

Отчет по лабораторной работе №14

дисциплина Операционные системы

Колобова Елизавета, гр. НММбд-01-22

22 марта 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Колобова Елизавета
- студент
- Российский университет дружбы народов

Вводная часть

- программирование в оболочке ОС UNIX/Linux.

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

- Процессор **pandoc** для входного формата Markdown
- Автоматизация процесса создания: **Makefile**

Создание презентации

Изучив приведённые в тексте лабораторной работы программы `server.c` и `client.c`, напомним аналогичные программы, внося следующие изменения:

1. Работает не 1 клиент, а несколько (например, два). Для этого (как объясняли на семинарских занятиях) достаточно просто запустить клиентскую программу в двух разных консолях
2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Используйте функцию `sleep()` для приостановки работы клиента. Для этого в переменную, содержащую сообщение передаем переведенное в строку время, после чего приостанавливаем работу клиента на 5 секунд с помощью `sleep`, а в заголовочный файл подключаем `time.h` (
3. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию `clock()` для определения времени работы сервера. Указанной в задании функцией получаем время работы сервера, и с помощью цикла

```
Открыть ▼  client.c
~lab13

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
14 int
15 main()
16 {
17     int writefd; /* дескриптор для записи в FIFO */
18     int msglen;
19
20     /* баннер */
21     printf("FIFO client...\n");
22
23     /* получим доступ к FIFO */
24     if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
25     {
26         fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
27             __FILE__, strerror(errno));
28         exit(-1);
29     }
30
31     /* передадим сообщение серверу */
32     for(int i = 1; i >= 0; i++)
33     {
34         char MESSAGE[40];
35         sprintf(MESSAGE, "%li", time(0), "\n");
36         msglen = strlen(MESSAGE);
37         sleep(5);
38         if(write(writefd, MESSAGE, msglen) != msglen)
39         {
40             fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
41                 __FILE__, strerror(errno));
42             exit(-2);
43         }
44     }
```

Исследование с помощью утилиты splint

```
Открыть ▾  server.c  
~\lab13  
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  
12 main()  
13 {  
14     int readfd; /* дескриптор для чтения из FIFO */  
15     int n;  
16     char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */  
17  
18     /* баннер */  
19     printf("FIFO Server...\n");  
20     clock_t time_req;  
21     time_req = clock();  
22     if(time_req == 30)  
23     {  
24         exit(0);  
25     }  
26     else  
27     {  
28         /* создаем файл FIFO с открытыми для всех  
29         * правами доступа на чтение и запись  
30         */  
31         if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)  
32         {  
33             fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",  
34                     __FILE__, strerror(errno));  
35             exit(-1);  
36         }  
37  
38         /* откроем FIFO на чтение */  
39         if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)  
40         {  
41             fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",  
42                     __FILE__, strerror(errno));  
43             exit(-1);  
44         }  
45     }  
46 }
```

Результаты

Результатом проделанной работы является приобретение практических навыков работы с именованными каналами.