

# Лабораторная работа 14 Именованные каналы

---

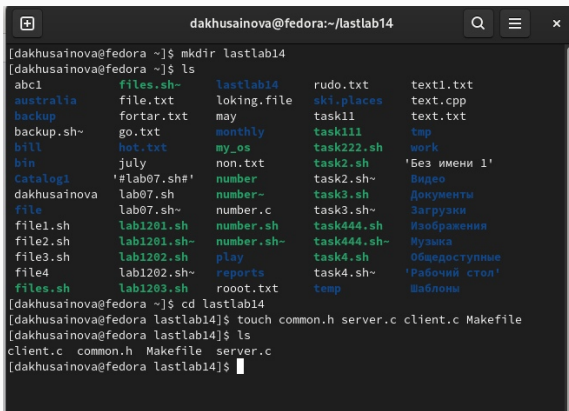
Хусаинова Динара Айратовна

02.06.2022

RUDN

Приобретение практических навыков работы с именованными каналами.

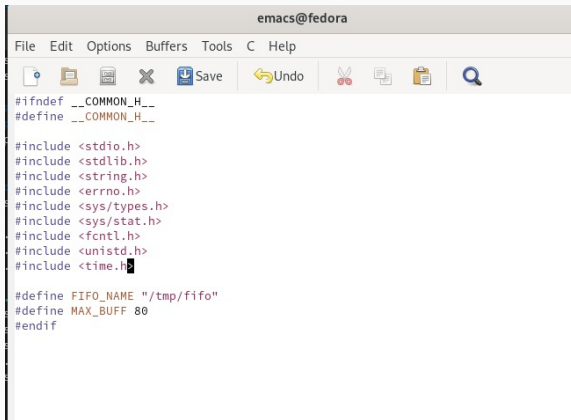
Создаем необходимые файлы(рис. 1).



```
dakhusainova@fedora:~/lastlab14
[dakhusainova@fedora ~]$ mkdir lastlab14
[dakhusainova@fedora ~]$ ls
abcl          files.sh~    lastlab14   rudo.txt     text1.txt
australia     file.txt     loking.file ski.places   text.cpp
backup        fortar.txt   may         task11       text.txt
backup.sh~    go.txt      monthly     task111      tmp
bill          hot.txt     my_os      task222.sh   work
bin           july       non.txt    task2.sh     'Без имени 1'
Catalog1     '#lab07.sh#' number      task2.sh~    Видео
dakhusainova lab07.sh    number~    task3.sh     Документы
file         lab07.sh~  number.c   task3.sh~    Загрузки
file1.sh     lab1201.sh number.sh   task444.sh   Изображения
file2.sh     lab1201.sh~ number.sh~  task444.sh~  Музыка
file3.sh     lab1202.sh play        task4.sh     Общедоступные
file4        lab1202.sh~ reports     task4.sh~    'Рабочий стол'
files.sh     lab1203.sh rooot.txt   temp         Шаблоны
[dakhusainova@fedora ~]$ cd lastlab14
[dakhusainova@fedora lastlab14]$ touch common.h server.c client.c Makefile
[dakhusainova@fedora lastlab14]$ ls
client.c common.h Makefile server.c
[dakhusainova@fedora lastlab14]$
```

Figure 1: Создание файлов и каталога

В файле common.h добавляем стандартные заголовочные файлы `unistd.h` и `time.h`, необходимые для работы кодов других файлов (рис. 2).



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

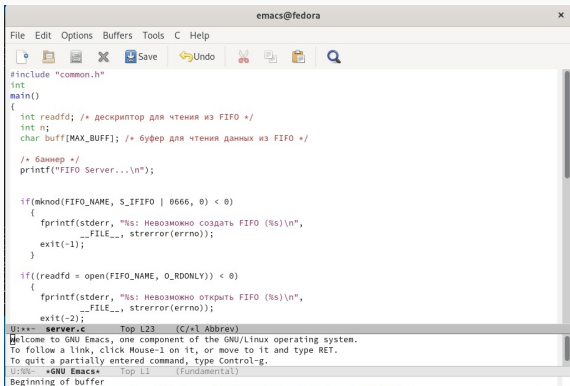
#ifndef __COMMON_H__
#define __COMMON_H__

