

---

# Front matter

---

lang: ru-RU

title: "Отчёт по лабораторной работе №15"

subtitle: "Операционные системы"

author: "Бирюкова Анастасия Анатольевна"

## Formatting

---

toc-title: "Содержание"

toc: true # Table of contents

toc\_depth: 2

lof: true # List of figures

lot: true # List of tables

fontsize: 12pt

linestretch: 1.5

papersize: a4paper

documentclass: scrreprt

polyglossia-lang: russian

polyglossia-otherlangs: english

mainfont: PT Serif

romanfont: PT Serif

sansfont: PT Sans

monofont: PT Mono

mainfontoptions: Ligatures=TeX

romanfontoptions: Ligatures=TeX

sansfontoptions: Ligatures=TeX,Scale=MatchLowercase

monofontoptions: Scale=MatchLowercase

indent: true

pdf-engine: lualatex

header-includes:

- \linepenalty=10 # the penalty added to the badness of each line within a paragraph (no associated penalty node) Increasing the value makes tex try to have fewer lines in the paragraph.

- `\interlinepenalty=0` # value of the penalty (node) added after each line of a paragraph.
  - `\hyphenpenalty=50` # the penalty for line breaking at an automatically inserted hyphen
  - `\exhyphenpenalty=50` # the penalty for line breaking at an explicit hyphen
  - `\binoppenalty=700` # the penalty for breaking a line at a binary operator
  - `\relpenalty=500` # the penalty for breaking a line at a relation
  - `\clubpenalty=150` # extra penalty for breaking after first line of a paragraph
  - `\widowpenalty=150` # extra penalty for breaking before last line of a paragraph
  - `\displaywidowpenalty=50` # extra penalty for breaking before last line before a display math
  - `\brokenpenalty=100` # extra penalty for page breaking after a hyphenated line
  - `\predisplaypenalty=10000` # penalty for breaking before a display
  - `\postdisplaypenalty=0` # penalty for breaking after a display
  - `\floatingpenalty = 20000` # penalty for splitting an insertion (can only be split footnote in standard LaTeX)
  - `\raggedbottom` # or `\flushbottom`
  - `\usepackage{float}` # keep figures where there are in the text
  - `\floatplacement{figure}{H}` # keep figures where there are in the text
- 

## Цель работы

---

Приобретение практических навыков работы с именованными каналами.

## Задание

---

Научиться работать с именованными каналами

## Выполнение лабораторной работы

---

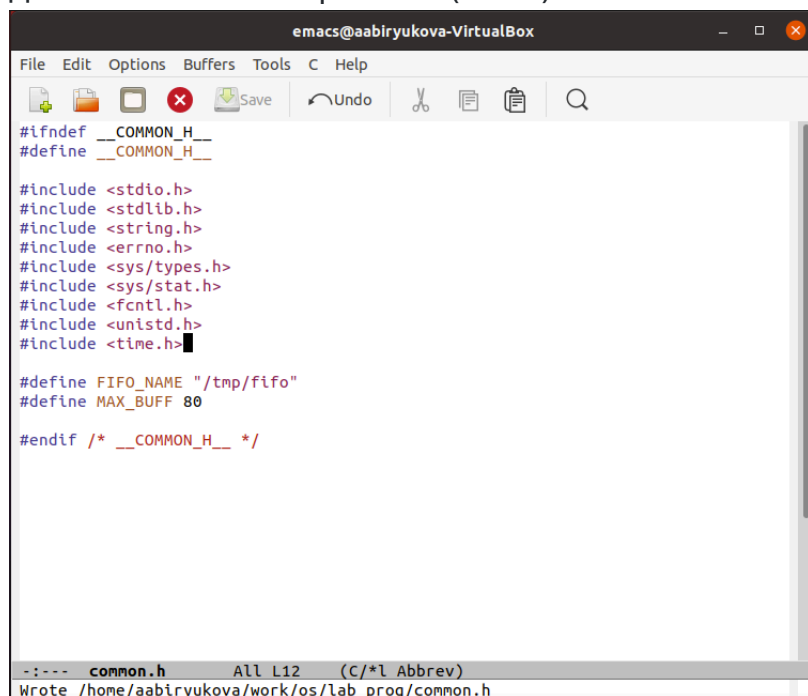
Для начала я создала необходимые файлы с помощью команды «touch» и открыла редактор emacs для их редактирования. (Рис.1-2)

```
aabiryukova@aabiryukova-VirtualBox:~/work/os/lab_prog$ touch server.c
```

```
aabiryukova@aabiryukova-VirtualBox:~/work/os/lab_prog$ touch client.c
```

