server.c добавим цикл while, контролирующий время работы сервера.

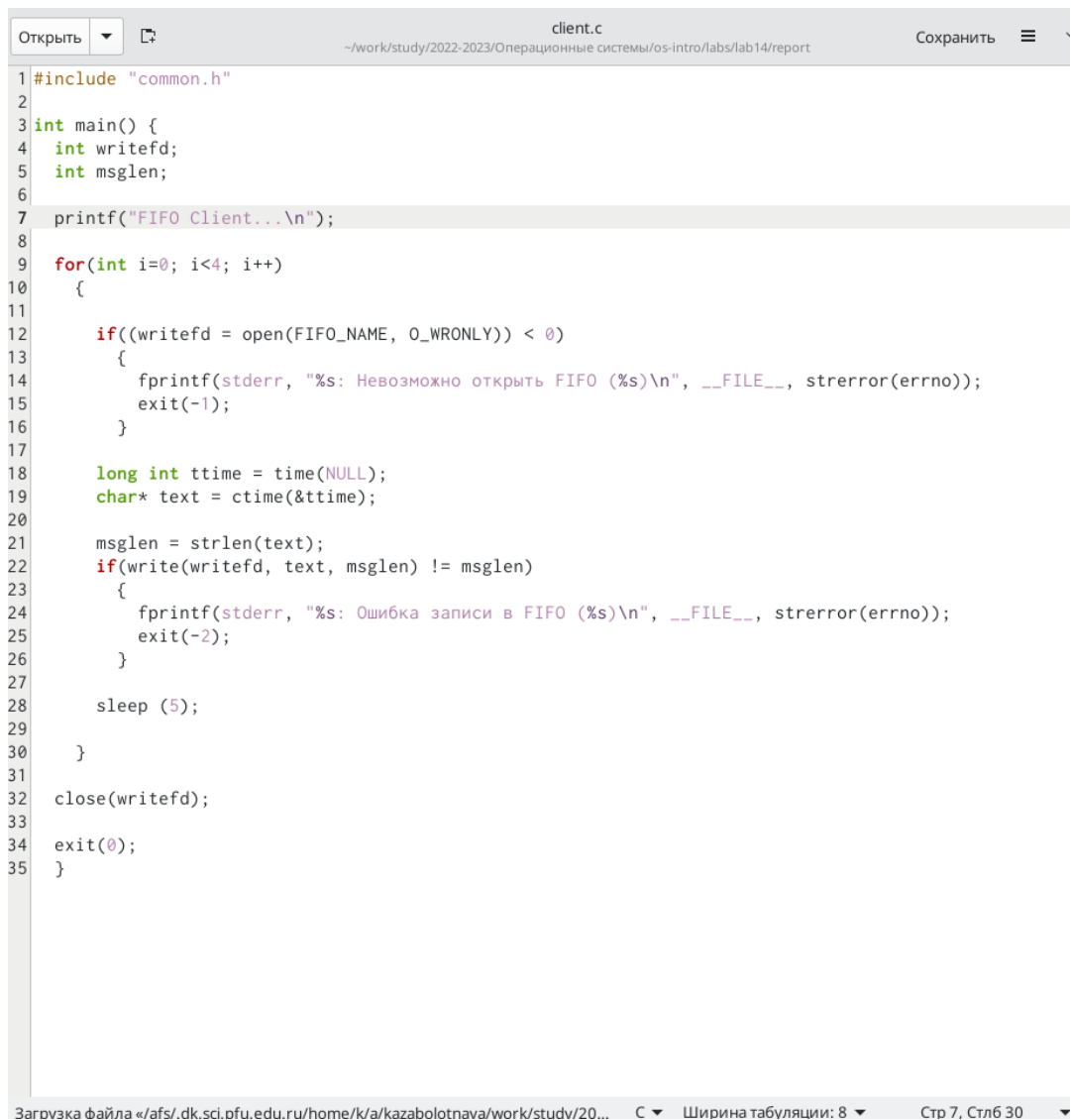


```
1 #include "common.h"
2
3 int main() {
4     int readfd;
5     int n;
6     char buff[MAX_BUFF];
7
8     printf("FIFO Server...\n");
9
10    if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
11    {
12        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n", __FILE__, strerror(errno));
13        exit(-1);
14    }
15
16    if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
17    {
18        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n", __FILE__, strerror(errno));
19        exit (-2);
20    }
21
22    clock_t start = time(NULL);
23
24    while(time(NULL)-start < 30)
25    {
26        while((n = read(readfd, buff, MAX_BUFF)) > 0)
27        {
28            if(write(1, buff, n) != n)
29            {
30                fprintf(stderr, "%s: Ошибка вывода (%s)\n", __FILE__, strerror(errno));
31                exit(-3);
32            }
33        }
34    }
35
36    close(readfd);
37
38    if(unlink(FIFO_NAME) < 0)
39    {
40        fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n", __FILE__, strerror(errno));
41        exit(-4);
42    }
43
44    exit(0);
45 }
```

