

# Лабораторная работа № 14.

---

Махорин Иван Сергеевич

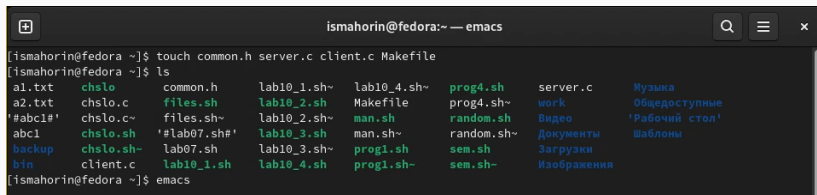
2022, 2 июня

RUDN, Москва, Россия

## Именованные каналы

---

# Создание файлов и переход в emacs



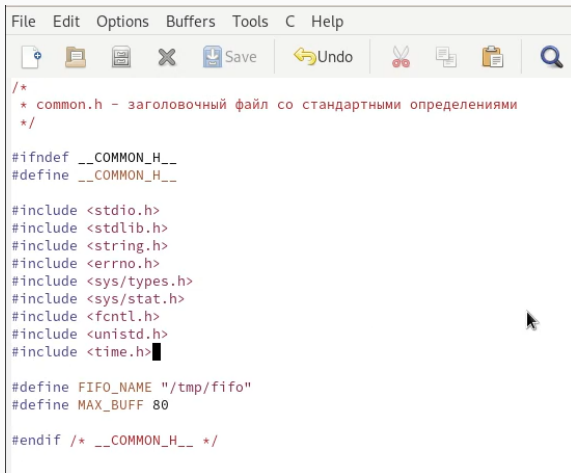
The image shows a terminal window titled "ismahorin@fedora:~ — emacs". The user has executed the command `touch common.h server.c client.c Makefile` to create four files in the home directory. Subsequently, the user has run `ls`, which displays a long listing of files and directories. The output of `ls` is as follows:

a1.txt	chsl0	common.h	lab10_1.sh~	lab10_4.sh~	prog4.sh	server.c	Музыка
a2.txt	chsl0.c	files.sh	lab10_2.sh	Makefile	prog4.sh~	work	Общедоступные
'#abc1#'	chsl0.c~	files.sh~	lab10_2.sh~	man.sh	random.sh	Видео	'Рабочий стол'
abc1	chsl0.sh	'#lab07.sh#'	lab10_3.sh	man.sh~	random.sh~	Документы	Шаблоны
backup	chsl0.sh~	lab07.sh	lab10_3.sh~	prog1.sh	sem.sh	Загрузки	
bin	client.c	lab10_1.sh	lab10_4.sh	prog1.sh~	sem.sh~	Изображения	

The prompt `[ismahorin@fedora ~]$ emacs` is visible at the bottom of the terminal window.

Figure 1: Создание файлов в домашнем каталоге

# Первый скрипт



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

#ifndef __COMMON_H__
#define __COMMON_H__

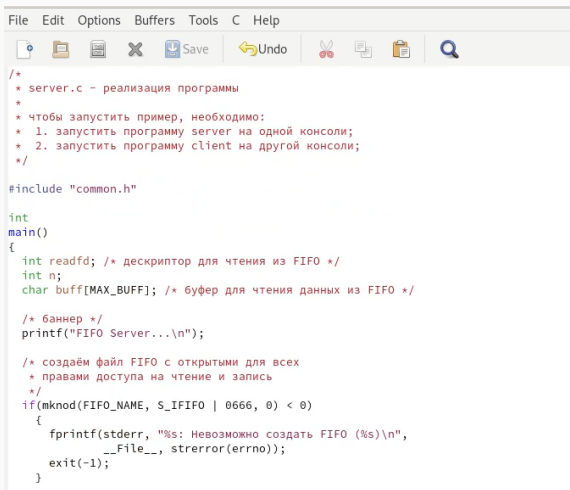
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <time.h>