Рис.1-2

Далее я изменила 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. (Рис3)



```

emacs@aabiryukova-VirtualBox
File Edit Options Buffers Tools C Help
Save Undo
#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

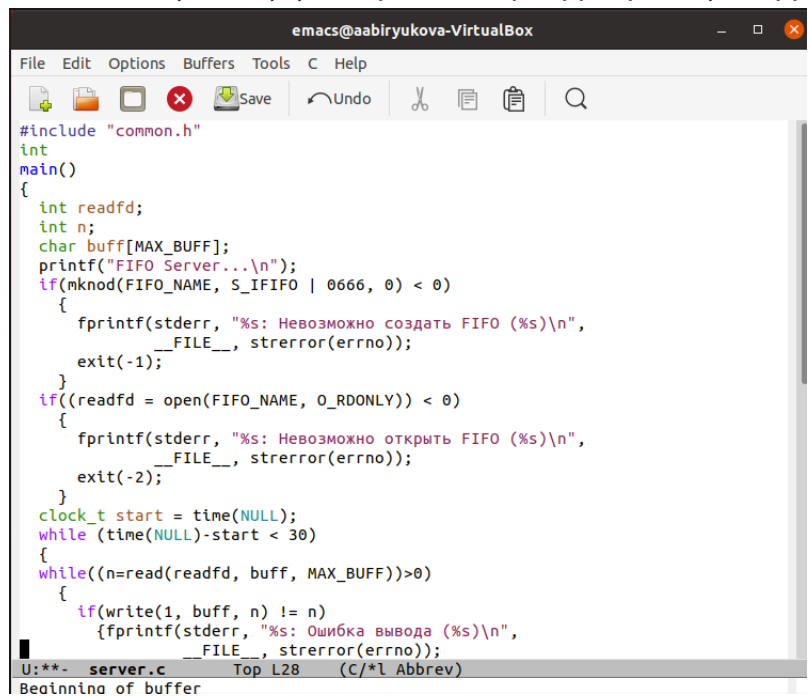
#endif /* __COMMON_H__ */

-:--- common.h All L12 (C/*l Abbrev)
Wrote /home/aabiryukova/work/os/lab_prog/common.h

```

Рис3

В файл `server.c` добавила 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 (Рис4-5)



