

Структура научной презентации

Простейший шаблон

Кулябов Д. С.

01 января 1970

Российский университет дружбы народов, Москва, Россия

Объединённый институт ядерных исследований, Дубна, Россия

Информация

- Мурзаев Замир Зейнадинович
- д.ф.-м.н., профессор
- профессор кафедры прикладной информатики и теории вероятностей
- Российский университет дружбы народов
- kulyabov-ds@rudn.ru
- <https://yamadharma.github.io/ru/>



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, внеся 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

Выполнение лабораторной работы

##Создаем файлы необходимые

```
zzmurzaev@dk2n25 ~/work/study/2022-2023/Операционные системы/study_2022-2023_os-intro/labs/lab14 $ touch  
common.h server.c client.c Makefile
```

Рис. 1: Команда touch

Реализация сервера, создаем

```
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
21     /* создаем файл FIFO с открытыми для всех
22      * правами доступа на чтение и запись
23      */
24     if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
25     {
26         fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
27             __FILE__, strerror(errno));
28         exit(-1);
29     }
30
31     /* откроем FIFO на чтение */
32     if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
33     {
34         fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
35             __FILE__, strerror(errno));
36         exit(-2);
37     }
38     /* начало отсчёта времени */
39     clock_t start = time(NULL);
40
41     /* цикл работает пока с момента начала отсчёта времени прошло меньше 30 секунд */
```

Изменяем еще один файл

```
9 #include "common.h"
10
11 #define MESSAGE "Hello Server!!!\n"
12
13 int
14 main()
15 {
16     int writefd; /* дескриптор для записи в FIFO */
17     int msglen;
18
19     /* баннер */
20     printf("FIFO Client...\n");
21
22     /* цикл отвечающий за отправку сообщения о текущем времени */
23     for(int i=0; i<4; i++)
24     {
25         /* получим доступ к FIFO */
26         if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
27         {
28             fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
29                     __FILE__, strerror(errno));
30             exit(-1);
31         }
32     }
33
34     /* текущее время */
35     long int ttime=time(NULL);
36     char* text=ctime(&ttime);
37
38     /* передадим сообщение серверу */
39     msglen = strlen(MESSAGE);
40     if(write(writefd, MESSAGE, msglen) != msglen)
41     {
42         fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
43                 __FILE__, strerror(errno));
44         exit(-2);
45     }
46     /* приостановка работы клиента на 5 секунд */
47     sleep(5);
48 }
```

Создаем файл, с помощью которого будем создавать исполняемые файлы

```
1 all: server client  
2  
3 server: server.c common.h  
4         gcc server.c -o server  
5  
6 client: client.c common.h  
7         gcc client.c -o client  
8  
9 clean:  
10        -rm server client *.o
```

Рис. 4: файл makefile

Создаем заголовочный файл

```
1 /*
2 * common.h - заголовочный файл со стандартными определениями
3 */
4 #ifndef __COMMON_H__
5 #define __COMMON_H__
6
7 #include <stdio.h>
8 #include <stdlib.h>
9 #include <string.h>
10 #include <errno.h>
11 #include <sys/types.h>
12 #include <sys/stat.h>
13 #include <fcntl.h>
14 #include <unistd.h>
15 #include <time.h>
16
17 #define FIFO_NAME "/tmp/fifo"
18 #define MAX_BUFF 80
19
20#endif /* __COMMON_H__ */
```

Выводы

Приобретены практические навыки работы с именованными каналами.