

Отчет по лабораторной работе №15

Дисциплина: Операционные системы

Чекалова Лилия Руслановна, ст.б. 1032201654

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выводы	10
4	Библиография	11

List of Figures

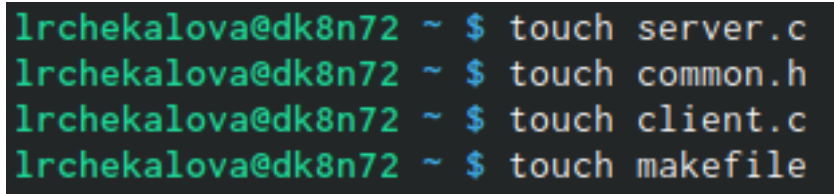
2.1	Создание файлов	5
2.2	Добавление библиотек	6
2.3	Реализация работы сервера (1)	7
2.4	Реализация работы сервера (2)	7
2.5	Реализация работы клиента	8
2.6	Makefile	9
2.7	Компиляция программ	9
2.8	Проверка работы программы	9

1 Цель работы

Приобретение практических навыков работы с именованными каналами.

2 Выполнение лабораторной работы

Создаю файлы client.c, common.h, server.c и makefile командой touch (рис. 2.1)



```
lrchekalova@dk8n72 ~ $ touch server.c
lrchekalova@dk8n72 ~ $ touch common.h
lrchekalova@dk8n72 ~ $ touch client.c
lrchekalova@dk8n72 ~ $ touch makefile
```

Figure 2.1: Создание файлов

Записываю в файл common.h используемые в ходе работы библиотеки из шаблонного файла, добавляю библиотеки time.h и unistd.h (рис. 2.2)

```
#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 <time.h>
#include <unistd.h>

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

#endif
```

Figure 2.2: Добавление библиотек

Описываю в файле `server.c` работу сервера: создание файла FIFO, открытие его на чтение, чтение данных из файла в течение 30 секунд, закрытие файла и удаление его из системы (рис. 2.3) (рис. 2.4)

```

#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));
                exit(-3);
            }
        }
    }
}

```

Figure 2.3: Реализация работы сервера (1)

```

    close(readfd);

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

    exit(0);
}

```

Figure 2.4: Реализация работы сервера (2)

Реализую в файле client.c работу клиента: получение доступа к FIFO, подсчет времени, передача сообщения серверу, приостановка работы на некоторое время

и закрытие доступа к файлу (рис. 2.5)

```
#include "common.h"

int
main()
{
    int writefd;
    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);
        }

        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);
}
```

Figure 2.5: Реализация работы клиента

Описываю правила компиляции файлов в Makefile (рис. 2.6)


```
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
```

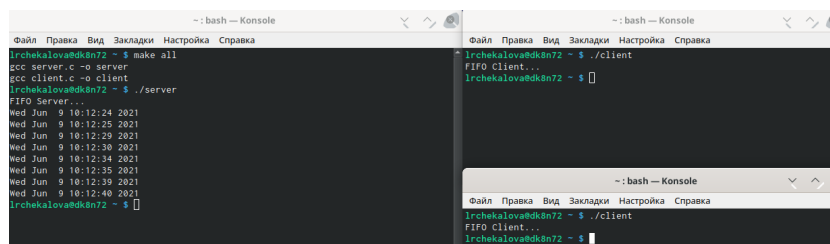
Figure 2.6: Makefile

Произвожу компиляцию файлов с помощью Makefile командой make all (рис. 2.7)

```
lrchekalova@dk8n72 ~ $ make all
gcc server.c -o server
gcc client.c -o client
lrchekalova@dk8n72 ~ $
```

Figure 2.7: Компиляция программ

Открываю три окна консоли, в первом запускаю сервер, в остальных - клиенты. Окно с сервером выводит информацию о работе клиентов и прекращает работу через 30 секунд. В случае, если канал не будет закрыт, мы не сможем создать еще один, так как он уже существует (рис. 2.8)



```
lrchekalova@dk8n72 ~ $ make all
gcc server.c -o server
gcc client.c -o client
lrchekalova@dk8n72 ~ $ ./server
FIFO Server...
Wed Jun  9 10:12:24 2021
Wed Jun  9 10:12:25 2021
Wed Jun  9 10:12:29 2021
Wed Jun  9 10:12:30 2021
Wed Jun  9 10:12:34 2021
Wed Jun  9 10:12:35 2021
Wed Jun  9 10:12:39 2021
Wed Jun  9 10:12:40 2021
lrchekalova@dk8n72 ~ $

lrchekalova@dk8n72 ~ $ ./client
FIFO Client...
lrchekalova@dk8n72 ~ $

lrchekalova@dk8n72 ~ $ ./client
FIFO Client...
lrchekalova@dk8n72 ~ $
```

Figure 2.8: Проверка работы программы

3 Выводы

После выполнения данной лабораторной работы я научилась работать с именованными каналами, организовывать работу с несколькими клиентами, реализовывать периодичность работы с помощью команды `sleep()` и определять время работы программы.

4 Библиография

1. Теоретические материалы к лабораторной работе: https://esystem.rudn.ru/pluginfile.php/1111111/mod_resource/content/1/ipc-fifo.pdf