

## Лабораторная работа №14. Именованные каналы.

---

Беличева Д.М.; НКНбд-01-21

<sup>1</sup>RUDN University, Moscow, Russian Federation

Приобретение практических навыков работы с именованными каналами.

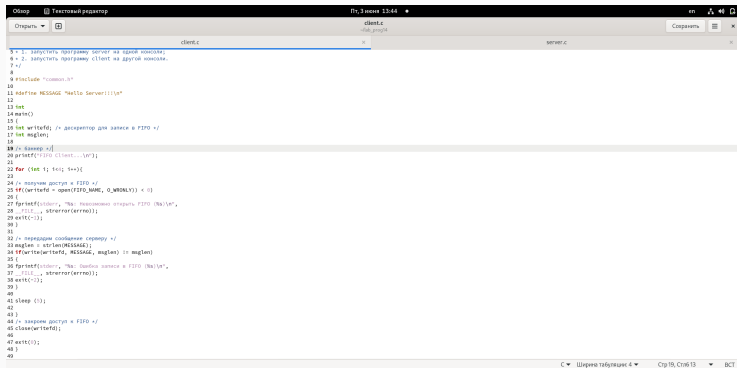
Изучите приведённые в тексте программы `server.c` и `client.c`. Взяв данные примеры за образец, напишите аналогичные программы, внося следующие изменения: 1. Работает не 1 клиент, а несколько (например, два). 2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). 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?

Одним из видов взаимодействия между процессами в операционных системах является обмен сообщениями. Под сообщением понимается последовательность байтов, передаваемая от одного процесса другому. В операционных системах типа UNIX есть 3 вида межпроцессорных взаимодействий: общенуксные (именованные каналы, сигналы), System V Interface Definition (SVID — разделяемая память, очередь сообщений, семафоры) и BSD (сокеты).

Для передачи данных между неродственными процессами можно использовать механизм именованных каналов (named pipes). Данные передаются по принципу FIFO (First In First Out) (первым записан — первым прочитан), поэтому они называются также FIFO pipes или просто FIFO. Именованные каналы отличаются от неименованных наличием идентификатора канала, который представлен как специальный файл (соответственно имя именованного канала — это имя файла). Поскольку файл находится на локальной файловой системе, данное IPC используется внутри одной системы. Файлы именованных каналов создаются функцией `mkfifo(3)`.

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. (рис. 1, 2)

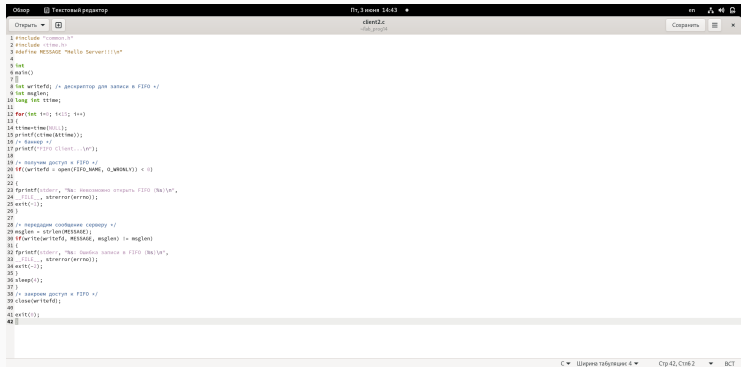
# Выполнение лабораторной работы



```
1 // Запустить программу server из этого кода;
2 // Запустить программу client из этого кода.
3 //
4
5 #include "common.h"
6
7 #define MESSAGE "Hello Server!!!!"
8
9 int
10 main()
11 {
12     int writefd; /* дескриптор для записи в FIFO */
13     int readfd;
14
15     /* Закрыть */
16     printf("FIFO client...\n");
17
18     /* Открыть доступ к FIFO */
19     if (writefd = open(FIFO_NAME, O_WRONLY) < 0)
20     {
21         fprintf(stderr, "Невозможно открыть FIFO (%s)\n",
22             FILE_, strerror(errno));
23         exit(-1);
24     }
25
26     /* Передача сообщения серверу */
27     msglen = strlen(MESSAGE);
28     if (fwrite(MESSAGE, sizeof(char), msglen, writefd) != msglen)
29     {
30         fprintf(stderr, "Не удалось записать в FIFO (%s)\n",
31             FILE_, strerror(errno));
32         exit(-1);
33     }
34
35     /* Закрыть доступ к FIFO */
36     close(writefd);
37
38     return 0;
39 }
```