#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
```

Figure 2: common.h

В файл server.c добавляем цикл while для контроля за временем работы сервера(рис. 3).



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

#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");

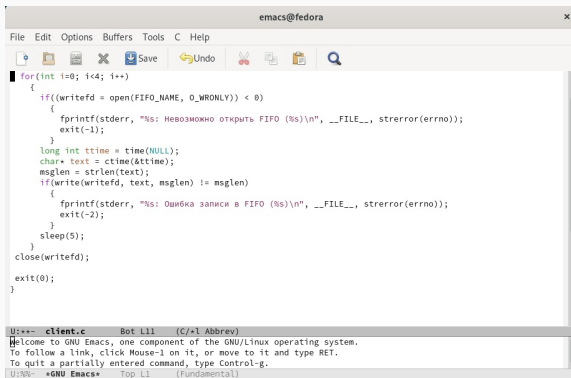
    if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
    {
        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
            __FILE__, strerror(errno));
        exit(-1);
    }

    if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
    {
        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
            __FILE__, strerror(errno));
        exit(-2);
    }

    U:*** server.c Top L23 (C/*1 Abbrev)
    Welcome to GNU Emacs, one component of the GNU/Linux operating system.
    To follow a link, click Mouse-1 on it, or move to it and type RET.
    To quit a partially entered command, type Control-g.
    U:*** GNU Emacs Top L1 (Fundamental)
    Beginning of buffer
```

Figure 3: server.c

В файл client.c добавляем цикл, который отвечает за количество сообщений о текущем времени, которое получается в результате выполнения команд, и команду sleep(5) для приостановки работы клиента на 5 секунд(рис. 4).



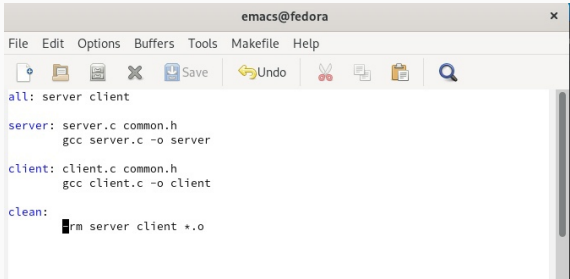
```
emacs@fedora
File Edit Options Buffers Tools C Help
[Icons: Open, Save, Undo, Redo, Find, etc.]

for(int i=0; i<4; i++)
{
    if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
    {
        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n", __FILE__, strerror(errno));
        exit(-1);
    }
    long int ttime = time(NULL);
    char* text = ctime(&ttime);
    msglen = strlen(text);
    if(write(writefd, text, msglen) != msglen)
    {
        fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n", __FILE__, strerror(errno));
        exit(-2);
    }
    sleep(5);
}
close(writefd);
exit(0);
}
```

U:--- client.c Bot L11 (C/\*l Abbrev)  
Welcome to GNU Emacs, one component of the GNU/Linux operating system.  
To follow a link, click Mouse-1 on it, or move to it and type RET.  
To quit a partially entered command, type Control-g.  
U:GNU Emacs Top L1 (Fundamental)

Figure 4: client.c

Makefile оставляем без изменений( рис. 5).

A screenshot of the Emacs editor window titled 'emacs@fedora'. The window has a menu bar with 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Makefile', and 'Help'. Below the menu bar is a toolbar with icons for file operations and editing. The main text area contains 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
```

Figure 5: Makefile

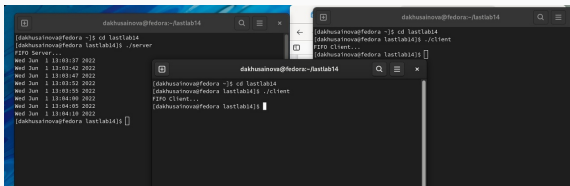
После написания кода, мы используем команду `make all`, чтобы скомпилировать все файлы( рис. 6).

```
[dakhusainova@fedora lastlab14]$ make all  
gcc server.c -o server  
gcc client.c -o client  
[dakhusainova@fedora lastlab14]$
```

Figure 6: Компиляция



Проверяем работу файлов( рис. 7).



The image shows three terminal windows from a Fedora system, demonstrating the execution of a FIFO (Named Pipe) program. The windows are titled 'dakhunainova@fedora--lastlab14'.

- Top-left window:** Shows the execution of the server program. The prompt is `[dakhunainova@fedora ~]$ cd lastlab14`. The user enters `[dakhunainova@fedora lastlab14]$ ./server`. The output shows the server running and receiving data from the client at various times on June 13, 2022.
- Top-right window:** Shows the execution of the client program. The prompt is `[dakhunainova@fedora ~]$ cd lastlab14`. The user enters `[dakhunainova@fedora lastlab14]$ ./client`. The output shows the client sending data to the server.
- Bottom window:** Shows the execution of the client program. The prompt is `[dakhunainova@fedora ~]$ cd lastlab14`. The user enters `[dakhunainova@fedora lastlab14]$ ./client`. The output shows the client sending data to the server.

Figure 7: Проверка работы файлов

Я приобрела практические навыки работы с именованными каналами.