

Лабораторная работа №14

Операционные системы

Серёгина Ирина Андреевна

13 мая 2023

Российский университет дружбы народов, Москва, Россия

Цель работы

Приобретение практических навыков работы с именованными каналами.

Задание

Изучите приведённые в тексте программы `server.c` и `client.c`. Взяв данные примеры за образец, напишите аналогичные программы, внося следующие изменения: 1. Работает не 1 клиент, а несколько (например, два). 2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Используйте функцию `sleep()` для приостановки работы клиента. 3. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию `clock()` для определения времени работы сервера. Что будет в случае, если сервер завершит работу, не закрыв канал?

Выполнение лабораторной работы

1. Я открываю терминал и создаю файлы, которые понадобятся мне при выполнении лабораторной работы (рис. 1).

```
[irina@fedora lab14]$ touch client.c clientsec.c server.c  
[irina@fedora lab14]$ ls  
client.c clientsec.c presentation report server.c  
[irina@fedora lab14]$
```

Figure 1: создание файлов

2. Программа 1 (рис.2).

```
* 2. запустить программу client на другой консоли.
*/

#include "common.h"

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

int
main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;

    /* баннер */
    printf("FIFO Client...\n");

    for (int i; i<4; i++){

        /* получим доступ к FIFO */
        if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)

        {
            fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-1);
        }

        /* передадим сообщение серверу */
        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);
    }
}
```


3. Программа 2 (рис. 3).

```
client.c

#include "common.h"
#include <time.h>
#define MESSAGE "Hello Server!!!\n"

int
main()
{
    int writefd; /* ДЕСКРИПТОР ДЛЯ ЗАПИСИ В FIFO */
    int msglen;
    long int ttime;

    for (int i=0; i<15; i++){
        ttime=time(NULL);
        printf(ctime(&ttime));

        /* БАННЕР */
        printf("FIFO Client...\n");

        /* ПОЛУЧИМ ДОСТУП К FIFO */
        if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)

        {
            fprintf(stderr, "%s: НЕВОЗМОЖНО ОТКРЫТЬ FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-1);
        }
        /* ПЕРЕДАДИМ СООБЩЕНИЕ СЕРВЕРУ */
        msglen = strlen(MESSAGE);
        if(write(writefd, MESSAGE, msglen) != msglen)
        {
            fprintf(stderr, "%s: ОШИБКА ЗАПИСИ В FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-2);
        }
        sleep(4)
    }
}
```

4. Программа 3 (рис. 4).

```
client.c

/* БАННЕР */
printf("FIFO Server...\n");

/* СОЗДАЕМ ФАЙЛ FIFO С ОТКРЫТЫМИ ДЛЯ ВСЕХ
 * ПРАВАМИ ДОСТУПА НА ЧТЕНИЕ И ЗАПИСЬ
 */
if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
{
    fprintf(stderr, "%s: НЕВОЗМОЖНО СОЗДАТЬ FIFO (%s)\n",
        __FILE__, strerror(errno));
    exit(-1);
}

/* ОТКРОЕМ FIFO НА ЧТЕНИЕ */
if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
{
    fprintf(stderr, "%s: НЕВОЗМОЖНО ОТКРЫТЬ FIFO (%s)\n",
        __FILE__, strerror(errno));
    exit(-2);
}

clock_t beginning=time(NULL), clock_t now=time(NULL);
while (beginning-now<30){

    /* ЧИТАЕМ ДАННЫЕ ИЗ FIFO И ВЫВОДИМ НА ЭКРАН */
    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); /* ЗАКРОЕМ FIFO */
}
```

5. После этого я создаю Makefile и заполняю его, все компилируется удачно (рис. 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
```

Выводы

Я приобрела практические навыки работы с именованными каналами.