

# lab14

2 Jun, 2022 Moscow, Russia

## Цель работы

Приобретение практических навыков работы с именованными каналами.

## Задача

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?

## Выполнение лабораторной работы

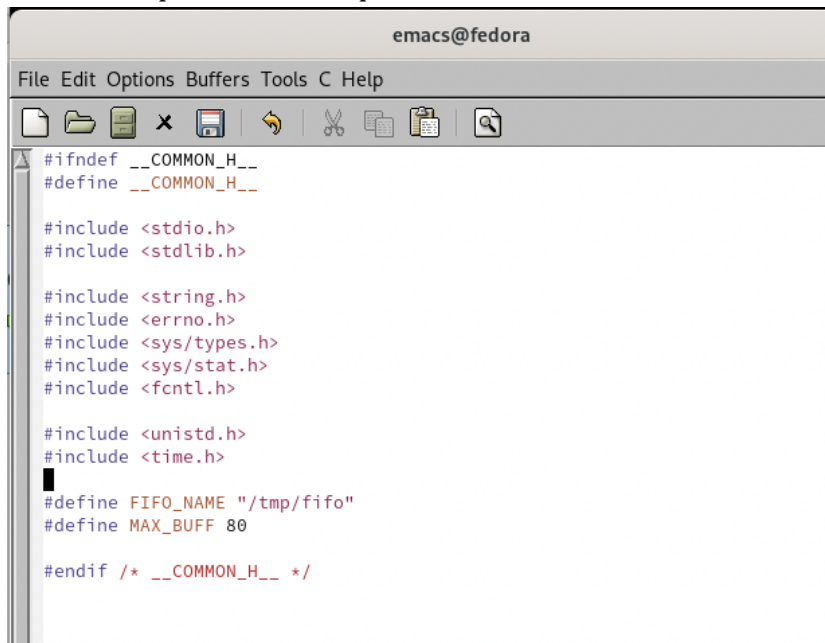
1. Я создала файлы `common.h`, `server.c`, `client.c`, `Makefile`.



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

Рис 1. Создание файлов

## 2. Открыла емакс и реализовала код.

A screenshot of the Emacs editor window titled 'emacs@fedora'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'C', and 'Help'. The toolbar contains icons for file operations and editing. The main text area displays the content of a file named 'common.h'. The code is as follows:

```
#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__ */
```

*Рис. 2. common.h*

```
#include "common.h"

int main()
{
    int readfd; /* дескриптор для чтения из FIFO */
    int n;
    char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */
    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);
    }
    clock_t start = time(NULL);
    while (time(NULL)-start < 30){
        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 */
    if(unlink(FIFO_NAME) < 0)
    {
        fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
            __FILE__, strerror(errno));
        exit(-4);
    }
    exit(0);
}
```

Рис 3. server.c

```

#include "common.h"
#define MESSAGE "Hello Server!!!\n"

int
main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;
    printf("FIFO Client...\n");
    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);
            break;
        }
        long int ttime=time(NULL)
        char *text=ctime(&ttime)

        msglen = strlen(MESSAGE);
        if(write(writefd, MESSAGE, msglen) != msglen)
        {
            fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-2);
        }
        sleep(5);
    }
    close(writefd);
    exit(0);
}

```

*Рис 4. client.c*

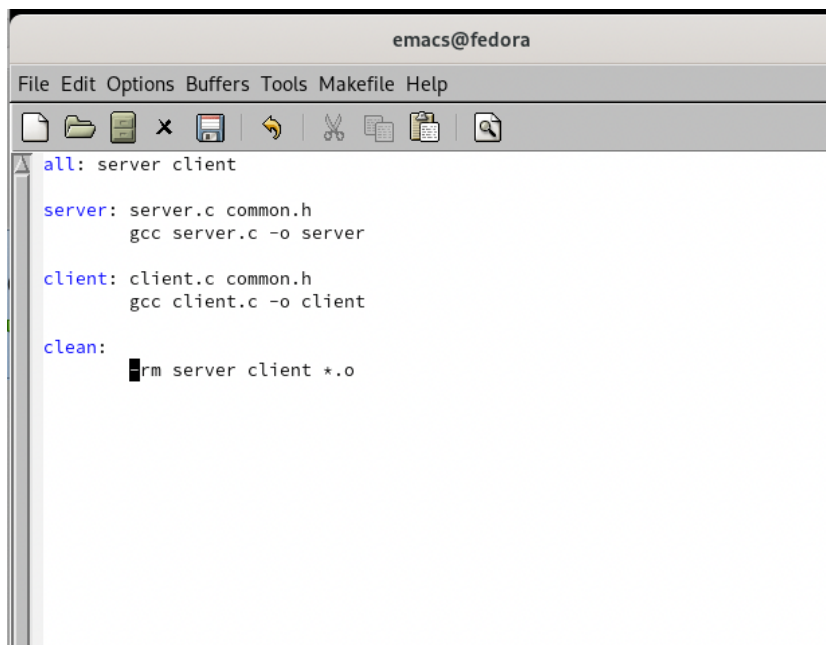


Рис 5. Makefile

### 3. Скомпилировала файл

```
[epmitichkina@fedora 14]$ make all
gcc server.c -o server
gcc client.c -o client
```

Рис 6. Компиляция программы

### 4. Проверила работу файлов

```
[epmitichkina@fedora 14]$ ./server
FIFO Server...
Hello Server!!!
[epmitichkina@fedora 14]$
```

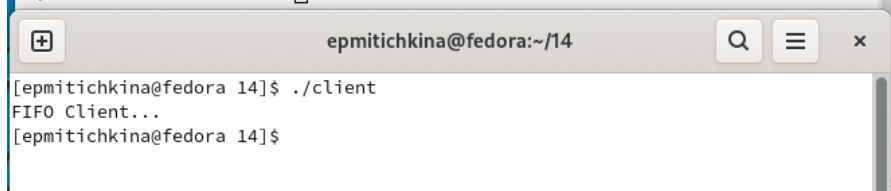


Рис 7. Проверка

## Выводы

В результате работы приобрела практические навыки работы с именованными каналами.