

Отчет по лабораторной работе №14

Алмазова Елизавета Андреевна

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №14

Цель работы и задание

Цель данной лабораторной работы - приобретение практических навыков работы с именованными каналами.

Задание:

Изучите приведённые в тексте программы `server.c` и `client.c`. Взяв примеры за образец, напишите аналогичные программы, внося следующие изменения:

1. Работает не 1 клиент, а несколько (например, два).

Ход работы

1. Изучила приведённые в тексте программы `server.c` и `client.c`. Создала файлы `common.h`, `server.c`, `client.c`, `Makefile` и, взяв примеры за образец, написала аналогичные программы, внося следующие изменения (рис.1, рис.2, рис.3, рис.4):
 1. Работает не 1 клиент, а несколько (например, два).
 2. Клиенты передают текущее время раз в пять секунд.
Для получения текущего времени в файле `client.c` я использовала функции библиотеки `time.h`, добавленной в файл `common.h`, а для приостановки работы клиента функцию `sleep(5)`. Это время передается три раза с помощью цикла.

```

1 /*
2  * client.c - реализация клиента
3  *
4  * чтобы запустить пример, необходимо:
5  * 1. запустить программу server на одной консоли;
6  * 2. запустить программу client на другой консоли.
7  */
8
9 #include "common.h"
10
11 #define MESSAGE "Hello Server!!!\n"
12
13 int main(){
14     int writefd; /* дескриптор для записи в FIFO */
15     int msglen;
16
17     /* баннер */
18     printf("FIFO Client...\n");
19
20     /* цикл передачи сообщений */
21
22     for (int i=0; i<3; i++) {
23         /* получим доступ к FIFO */
24         if((writefd = open(FIFO_NAME, O_WRONLY)) <0){
25             fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n", __FILE__, strerror(errno));
26             exit(-1);
27         }
28
29         /* получим текущее время */
30         long int timer=time(0);
31         char* timeprint=ctime(&timer);
32
33         /* передадим сообщение серверу */
34         msglen = strlen(timeprint);
35         if(write(writefd, timeprint, msglen) != msglen) {
36             fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n");
37             exit(-2);
38         }
39
40         /* приостановка работы */
41
42         sleep(5);
43     }
44
45     /* закроем доступ к FIFO */
46     close(writefd);
47
48     exit(0);
49 }

```

Рисунок 1 - client.c.

```
/*
 * 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 <time.h>

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

#endif /* __COMMON_H__ */
```

Рисунок 2 - common.h.

```

1 /*
2  * server.c - реализация сервера
3  *
4  * чтобы запустить пример, необходимо:
5  * 1. запустить программу server на одной консоли;
6  * 2. запустить программу client на другой консоли.
7  */
8
9 #include "common.h"
10
11 int main() {
12     int readfd; /* дескриптор для чтения из FIFO */
13     int n;
14     char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */
15
16     /* баннер */
17     printf("FIFO Server...\n");
18
19     /* создаем файл FIFO с открытыми для всех
20      * правами доступа на чтение и запись
21      */
22
23     if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0) {
24         fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n", __FILE__, strerror(errno));
25         exit(-1);
26     }
27
28     /* откроем FIFO на чтение */
29     if((readfd = open(FIFO_NAME, O_RDONLY)) < 0) {
30         fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n", __FILE__, strerror(errno));
31         exit(-2);
32     }
33
34     /* отсчет времени работы */
35
36     clock_t count = clock();
37
38     while(clock() - count < 36) {
39         /* читаем данные из FIFO и выводим на экран */
40         while((n = read(readfd, buff, MAX_BUFF)) > 0) {
41             if(write(1, buff, n) != n) {
42                 printf(stderr, "%s: Ошибка вывода (%s)\n", __FILE__, strerror(errno));
43                 exit(-3);
44             }
45         }
46     }
47
48     close(readfd); /* закроем FIFO */
49
50     /* Удалим FIFO из системы */
51     if(unlink(FIFO_NAME) < 0) {
52         fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n", __FILE__, strerror(errno));
53         exit(-4);
54     }
55     exit(0);
56 }

```

Рисунок 3 - server.c.

```
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
```

Рисунок 4 - Makefile.