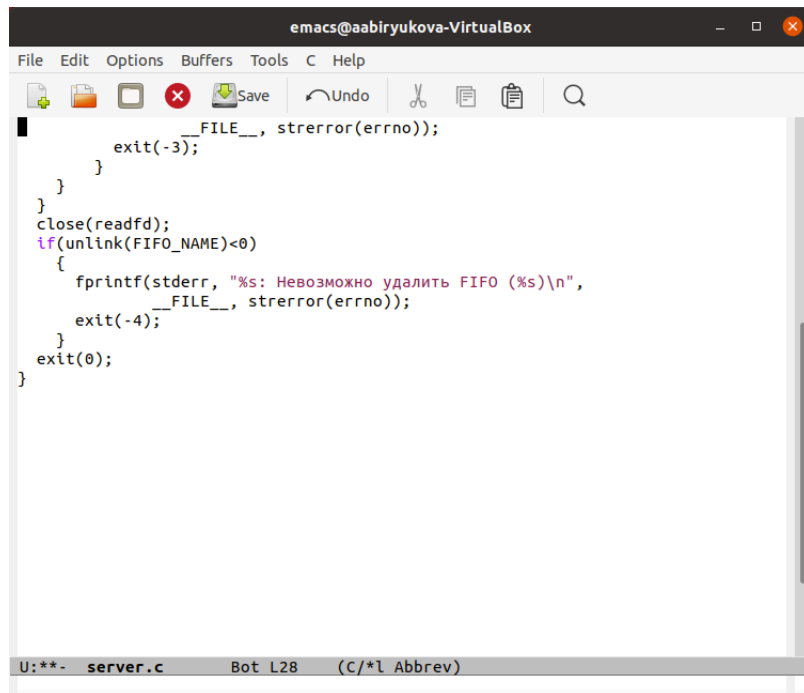
```

emacs@aabiryukova-VirtualBox
File Edit Options Buffers Tools C Help
Save Undo
#include "common.h"
int
main()
{
    int readfd;
    int n;
    char buff[MAX_BUFF];
    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));
            }
        }
    }
}

U:*** server.c Top L28 (C/*l Abbrev)
Beginning of buffer

```

Рис4



```

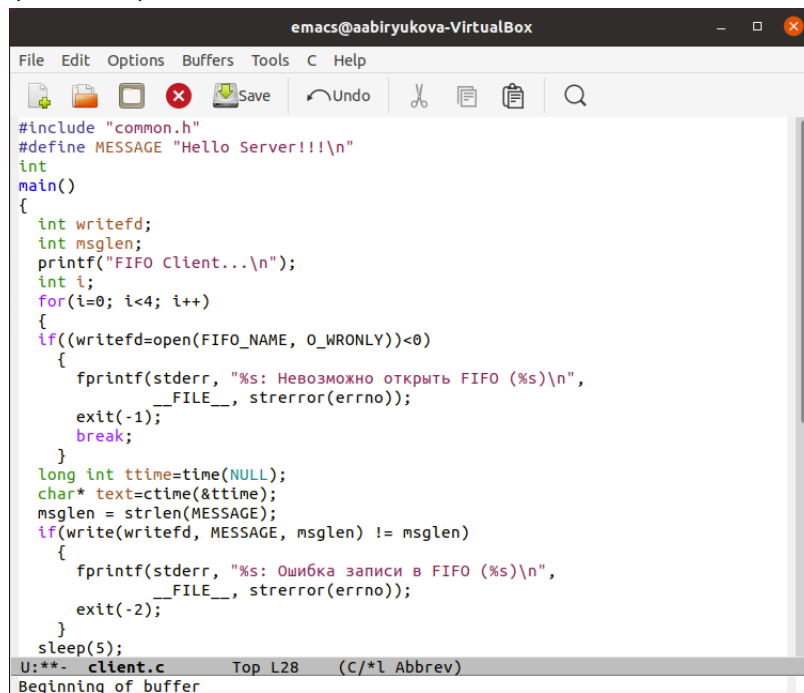
    __FILE__, strerror(errno));
    exit(-3);
}
}
close(readfd);
if(unlink(FIFO_NAME)<0)
{
    fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
        __FILE__, strerror(errno));
    exit(-4);
}
exit(0);
}

```

U:\*\*- server.c Bot L28 (C/\*l Abbrev)

