

Презентация по лабораторной работе №14

Именованные каналы

Лушин А.А.

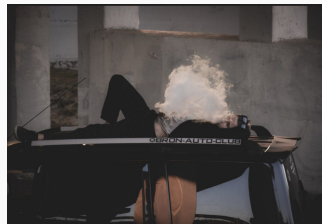
18 февраля 2005

Российский университет дружбы народов, Москва, Россия

Факультет Физико-математических и естественных наук

Информация

- Лушин Артем Андреевич
- бакалавр направления компьютерная и информационные науки
- ученик кафедры информационных технологий
- Российский университет дружбы народов
- lusin5745@gmail.com



Вводная часть

- Приобретение практических навыков работы с именованными каналами.

Ход работы

Я создал каталог 14 и в нем создал файлы: client.c, client2.c, common.h, server.c. Основы для программ я взял с файла для лабораторной. Там описана программа для работы 1 пользователя, но я немного изменил и добавил файл client2.c для работы сразу двух клиентов. Каждый из 4 файлов отвечает за свою часть. Файл server отвечает за создание сервера для работы. Файл common.h содержит все используемые библиотеки. Файл client2.c, client.c содержит код для работы пользователей.

```
common.h      [-----]  0 L:[ 1+ 0  1/ 16] *(0  / 263b) 0035 0x023  [*][X]  
#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>
```

Файл Makefile

Код для файла makefile есть в лабораторной, но его надо немного изменить. Данный файл создает сами программы. Когда у нас написаны текстовые файлы, makefile создает программы. Мы можем это сделать вручную, но для большей удобства использует файл.

```
[aalushing@fedora 14]$ ls
client2.c client.c common.h makefile server.c
[aalushing@fedora 14]$ make
gcc server.c -o server
server.c: В функции «main»:
server.c:23:15: предупреждение: неявная декларация функции «time» [-Wimplicit-function-declaration]
 23 |     clock_t now=time(NULL), start=time(NULL);
    |                   ^~~~~
    |                   |
server.c:2:1: замечание: «time» is defined in header «<time.h>»; did you forget to «#include <time.h>»?
 1 | #include "common.h"
++ | ++#include <time.h>
 2 | int
server.c:26:18: предупреждение: неявная декларация функции «read»; имелось в виду «fread»? [-Wimplicit-function-declaration]
 26 |     while((n = read(readfd, buff, MAX_BUFF)) > 0)
    |                   ^~~~~
    |                   fread
server.c:28:14: предупреждение: неявная декларация функции «write»; имелось в виду «fwrite»? [-Wimplicit-function-declaration]
 28 |         if(write(l, buff, n) != n)
    |             ^~~~~
    |             fwrite
server.c:37:3: предупреждение: неявная декларация функции «close»; имелось в виду «pclose»? [-Wimplicit-function-declaration]
 37 |     close(readfd);
    |     ^~~~~
    |     pclose
server.c:39:6: предупреждение: неявная декларация функции «unlink» [-Wimplicit-function-declaration]
 39 |     if(unlink(FIFO_NAME) < 0)
    |         ^~~~~
gcc client.c -o client
client.c: В функции «main»:
client.c:13:5: предупреждение: неявная декларация функции «sleep» [-Wimplicit-function-declaration]
 13 |     sleep(3);
    |     ^~~~~
client.c:14:7: предупреждение: неявная декларация функции «time» [-Wimplicit-function-declaration]
 14 |     t=time(NULL);
    |     ^~~~~
client.c:2:1: замечание: «time» is defined in header «<time.h>»; did you forget to «#include <time.h>»?
 1 | #include "common.h"
```


Запуск программ

Программы я запускаю с двух консолей. Так как программа рассчитана для двух пользователей, то я имитирую второго пользователя второй консолью. Если мы запустим только одну из программ, ничего не будет происходить, так как файл просто будет ждать информации от второго пользователя.

```
client client2.c client.c common.h makefile server server.c
[aalushin@fedora 14]$ ./server
FIFO Server...
Hello Server!!!
Hello Server!!!
█
```

```
[aalushin@fedora 14]$ ./client
FIFO Client...
FIFO Client...
```

Результаты

Контрольные вопросы

1. В чем ключевое отличие именованных каналов от неименованных?
2. Возможно ли создание неименованного канала из командной строки?
3. Возможно ли создание именованного канала из командной строки?
4. Опишите функцию языка C, создающую неименованный канал.
5. Опишите функцию языка C, создающую именованный канал.
6. Что будет в случае прочтения из `fifo` меньшего числа байтов, чем находится в канале?
Большого числа байтов?
7. Аналогично, что будет в случае записи в `fifo` меньшего числа байтов, чем позволяет буфер? Большого числа байтов?
8. Могут ли два и более процессов читать или записывать в канал?

Я приобрел практические навыки работы с именованными файлами.

Ты сильнее, чем то, что заставляет тебя тревожиться.