

Отчет по лабораторной работе №15

Именованные каналы.

Кузнецов Юрий Владимирович

Содержание

Цель.....	1
Задание	1
Выполнение лабораторной работы.....	1
Вывод	5

Цель

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.

Выполнение лабораторной работы

Написал файлы, которые нужны для работы будущей программы. Файл `server1.c`: (рис. 1) Файл `client.c`: (рис. 2) Файл `client2.c`: (рис. 3) Файл `common.h`: (рис. 4) Файл `Makefile`: (рис. 5)

```

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 int
11 main()
12 {
13     int readfd; /* дескриптор для чтения из FIFO */
14     int n;
15     char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */
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     if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
23     {
24         fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
25             __FILE__, strerror(errno));
26         exit(-1);
27     }
28
29     /* откроем FIFO на чтение */
30     if((readfd=open(FIFO_NAME, O_RDONLY)) < 0)
31     {
32         fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
33             __FILE__, strerror(errno));
34         exit(-2);
35     }
36     /* читаем данные из FIFO и выводим на экран */
37     clock_t now = time(NULL), start = time(NULL);
38     while((now-start)<30){
39         while((n = read(readfd, buff, MAX_BUFF)) > 1)
40         {
41             if(write(2, buff, n) != n)
42             {
43                 fprintf(stderr, "%s: Ошибка вывода (%s)\n",
44                     __FILE__, strerror(errno));
45                 exit(-3);

```

Figure 1: server1.c

```

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 #include "common.h"
9 #define MESSAGE "Hello Server!!! from 1\n"
10 int
11 main()
12 {
13     int writefd; /* дескриптор для записи в FIFO */
14     int msglen;
15     long long int T;
16     for(int i=0; i<5;i++){
17         sleep(5);
18         T=time(NULL);
19         /* баннер */
20         printf("FIFO Client...\n");
21     }
22     /* получим доступ к FIFO */
23     if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
24     {
25         fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
26             __FILE__, strerror(errno));
27         exit(-1);
28     }
29     /* передадим сообщение серверу */
30     msglen = strlen(MESSAGE);
31     if(write(writefd, MESSAGE, msglen) != msglen)
32     {
33         fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
34             __FILE__, strerror(errno));
35         exit(-2);
36     }
37
38     /* закроем доступ к FIFO */
39     close(writefd);
40 }
41 exit(0);
42 }

```

Figure 2: client.c

```

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 #include "common.h"
9 #define MESSAGE "Hello Server!!! from 2\n"
10 int
11 main()
12 {
13     int writefd; /* дескриптор для записи в FIFO */
14     int msglen;
15     long long int T;
16     for(int i=0; i<3; i++){
17         sleep(5);
18         T=time(NULL);
19         /* баннер */
20         printf("FIFO Client...\n");
21
22         /* получим доступ к FIFO */
23         if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
24         {
25             fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
26                 __FILE__, strerror(errno));
27             exit(-1);
28         }
29         /* передадим сообщение серверу */
30         msglen = strlen(MESSAGE);
31         if(write(writefd, MESSAGE, msglen) != msglen)
32         {
33             fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
34                 __FILE__, strerror(errno));
35             exit(-2);
36         }
37
38         /* закроем доступ к FIFO */
39         close(writefd);
40     }
41     exit(0);
42 }

```

Figure 3: client2.c

```

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

```

Figure 4: common.h

```

1 all: server1 client client2
2
3 server: server1.c common.h
4     gcc server1.c -o server1
5
6 client: client.c common.h
7     gcc client.c -o client
8     gcc client2.c -o client2
9
10 clean:
11     -rm server1 client *.o