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

#endif /* __COMMON_H__ */
```

Figure 2: Перенос скрипта для common.h

## Второй скрипт



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

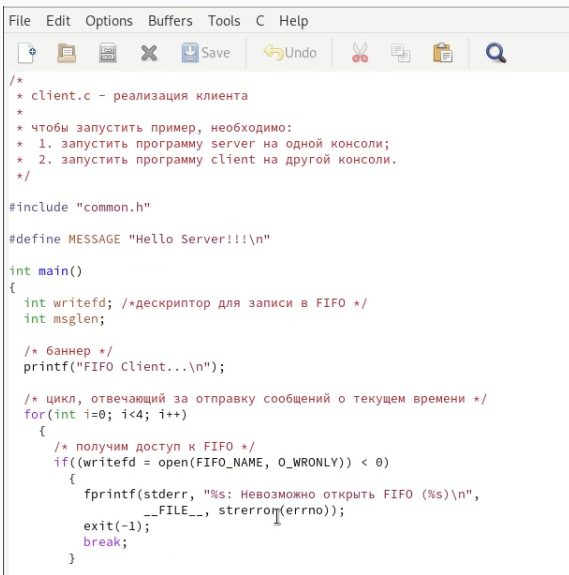
#include "common.h"

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");

    /* создаём 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);
    }
}
```

Figure 3: Перенос скрипта для server.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;

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

    /* цикл, отвечающий за отправку сообщений 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;
        }
    }
```

Figure 4: Перенос скрипта для main.c



The image shows a screenshot of a text editor window. The title bar is not visible. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Makefile', and 'Help'. The toolbar contains icons for 'New' (a document with a plus sign), 'Open' (a folder), 'Save' (a floppy disk), 'Close' (an 'X'), 'Save As' (a floppy disk with a plus sign), 'Undo' (a curved arrow), 'Cut' (scissors), 'Copy' (two overlapping documents), 'Paste' (a document with a plus sign), and 'Find' (a magnifying glass). The text area contains a Makefile script with the following content:

```
all: server client

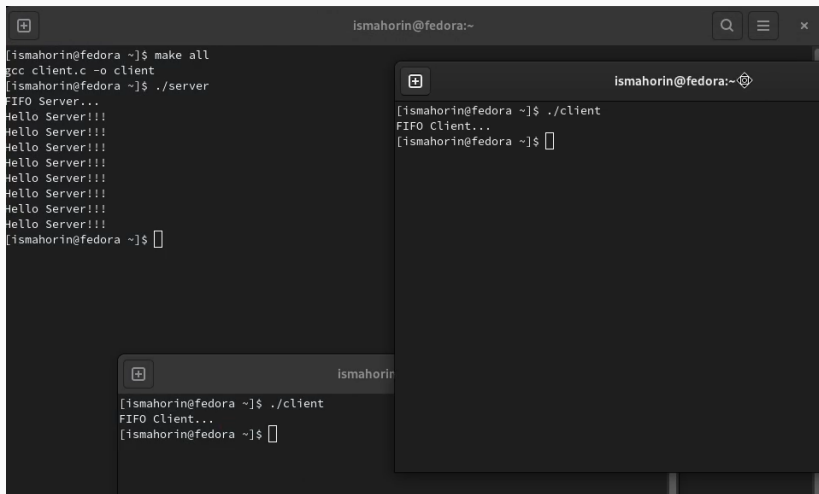
server: server.c common.h
    gcc server.c -o server

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

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

Figure 5: Перенос скрипта для Makefile

# Проверка выполнения



The image displays three terminal windows from a Fedora system, illustrating the execution of a FIFO-based communication program. The main window shows the compilation and execution of the server, while two smaller windows show the execution of the client.

```
ismahorin@fedora:~  
[ismahorin@fedora ~]$ make all  
gcc client.c -o client  
[ismahorin@fedora ~]$ ./server  
FIFO Server...  
Hello Server!!!  
Hello Server!!!  
Hello Server!!!  
Hello Server!!!  
Hello Server!!!  
Hello Server!!!  
Hello Server!!!  
Hello Server!!!  
[ismahorin@fedora ~]$  
  
ismahorin@fedora:~  
[ismahorin@fedora ~]$ ./client  
FIFO Client...  
[ismahorin@fedora ~]$  
  
ismahorin@fedora:~  
[ismahorin@fedora ~]$ ./client  
FIFO Client...  
[ismahorin@fedora ~]$
```

Figure 6: Проверка выполнения



## Выводы

---

В ходе выполнения лабораторной работы были приобретены практические навыки работы с именованными каналами.

Спасибо за внимание!