Парная скобка найдена в строке: 20 C Ширина табуляции: 8 Стр 17, Стлб 6

Рис. 3.3: server.c

4. В файле client.c добавим цикл while, отвечающий за количество сообщений о текущем времени, которое получается в результате выполнения команд.



```
1 #include "common.h"
2
3 int main() {
4     int writefd;
5     int msglen;
6
7     printf("FIFO Client...\n");
8
9     for(int i=0; i<4; i++)
10     {
11         if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
12         {
13             fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n", __FILE__, strerror(errno));
14             exit(-1);
15         }
16
17         long int ttime = time(NULL);
18         char* text = ctime(&ttime);
19
20         msglen = strlen(text);
21         if(write(writefd, text, msglen) != msglen)
22         {
23             fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n", __FILE__, strerror(errno));
24             exit(-2);
25         }
26
27         sleep (5);
28     }
29
30     close(writefd);
31
32     exit(0);
33 }
34
35
```

Загрузка файла «/afs/.dk.sci.sfu.edu.ru/home/k/a/kazaboltnava/work/studv/20... C ▾ Ширина табуляции: 8 ▾ Стр 7, Стлб 30 ▾

Рис. 3.4: client.c

## 5. Makefile - файл для сборки.

```
Открыть ▼  Makefile
~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14

1 all: server client
2
3 server: server.c common.h
4     gcc server.c -o server
5
6 client: client.c common.h
7     gcc client.c -o client
8
9 clean:
10     -rm server client *.o
```

Рис. 3.5: less

6. Скомпилируем необходимые файлы с помощью команды `make all`.

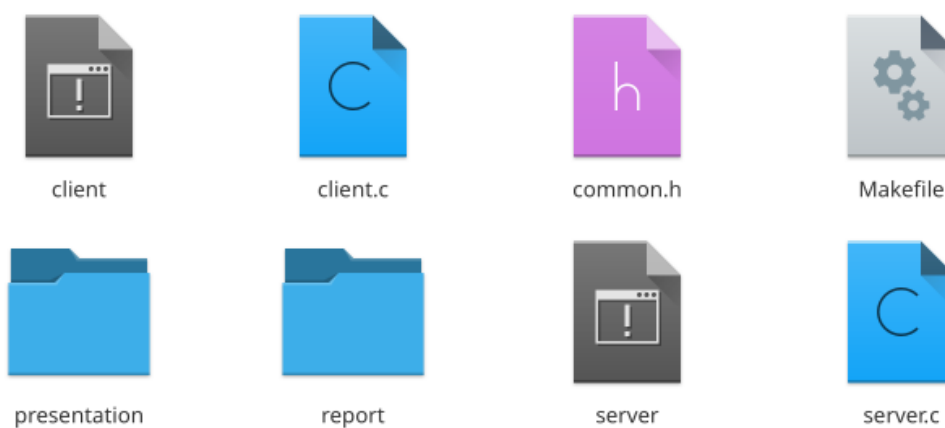
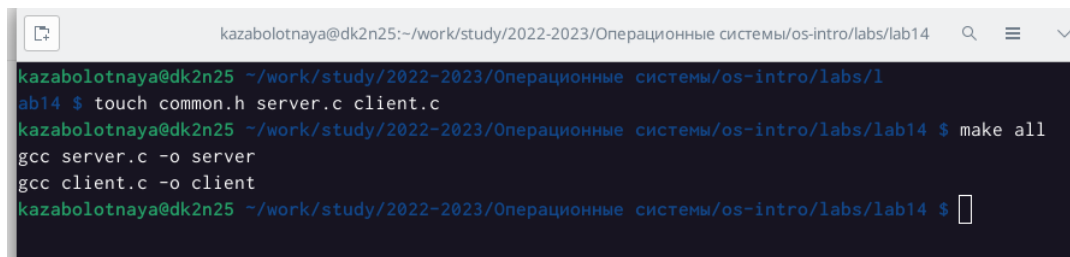


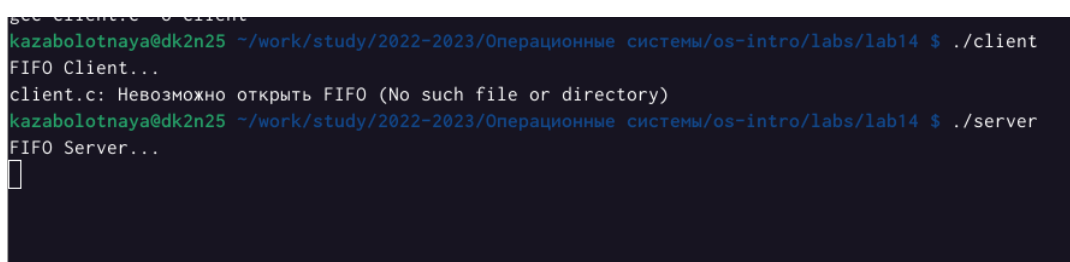
Рис. 3.6: файлы



```
kazaboltnaya@dk2n25:~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14
kazaboltnaya@dk2n25 ~/work/study/2022-2023/Операционные системы/os-intro/labs/1
ab14 $ touch common.h server.c client.c
kazaboltnaya@dk2n25 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14 $ make all
gcc server.c -o server
gcc client.c -o client
kazaboltnaya@dk2n25 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14 $
```

Рис. 3.7: make all

7. Проверяем работу написанного кода.



```
kazaboltnaya@dk2n25 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14 $ ./client
FIFO Client...
client.c: Невозможно открыть FIFO (No such file or directory)
kazaboltnaya@dk2n25 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14 $ ./server
FIFO Server...

```

Рис. 3.8: ./server

## **4 Выводы**

В ходе выполнения данной лабораторной работы мы приобрели практические навыки работы с именованными каналами.

## **Список литературы**