Figure 1: Текст программы

# Выполнение лабораторной работы



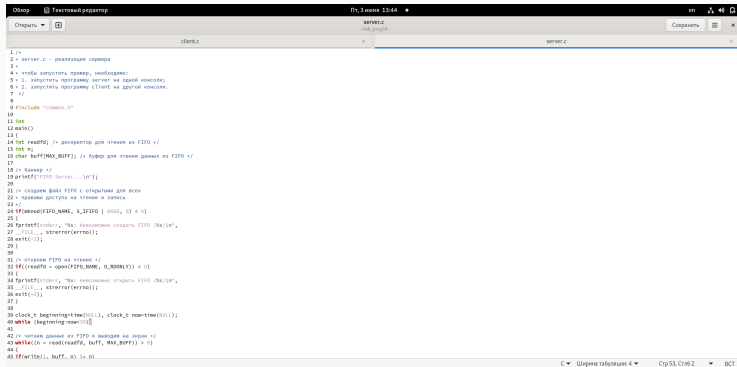
```
1 #include "common.h"
2 #include <time.h>
3 #define MESSAGE "Hello Server!!!!"
4
5 int
6 main()
7 {
8     int writefd; /* дескриптор для записи в FIFO */
9     int mglen;
10     long sec_ttime;
11
12     for (sec_ttime = 0; sec_ttime < 10; sec_ttime++)
13     {
14         struct timeval tv;
15         printf("ctime(%d)\n", tv.tv_sec);
16         /* sleep */
17         printf("ctime(%d)\n", tv.tv_sec);
18
19         /* получаем дескриптор к FIFO */
20         if ((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
21         {
22             fprintf(stderr, "на: невозможно открыть FIFO (%d)\n",
23                     _FILE_, strerror(errno));
24             exit(-1);
25         }
26
27         /* передаем сообщение серверу */
28         mglen = strlen(MESSAGE);
29         if (write(writefd, MESSAGE, mglen) != mglen)
30         {
31             fprintf(stderr, "на: ошибка записи в FIFO (%d)\n",
32                     _FILE_, strerror(errno));
33             exit(-1);
34         }
35         sleep(1);
36
37         /* закрываем дескриптор к FIFO */
38         close(writefd);
39         exit(0);
40     }
41 }
```

Figure 2: Текст программы



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? (рис. 3, 4, 5, 6)

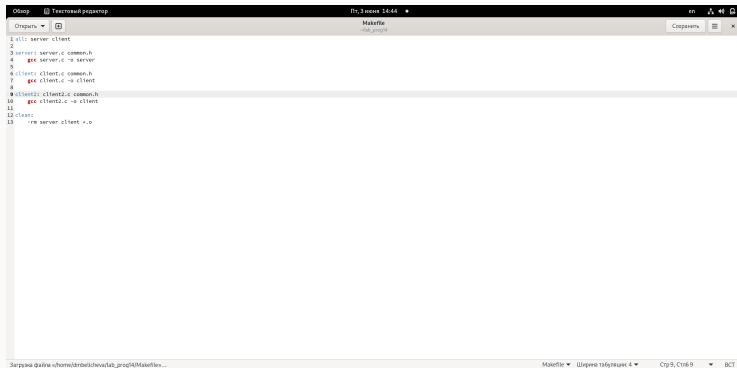
# Выполнение лабораторной работы



```
1 /*
2  * server.c - реализация сервера
3  */
4  * чтобы запустить пример, необходимо:
5  * 1. запустить программу сервер на одной консоли;
6  * 2. запустить программу client на другой консоли.
7  */
8
9 #include "common.h"
10
11 int
12 main()
13 {
14     int readfd; /* дескриптор для чтения из FIFO */
15     int rc;
16     char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */
17
18     /* Backup */
19     printf("FIFO Server...\n");
20
21     /* создаем файл FIFO с открытием для всех
22      * правами доступа на чтение и запись
23      */
24     if(mkfifo(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
25     {
26         fprintf(stderr, "на: невозможно создать FIFO (%d)\n",
27                 errno);
28         exit(-1);
29     }
30
31     /* открываем FIFO на чтение */
32     if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
33     {
34         fprintf(stderr, "на: невозможно открыть FIFO (%d)\n",
35                 errno);
36         exit(-1);
37     }
38
39     clock_t beginning_time[100], clock_t now_time[100];
40     while (beginning_now[0])
41     {
42         /* чтение данных из FIFO и выход на экран */
43         while((n = read(readfd, buff, MAX_BUFF)) > 0)
44         {
45             if(write(1, buff, n) != n)
```

