

# Краткий отчёт по лабораторной работе №14

Samsonova Maria, Student of RUDN University, Moscow, Russian Federation

## Цель выполнения лабораторной работы №14

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?4. 4.

## Ход выполнения лабораторной работы №14

1. Для начала мы создали необходимые файлы с помощью команды «`touch common.h server.c client.c Makefile`» и открыли редактор `emacs` для их редактирования (рис. -@fig:001 ).



```
misamsonova@fedora:~  
[misamsonova@fedora ~]$ touch common.h server.c client.c Makefile  
[misamsonova@fedora ~]$ emacs &
```

### Создание файлов

2. Далее изменили коды программ, представленных в тексте лабораторной работы. В файл `common.h` добавила стандартные заголовочные файлы `unistd.h` и `time.h`, необходимые для работы кодов других файлов. `Common.h` предназначен для заголовочных файлов, чтобы в остальных программах их не прописывать каждый раз (рис. -@fig:002 ).

*Программа в файле common.h*

- В файл server.c добавили цикл while для контроля за временем работы сервера. Разница между текущим временем time(NULL) и временем начала работы clock\_t start=time(NULL) (инициализация до цикла) не должна превышать 30 секунд (рис. -@fig:003 , -@fig:004 ).

```
emacs@fedora
File Edit Options Buffers Tools C Help
[Icons: New, Open, Save, Undo, Cut, Copy, Paste, Find]

/*
 * server.c - реализация сервера
 *
 * чтобы запустить пример, необходимо:
 * 1. запустить программу server на одной консоли;
 * 2. запустить программу client на другой консоли.
 */

#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);
    }
}
```

*Программа в файле server.c*

```
emacs@fedora
File Edit Options Buffers Tools C Help
[Icons: New, Open, Save, Undo, Redo, Find, etc.]

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);
}

/* начало отсчёта времени*/
clock_t start = time(NULL);

/*цикл работает, пока с момента начала отсчёта времени прошло меньше 30 секунд*/
while(time(NULL)-start<30)
{
    /* читаем данные из FIFO и выводим на экран */
    while((n = read(readfd, buff, MAX_BUFF)) > 0)
    {
        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);
}
```