Рис5

В файл client.c добавила цикл, который отвечает за количество сообщений о текущем времени, и команду sleep для приостановки работы (Рис6-7)



```

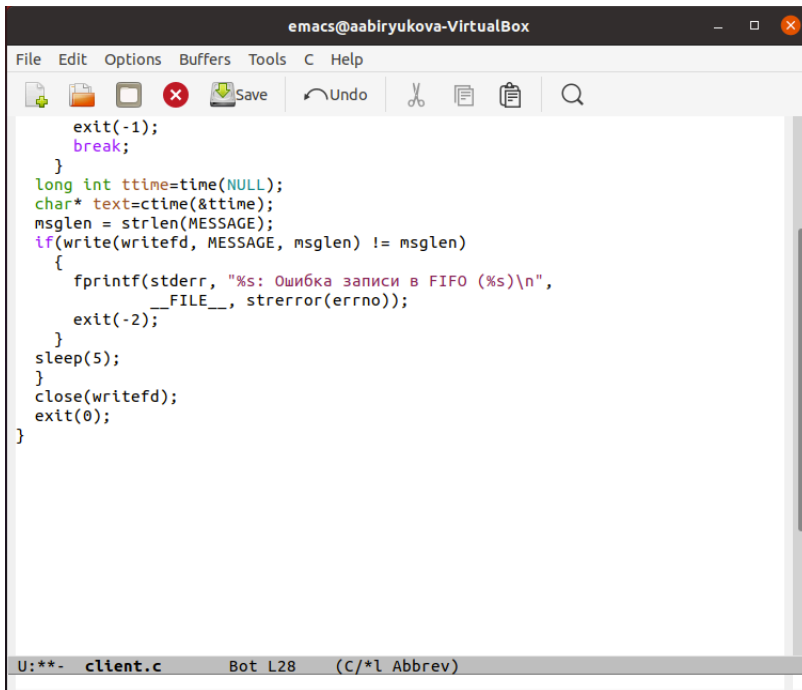
#include "common.h"
#define MESSAGE "Hello Server!!!\n"
int
main()
{
    int writefd;
    int msglen;
    printf("FIFO Client...\n");
    int i;
    for(i=0; i<4; i++)
    {
        if((writefd=open(FIFO_NAME, O_WRONLY))<0)
        {
            fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
                __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);
    }
}

```

U:\*\*- client.c Top L28 (C/\*l Abbrev)

Beginning of buffer

Рис6



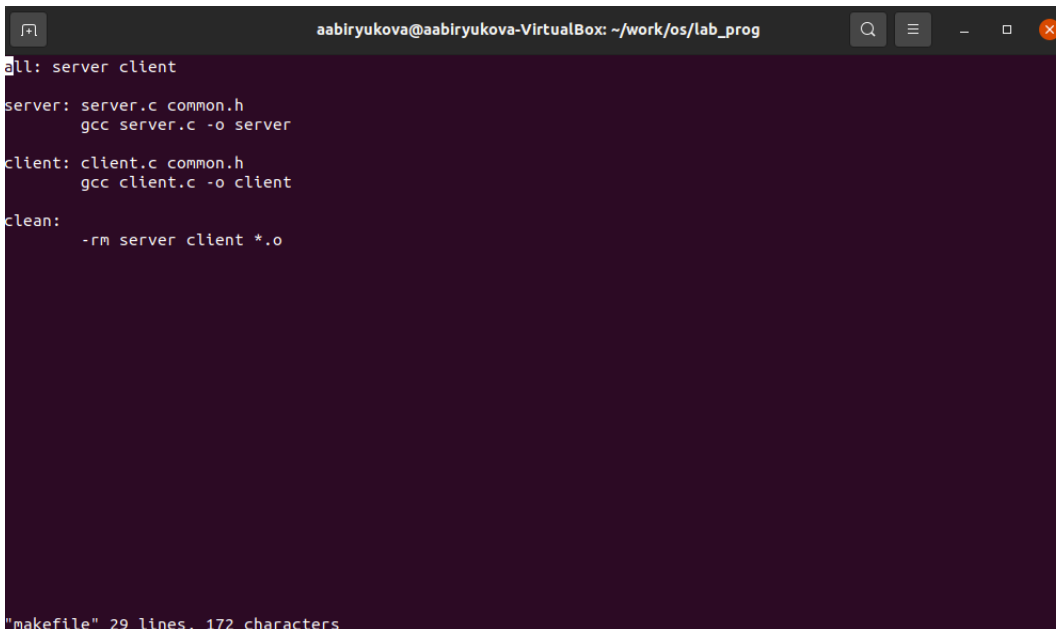
The screenshot shows an Emacs editor window titled 'emacs@aabiryukova-VirtualBox'. The menu bar includes File, Edit, Options, Buffers, Tools, C, and Help. The toolbar contains icons for saving, undo, redo, and search. The code in the buffer is as follows:

```
    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);
}
```

The status bar at the bottom indicates the current file is 'client.c' at line 28, with C/\*l abbreviations.