Figure 3: Текст программы

# Выполнение лабораторной работы



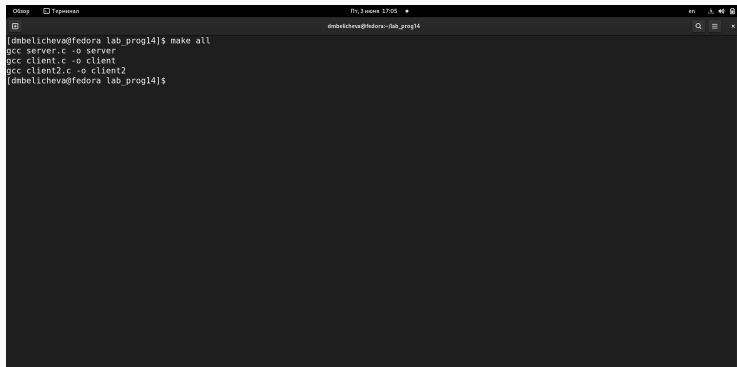
The screenshot shows a code editor window titled "Текстовый редактор" (Text Editor) with a dark theme. The editor displays a Makefile with the following content:

```
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 client2: client2.c common.h
10    gcc client2.c -o client2
11
12 clean:
13     rm server client *.o
```

The editor interface includes a top bar with "Фир, 3 июня 14:44" and "en". A right sidebar shows "Makefile" and "lab\_prog4". The bottom status bar indicates the file path "Зарядка g:\file\chrome\download\lab\_prog4\Makefile...", the editor name "Makefile", the tab count "Ширине таблички 4", the page number "Стр 9, Стр 9", and the encoding "ВСТ".

Figure 4: Текст программы

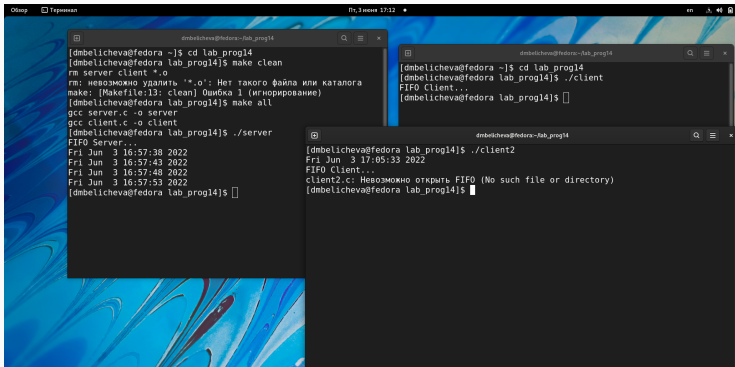
# Выполнение лабораторной работы

A terminal window titled "Терминал" (Terminal) showing a series of compilation commands. The user is in a directory named "lab\_prog14" on a Fedora system. The commands executed are: "make all", "gcc server.c -o server", "gcc client.c -o client", and "gcc client2.c -o client2". The prompt returns to the user after each command.

```
dnbelicheva@fedora lab_prog14$ make all
gcc server.c -o server
gcc client.c -o client
gcc client2.c -o client2
dnbelicheva@fedora lab_prog14$
```

Figure 5: Компиляция

# Выполнение лабораторной работы



```
[dmbelicheva@fedora ~]$ cd lab_prog14
[dmbelicheva@fedora lab_prog14]$ make clean
rm server client *.o
rm: невозможно удалить '*.o': Нет такого файла или каталога
make: [Makefile:13: clean] Ошибка 1 (игнорирование)
[dmbelicheva@fedora lab_prog14]$ make all
gcc server.c -o server
gcc client.c -o client
[dmbelicheva@fedora lab_prog14]$ ./server
FIFO Server...
Fri Jun 3 16:57:38 2022
Fri Jun 3 16:57:43 2022
Fri Jun 3 16:57:48 2022
Fri Jun 3 16:57:53 2022
[dmbelicheva@fedora lab_prog14]$

[dmbelicheva@fedora ~]$ cd lab_prog14
[dmbelicheva@fedora lab_prog14]$ ./client
FIFO Client...
[dmbelicheva@fedora lab_prog14]$

[dmbelicheva@fedora lab_prog14]$ ./client2
Fri Jun 3 17:05:33 2022
FIFO Client...
client2.c: Невозможно открыть FIFO (No such file or directory)
[dmbelicheva@fedora lab_prog14]$
```

Figure 6: Результат

В процессе выполнения лабораторной работы я приобрела практические навыки работы с именованными каналами.

1. Лабораторная работа № 14. Именованные каналы [Электронный ресурс]. URL: <https://esystem.rudn.ru/>.

Спасибо за внимание!