### Программа в файле server.c

- В файл client.c добавили цикл, который отвечает за количество сообщений о текущем времени (4 сообщения), которое получается в результате выполнения команд на рис. 7 (/текущее время/) и команду sleep(5) для приостановки работы клиента на 5 секунд (рис. -@fig:005 , -@fig:006 ).

```
emacs@fedora
File Edit Options Buffers Tools C Help
[Icons: New, Open, Save, Undo, Redo, Find, etc.]

/*
 * 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;

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

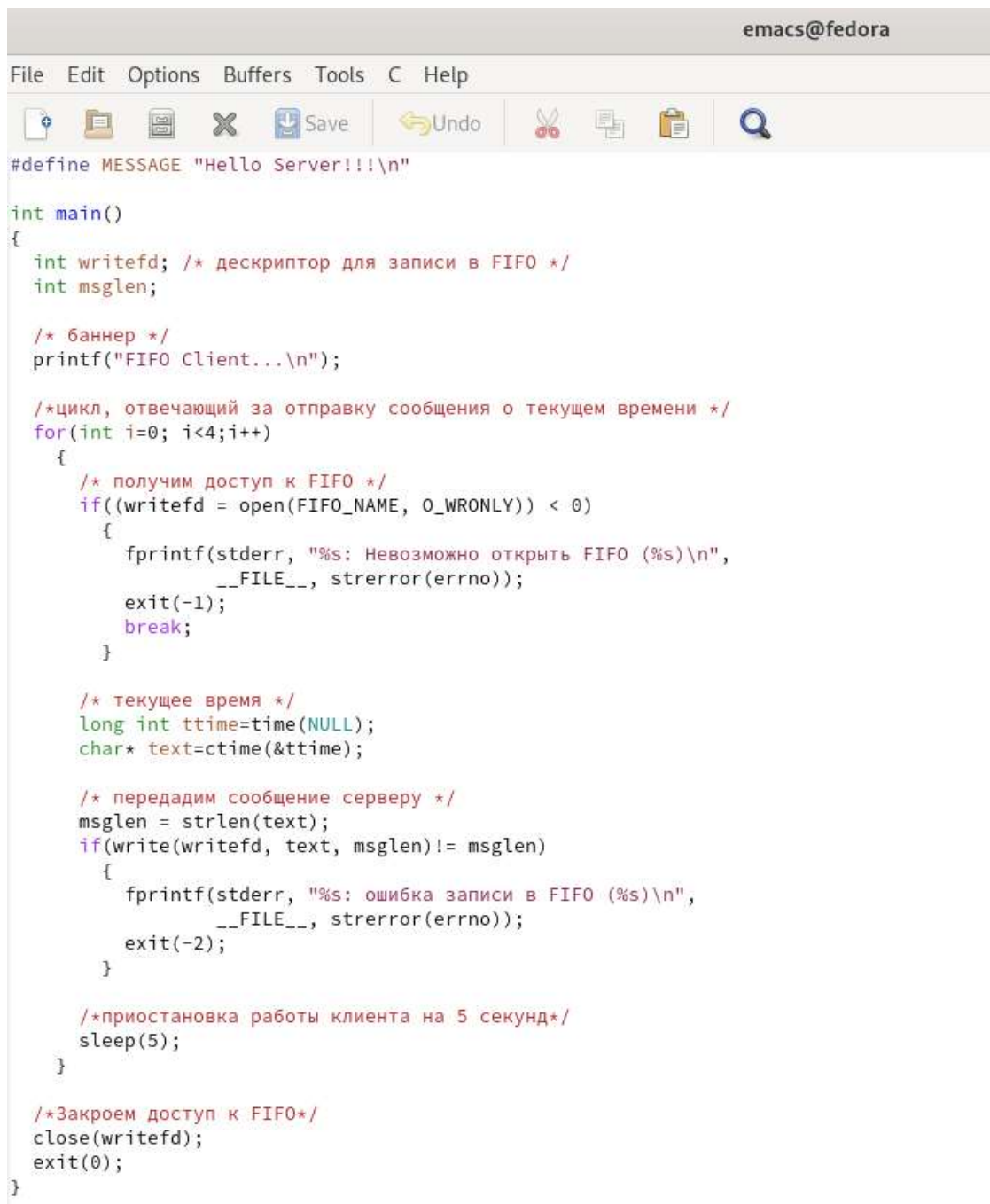
    /*цикл, отвечающий за отправку сообщения о текущем времени */
    for(int i=0; i<4;i++)
    {
        /* получим доступ к FIFO */
        if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
                    __FILE__, strerror(errno));
            exit(-1);
            break;
        }

        /* текущее время */
        long int ttime=time(NULL);
        char* text=ctime(&ttime);

        /* передадим сообщение серверу */
        msglen = strlen(text);
        if(write(writefd, text, msglen)!= msglen)
        {
            fprintf(stderr, "%s: ошибка записи в FIFO (%s)\n",
                    __FILE__, strerror(errno));
            exit(-2);
        }

        /*приостановка работы клиента на 5 секунд*/
        sleep(5);
    }
}
```

