

Отчёт по лабораторной работе №14

Паращенко Антонина Дмитриевна

Содержание

1	Цель работы	5
2	Ход лабораторной работы	6
3	Вывод	11

Список иллюстраций

2.1	Создание файлов	6
2.2	Скрипт	7
2.3	Скрипт	8
2.4	Скрипт	9
2.5	Скрипт	9
2.6	Компиляция	10
2.7	Скрипт кода	10

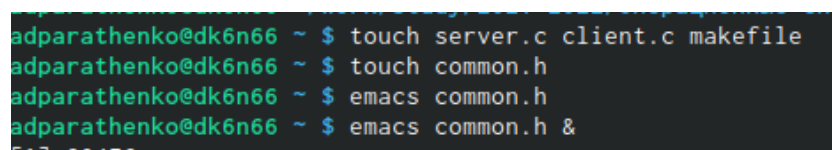
Список таблиц

1 Цель работы

Приобретение практических навыков работы с именованными каналами.

2 Ход лабораторной работы

- 1) В домашнем каталоге создаём файлы common.h server.c client.c makefile (рис. 2.1)



```
adparathenko@dk6n66 ~ $ touch server.c client.c makefile
adparathenko@dk6n66 ~ $ touch common.h
adparathenko@dk6n66 ~ $ emacs common.h
adparathenko@dk6n66 ~ $ emacs common.h &
```

Рис. 2.1: Создание файлов

- 2) Пишем скрипт файла common.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 <unistd.h>
#include <time.h>

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

#endif/* __COMMON_H__ */
```

Рис. 2.2: Скрипт

3) Пишем скрипт файла server.c (рис. 2.3)

```

#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);
            }
        }
    }
    close(readfd);

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

    exit(0);
}

```

Рис. 2.3: Скрипт

4) Пишем скрипт файла client.c (рис. 2.4)


```

#include "common.h"

#define MESSAGE "Hello Server!!!\n"

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);
            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. 2.4: Скрипт

5) Пишем скрипт файла makefile (рис. 2.5)

```

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

```

Рис. 2.5: Скрипт

6) Скомпилируем файлы с помощью команды make all (рис. 2.6)

```
adparathenko@dk6n66 ~ $ make all
gcc client.c -o client
adparathenko@dk6n66 ~ $
```

Рис. 2.6: Компиляция

7) Проверяем работу написанной программы. Открываем 3 консоли и в запускем ./server.c, а в других двух ./client.c и видим результат работы. (рис. 2.7)

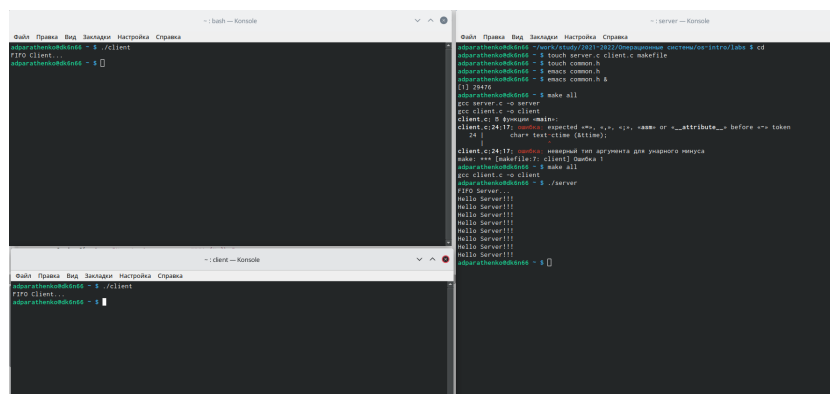


Рис. 2.7: Скрипт кода

8) Если сервер завершит свою работу, не закрыв канал, то, когда мы будем запускать этот сервер снова, появится ошибка “Невозможно создать FIFO”, так как у нас уже есть один канал.

3 Вывод

Приобрела практические навыки работы с именованными каналами.