

# **Отчет по лабораторной работе №14**

**Операционные системы**

Ничипорова Елена Дмитриевна

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
	Список литературы	11

## Список иллюстраций

2.1	Создание файлов . . . . .	6
2.2	Программа в файле common.h . . . . .	7
2.3	Программа в файле server.c . . . . .	7
2.4	Программа в файле server.c . . . . .	8
2.5	Программа в файле client.c . . . . .	8
2.6	Программа в файле client.c . . . . .	9
2.7	Команда make all . . . . .	9
2.8	Проверка работы . . . . .	10

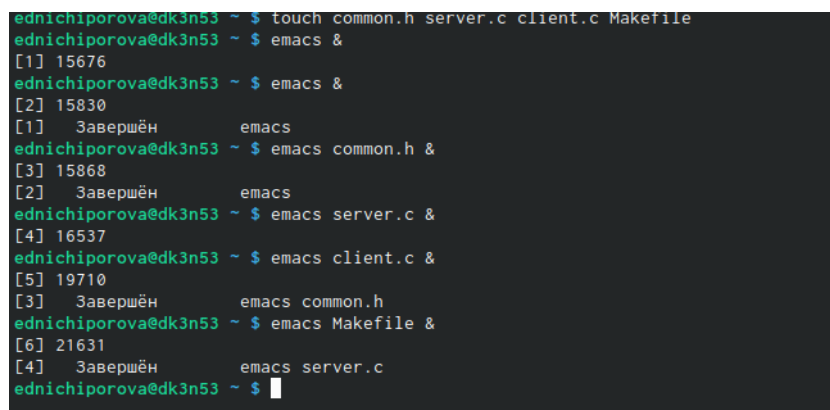
## Список таблиц

# **1 Цель работы**

Приобретение практических навыков работы с именованными каналами.

## 2 Выполнение лабораторной работы

- В начале я создала необходимые файлы и открыла их в редакторе emacs(рис. 2.1)



```
ednichiporova@dk3n53 ~ $ touch common.h server.c client.c Makefile
ednichiporova@dk3n53 ~ $ emacs &
[1] 15676
ednichiporova@dk3n53 ~ $ emacs &
[2] 15830
[1] Завершён      emacs
ednichiporova@dk3n53 ~ $ emacs common.h &
[3] 15868
[2] Завершён      emacs
ednichiporova@dk3n53 ~ $ emacs server.c &
[4] 16537
ednichiporova@dk3n53 ~ $ emacs client.c &
[5] 19710
[3] Завершён      emacs common.h
ednichiporova@dk3n53 ~ $ emacs Makefile &
[6] 21631
[4] Завершён      emacs server.c
ednichiporova@dk3n53 ~ $
```

Рис. 2.1: Создание файлов

- 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 других файлов. Этот файл предназначен для заголовочных файлов, чтобы не прописывать их в других программах каждый раз.(рис. 2.2)

```

/*
 * 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__ */

```

Рис. 2.2: Программа в файле common.h

- В файл server.c добавила цикл while для контроля за временем работы сервера. Разница между временем работы time(NULL) и временем начала работы не должна превышать 30 секунд.(рис. 2.3)(рис. 2.4)

```

/*
 * 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;
    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 | O_NONBLOCK))< 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);
            }
        }
    }
}

```

Рис. 2.3: Программа в файле server.c

```

        exit(-3);
    }
}

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

exit(0);
}

```

Рис. 2.4: Программа в файле server.c

- В файле client.c добавила цикл, который отвечает за количество сообщений 0 текущем времени (4 сообщения), которое получается в результате выполнения командна Рисунке 7 и команду sleep(5) для приостановки работы клиента на 5 секунд (рис. 2.5)(рис. 2.6)

```

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

        sleep(5);
    }
}

```

Рис. 2.5: Программа в файле client.c



```

    sleep(5);
}
close (writefd);
exit(0);
}

```

Рис. 2.6: Программа в файле client.c

- Мекеfile, то есть файл для сборки, я оставила без изменения
- После написания кодов, я использую команду “make all” для компиляции необходимых файлов(рис. 2.7)

```

ednichiporova@dk3n53 ~ $ make all
gcc client.c -o client
client.c: 8 функции «main»:
client.c:30:16: ошибка: expected «=», «,», «;», «asm» or «__attribute__» before «-» token
   30 |     char* text=ctime(&time);
      |                ^
client.c:30:16: ошибка: неверный тип аргумента для унарного минуса
client.c:32:21: ошибка: «text» не описан (первое использование в этой функции)
   32 |     msglen= strlen(text);
      |                ^
client.c:32:21: замечание: сообщение о каждом неопisanном идентификаторе выдается один раз в каждой функции, где он встречается
make: *** [Makefile:7: client] Ошибка 1
ednichiporova@dk3n53 ~ $ make all
gcc client.c -o client
ednichiporova@dk3n53 ~ $ make all
make: цель «all» не требует выполнения команд.

```

Рис. 2.7: Команда make all

- Откроем три терминала. В первом окне запустим команду ./server.c, во втором и третьем - ./client.c. В результате каждый терминал-клиент вывел по четыре сообщения о текущем времени. Спустя 30 секунд работа сервера была прекращена. Если клиент завершит свою работу, не закрыв канал, то при повторном запуске сервера, появится ошибка “Невозможно создать FIFO”, так как уже существует один канал.(рис. 2.8)(рис. ??)

```
ednichiporova@dk3n53 ~ $ ./server
FIFO Server...
Thu May 26 14:55:04 2022
Thu May 26 14:55:09 2022
Thu May 26 14:55:14 2022
Thu May 26 14:55:19 2022
ednichiporova@dk3n53 ~ $
```

Рис. 2.8: Проверка работы

```
ednichiporova@dk3n53 ~ $ ./client
FIFO Client...
ednichiporova@dk3n53 ~ $
```

# Выводы

Я приобрела практические навыки работы с именованными каналами.

## **Список литературы**