```

Figure 5: Makefile

Потом скомпилировал файлы используя Makefile и команду make (рис. 6):

```

cc server1.c -o server1
server1.c: 8 функции «main»:
server1.c:37:17: предупреждение: неявная декларация функции «time» [-Wimplicit-function-declaration]
37 | clock_t now = time(NULL), start = time(NULL);
    |               ^~~~~~
server1.c:39:15: предупреждение: неявная декларация функции «read»; имелось в виду «fread»? [-Wimplicit-function-declaration]
39 | while((n = read(readfd, buff, MAX_BUFF)) > 1)
    |               ^~~~~~
server1.c:41:11: предупреждение: неявная декларация функции «write»; имелось в виду «fwrite»? [-Wimplicit-function-declaration]
41 | if(write(2, buff, n) != n)
    |     ^~~~~~
server1.c:51:5: предупреждение: неявная декларация функции «close»; имелось в виду «pclose»? [-Wimplicit-function-declaration]
51 | close(readfd); /* закроем FIFO */
    |     ^~~~~~
server1.c:56:6: предупреждение: неявная декларация функции «unlink» [-Wimplicit-function-declaration]
56 | if(unlink(FIFO_NAME) < 0)
    |     ^~~~~~
gcc client.c -o client
client.c: 8 функции «main»:
client.c:17:3: предупреждение: неявная декларация функции «sleep» [-Wimplicit-function-declaration]
17 | sleep(5);
    |     ^~~~~
client.c:18:3: предупреждение: неявная декларация функции «time» [-Wimplicit-function-declaration]
18 | T=time(NULL);
    |     ^~~~~~
client.c:31:6: предупреждение: неявная декларация функции «write»; имелось в виду «fwrite»? [-Wimplicit-function-declaration]
31 | if(write(writefd, MESSAGE, msglen) != msglen)
    |     ^~~~~~
client.c:39:3: предупреждение: неявная декларация функции «close»; имелось в виду «pclose»? [-Wimplicit-function-declaration]
39 | close(writefd);
    |     ^~~~~~
gcc client2.c -o client2
client2.c: 8 функции «main»:
client2.c:17:3: предупреждение: неявная декларация функции «sleep» [-Wimplicit-function-declaration]
17 | sleep(5);
    |     ^~~~~
client2.c:18:5: предупреждение: неявная декларация функции «time» [-Wimplicit-function-declaration]
18 | T=time(NULL);
    |     ^~~~~~
client2.c:31:6: предупреждение: неявная декларация функции «write»; имелось в виду «fwrite»? [-Wimplicit-function-declaration]
31 | if(write(writefd, MESSAGE, msglen) != msglen)
    |     ^~~~~~

```

Figure 6: make

Вышло много предупреждений, но это не мешает дальнейшей работе. Главное нет ошибок.

Здесь показано завершение работы сервера, но при этом клиент не успел провести всю работу. Вышло предупреждение: (рис. 7)

```

FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
client2.c: Невозможно открыть FIFO (No such file or directory)

FIFO Server...
Hello Server!!! from 1
Hello Server!!! from 1

```

Figure 7: Прекращение работы клиента

Если бы я прекратил работу сервера принудительно с помощью Ctrl+C, то я не мог бы в дальнейшем запустить программу еще раз. Мне пришлось бы искать файл fifo и удалять его вручную.

Итог

Ниже показана работа программы в целом (рис. 8):

```

FIFO Server...
Hello Server!!! from 1
Hello Server!!! from 1
Hello Server!!! from 2
Hello Server!!! from 1
Hello Server!!! from 2
Hello Server!!! from 1
Hello Server!!! from 2
Hello Server!!! from 1
Hello Server!!! from 2
Hello Server!!! from 1
Hello Server!!! from 2
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...

```

```

client2.c: В функции «main»:
client2.c:17:3: предупреждение: неявная декларация функции «sleep» [
unction-declaration]
   17 |     sleep(5);
       |     ~~~~~
client2.c:18:5: предупреждение: неявная декларация функции «time» [~
nction-declaration]
   18 |     T=time(NULL);
       |     ~~~~~
client2.c:31:6: предупреждение: неявная декларация функции «write»;
дуга «fwrite»? [-Wimplicit-function-declaration]
   31 |     if(write(writefd, MESSAGE, msglen) != msglen)
       |     ~~~~~
       |     fwrite
client2.c:39:3: предупреждение: неявная декларация функции «close»;
дуга «pclose»? [-Wimplicit-function-declaration]
   39 |     close(writefd);
       |     ~~~~~
       |     pclose

```

Figure 8: Итог

Вывод

Я успешно приобрел практические навыки работы с именованными каналами.