2. Программы работают верно (рис.5, рис.6).







 client	25.5 kB	14:44
 client.c	1.3 kB	14:41
 common.h	395 bytes	14:42
 Makefile	152 bytes	14:42
 server	25.6 kB	14:44
 server.c	1.7 kB	14:44

Рисунок 5 - Созданные с помощью make all файлы.

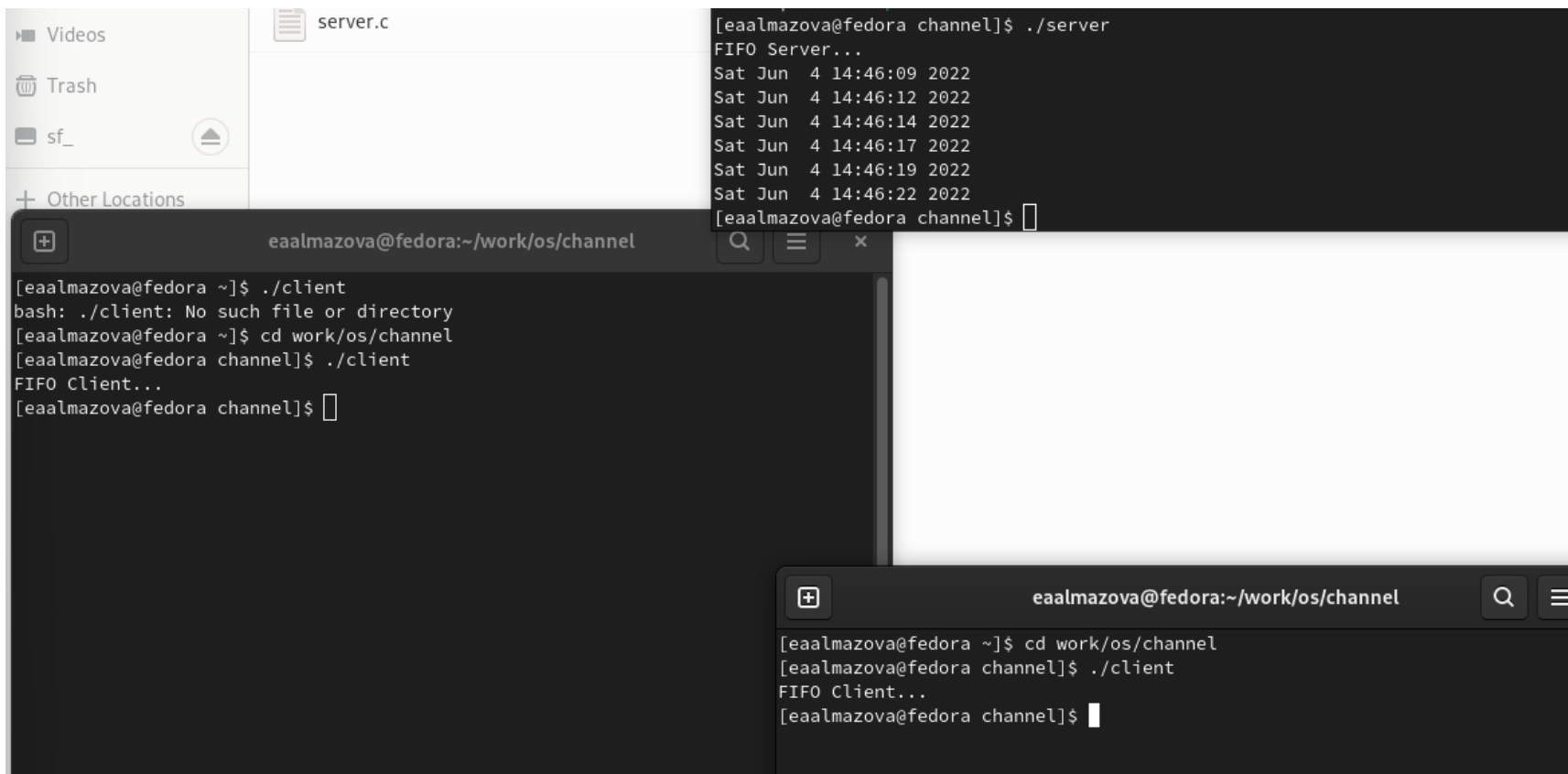


Рисунок 6 - Одновременная работа двух клиентов.

Выводы

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

Спасибо за внимание!