Рис7

## Makefile (Рис8)



The screenshot shows a terminal window titled 'aabiryukova@aabiryukova-VirtualBox: ~/work/os/lab\_prog'. The terminal displays the following Makefile content:

```
all: server client

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

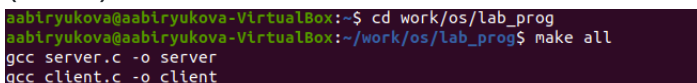
client: client.c common.h
    gcc client.c -o client

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

At the bottom of the terminal, it says '"makefile" 29 lines, 172 characters'.

Рис8

После написания кодов, я использовала команду «make all» и скомпилировала файлы (Рис9)



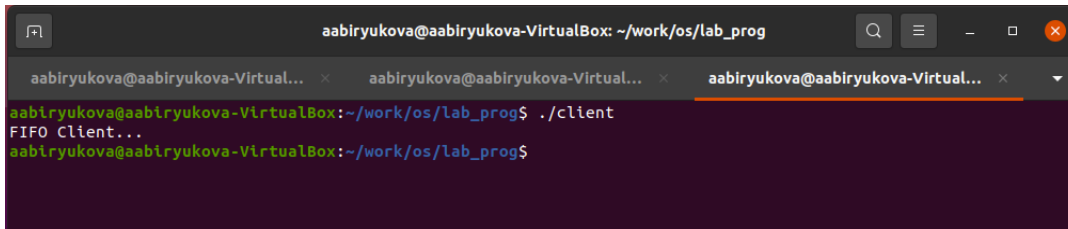
The screenshot shows a terminal window with the following commands and output:

```
aabiryukova@aabiryukova-VirtualBox:~$ cd work/os/lab_prog
aabiryukova@aabiryukova-VirtualBox:~/work/os/lab_prog$ make all
gcc server.c -o server
gcc client.c -o client
```

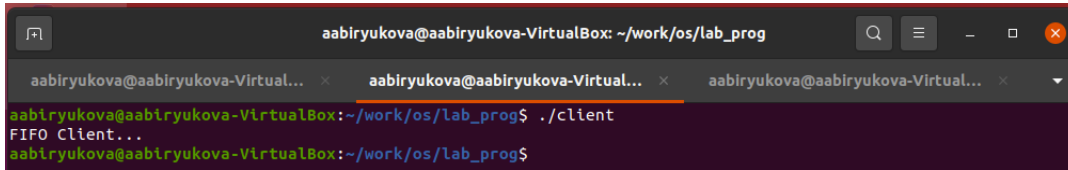
Рис9

Далее я проверила работу написанного кода. Открыла 3 консоли (терминала)

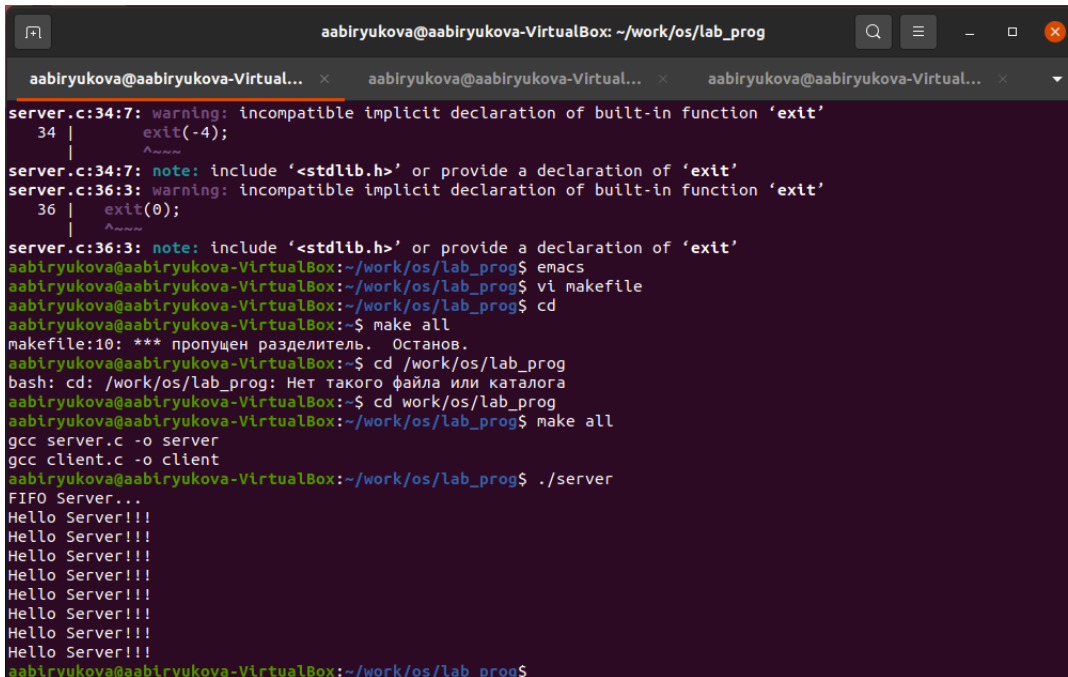
и запустила: в первом терминале – «./server», в остальных двух – «./client».