*Прграмма в файле client.c*



```
#define MESSAGE "Hello Server!!!\n"

int main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;

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

    /*цикл, отвечающий за отправку сообщения о текущем времени */
    for(int i=0; i<4;i++)
    {
        /* 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);
            break;
        }

        /* текущее время */
        long int ttime=time(NULL);
        char* text=ctime(&ttime);

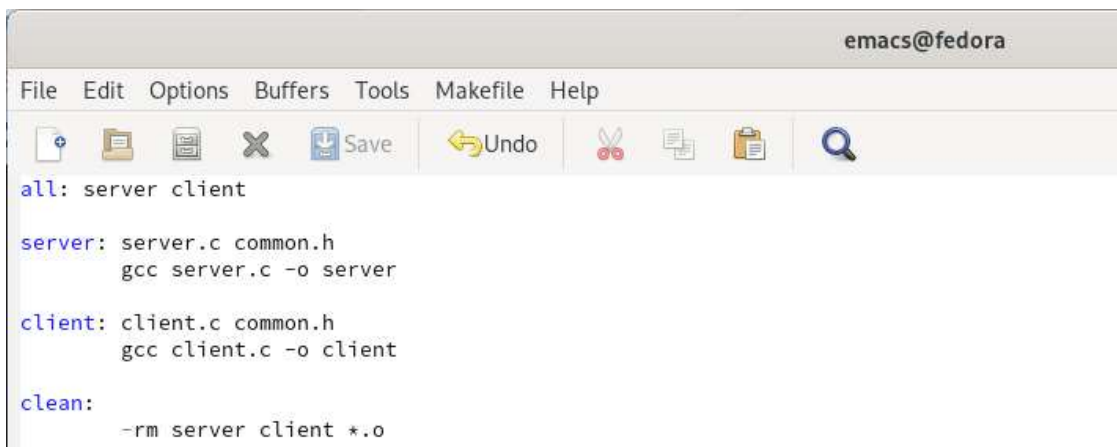
        /* передадим сообщение серверу */
        msglen = strlen(text);
        if(write(writefd, text, msglen)!= msglen)
        {
            fprintf(stderr, "%s: ошибка записи в FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-2);
        }

        /*приостановка работы клиента на 5 секунд*/
        sleep(5);
    }

    /*Закроем доступ к FIFO*/
    close(writefd);
    exit(0);
}
```

Программа в файле *client.c*

- Makefile (файл для сборки) не изменяли (рис. -@fig:007 ).



The screenshot shows the Emacs editor window titled "emacs@fedora". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Makefile", and "Help". The toolbar contains icons for file operations and editing. The main text area displays a Makefile with the following content:

```
all: server client

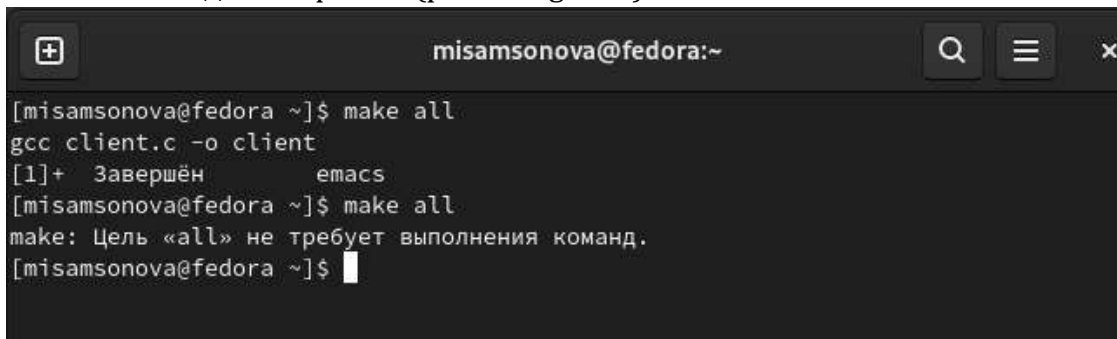
server: server.c common.h
    gcc server.c -o server

client: client.c common.h
    gcc client.c -o client

clean:
    -rm server client *.o
```

### Программа в Makefile

3. После написания кодов, используя команду «make all», скомпилировали необходимые файлы (рис. -@fig:008 ).

