

Презентация по лабораторной работе №14

Дисциплина: операционные системы

Лобанова П. И.

09.05.2023

Российский университет дружбы народов, Москва, Россия

Информация

- Лобанова Полина Иннокентьевна
- учащаяся факультета физико-математических и естественных наук
- учащаяся направления “Компьютерные и информационные науки”
- Российский университет дружбы народов
- polla-2004@mail.ru

Цель работы

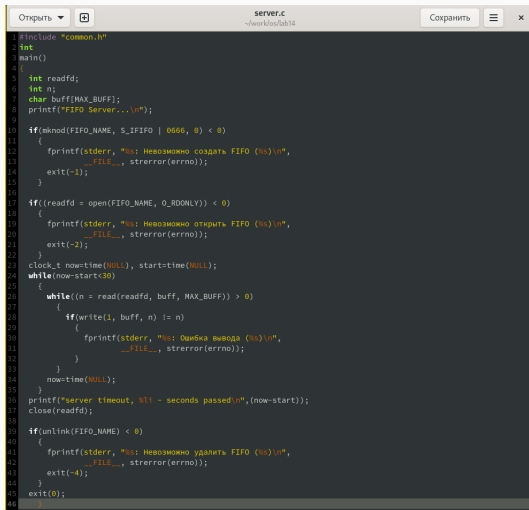
Приобретение практических навыков работы с именованными каналами.

Ход работы

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.

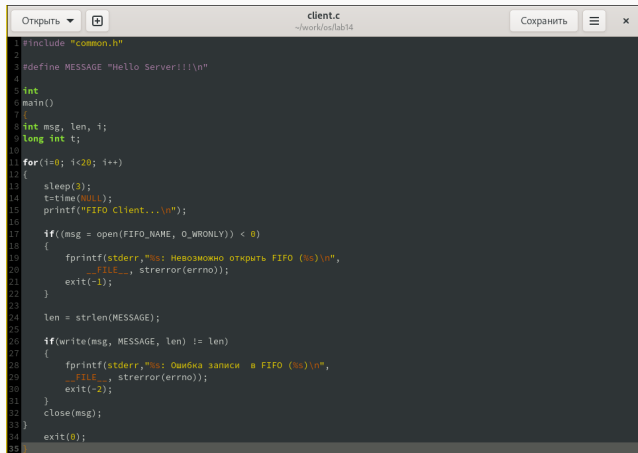
Написание файлов

Первое, что нам нужно сделать, это просмотреть исходные файлы и внести свои изменения.



```
server.c
~\work\os\lab14

1 #include "common.h"
2 int
3 main()
4 {
5     int readfd;
6     int n;
7     char buff[MAX_BUFF];
8     printf("FIFO Server...\n");
9
10    if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
11    {
12        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
13                __FILE__, strerror(errno));
14        exit(-1);
15    }
16
17    if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
18    {
19        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
20                __FILE__, strerror(errno));
21        exit(-2);
22    }
23    clock_t now=time(NULL), start=time(NULL);
24    while(now-start<30)
25    {
26        while((n = read(readfd, buff, MAX_BUFF)) > 0)
27        {
28            if(write(1, buff, n) != n)
29            {
30                fprintf(stderr, "%s: Ошибка вывода (%s)\n",
31                        __FILE__, strerror(errno));
32            }
33        }
34        now=time(NULL);
35    }
36    printf("server timeout, %li - seconds passed\n", (now-start));
37    close(readfd);
38
39    if(unlink(FIFO_NAME) < 0)
40    {
41        fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
42                __FILE__, strerror(errno));
43        exit(-4);
44    }
45    exit(0);
46 }
```

The image shows a code editor window titled "client.c" with the file path "~\work\os\lab14". The editor contains the following C code:

```
1 #include "common.h"
2
3 #define MESSAGE "Hello Server!!!\n"
4
5 int
6 main()
7 {
8     int msg, len, i;
9     long int t;
10
11     for(i=0; i<20; i++)
12     {
13         sleep(3);
14         t=time(NULL);
15         printf("FIFO Client...\n");
16
17         if((msg = open(FIFO_NAME, O_WRONLY)) < 0)
18         {
19             fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
20                     __FILE__, strerror(errno));
21             exit(-1);
22         }
23
24         len = strlen(MESSAGE);
25
26         if(write(msg, MESSAGE, len) != len)
27         {
28             fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
29                     __FILE__, strerror(errno));
30             exit(-2);
31         }
32         close(msg);
33     }
34     exit(0);
35 }
```

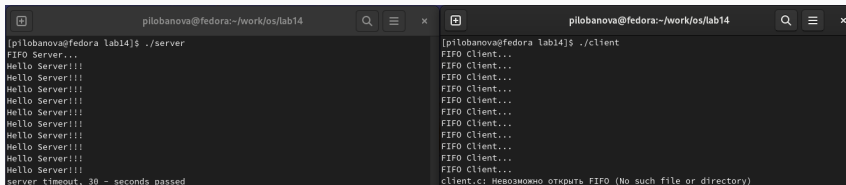
Рис. 2: Содержание файла *client.c*.

Далее необходимо скомпилировать два этих файла, это мы делаем с помощью файла Makefile.

[illegible]

Рис. 3: Компиляция.

И последнее, что мы делаем, это запуск самой программы.



```
[pilobanova@fedora lab14]$ ./server
FIFO Server...
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
server timeout, 30 - seconds passed

[pilobanova@fedora lab14]$ ./client
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
client.c: Невозможно открыть FIFO (No such file or directory)
```

Рис. 4: Результат.

Контрольные вопросы

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?
9. Опишите функцию `write` (тип возвращаемого значения, аргументы и логику работы). Что означает 1 (единица) в вызове этой функции в программе `server.c` (строка 42)?
10. Опишите функцию `strerror`.

Вывод

Я приобрела практические навыки работы с именованными каналами.