```
aabiryukova@aabiryukova-VirtualBox: ~/work/os/lab_prog
aabiryukova@aabiryukova-VirtualBox:~/work/os/lab_prog$ ./client
FIFO Client...
aabiryukova@aabiryukova-VirtualBox:~/work/os/lab_prog$
```



```
aabiryukova@aabiryukova-VirtualBox: ~/work/os/lab_prog
aabiryukova@aabiryukova-VirtualBox:~/work/os/lab_prog$ ./client
FIFO Client...
aabiryukova@aabiryukova-VirtualBox:~/work/os/lab_prog$
```



```
aabiryukova@aabiryukova-VirtualBox: ~/work/os/lab_prog
server.c:34:7: warning: incompatible implicit declaration of built-in function 'exit'
 34 |     exit(-4);
    |     ^~~~~~
server.c:34:7: note: include '<stdlib.h>' or provide a declaration of 'exit'
server.c:36:3: warning: incompatible implicit declaration of built-in function 'exit'
 36 |     exit(0);
    |     ^~~~~~
server.c:36:3: note: include '<stdlib.h>' or provide a declaration of 'exit'
aabiryukova@aabiryukova-VirtualBox:~/work/os/lab_prog$ emacs
aabiryukova@aabiryukova-VirtualBox:~/work/os/lab_prog$ vi makefile
aabiryukova@aabiryukova-VirtualBox:~/work/os/lab_prog$ cd
aabiryukova@aabiryukova-VirtualBox:~$ make all
makefile:10: *** пропущен разделитель. Останов.
aabiryukova@aabiryukova-VirtualBox:~$ cd /work/os/lab_prog
bash: cd: /work/os/lab_prog: Нет такого файла или каталога
aabiryukova@aabiryukova-VirtualBox:~$ cd work/os/lab_prog
aabiryukova@aabiryukova-VirtualBox:~/work/os/lab_prog$ make all
gcc server.c -o server
gcc client.c -o client
aabiryukova@aabiryukova-VirtualBox:~/work/os/lab_prog$ ./server
FIFO Server...
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
aabiryukova@aabiryukova-VirtualBox:~/work/os/lab_prog$
```

В результате каждый терминал- клиент вывел сообщения. Спустя 30 секунд работа сервера была прекращена. Программа работает без ошибок.

## Вывод

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