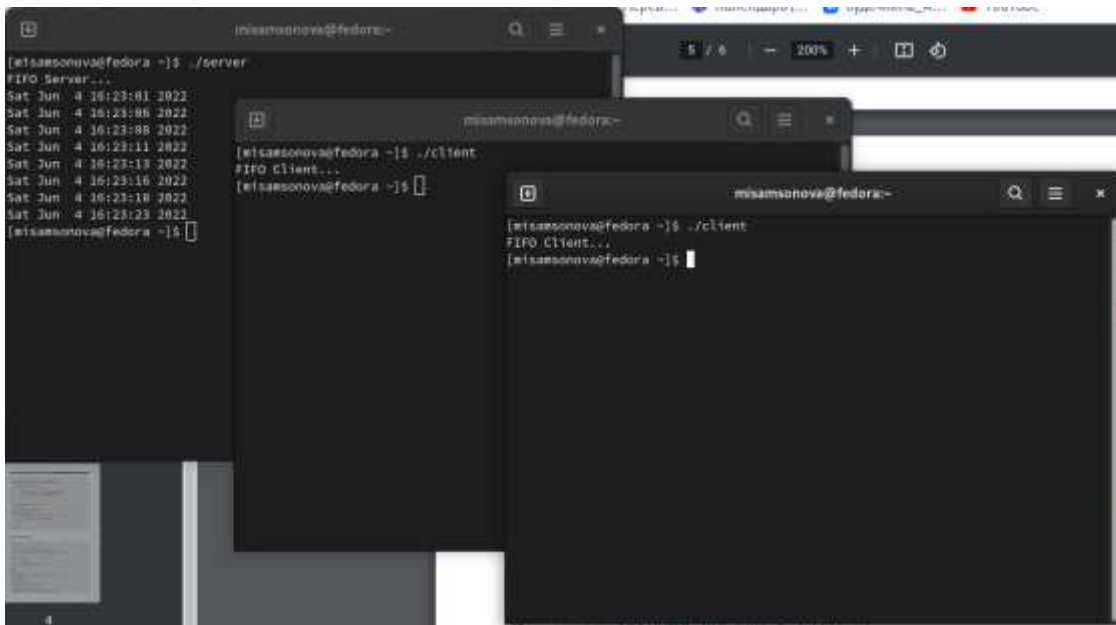


The screenshot shows a terminal window titled "misamsonova@fedora:~". The terminal output is as follows:

```
[misamsonova@fedora ~]$ make all
gcc client.c -o client
[1]+  Завершён      emacs
[misamsonova@fedora ~]$ make all
make: Цель «all» не требует выполнения команд.
[misamsonova@fedora ~]$
```

### Команда make all

- Далее проверили работу написанного кода. Открыли 3 консоли (терминала) и запустили: в первом терминале – «./server», в остальных двух – «./client». В результате каждый терминал-клиент вывел по 4 сообщения. Спустя 30 секунд работа сервера была прекращена (рис. -@fig:009 ). Программа работает корректно.



### *Команда take all*

- Также отдельно проверили длительность работы сервера, введя команду «./server» в одном терминале. Он завершил свою работу через 30 секунд (рис. - @fig:010 ). Если сервер завершит свою работу, не закрыв канал, то, когда мы будем запускать этот сервер снова, появится ошибка «Невозможно создать FIFO», так как у нас уже есть один канал.

```

misamsonova@fedora:~
[ misamsonova@fedora ~]$ ./server
FIFO Server...
Sat Jun  4 16:23:01 2022
Sat Jun  4 16:23:06 2022
Sat Jun  4 16:23:08 2022
Sat Jun  4 16:23:11 2022
Sat Jun  4 16:23:13 2022
Sat Jun  4 16:23:16 2022
Sat Jun  4 16:23:18 2022
Sat Jun  4 16:23:23 2022

```

### *Проверка длительности работы сервера*

## **Вывод выполнения лабораторной работы №14**

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