

Лабораторная работа 14

Именованные каналы

Паласиос Ф.

08 мая 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Паласиос Фелипе
- студент группы НКАбд - 04 - 22
- Российский университет дружбы народов

Вводная часть

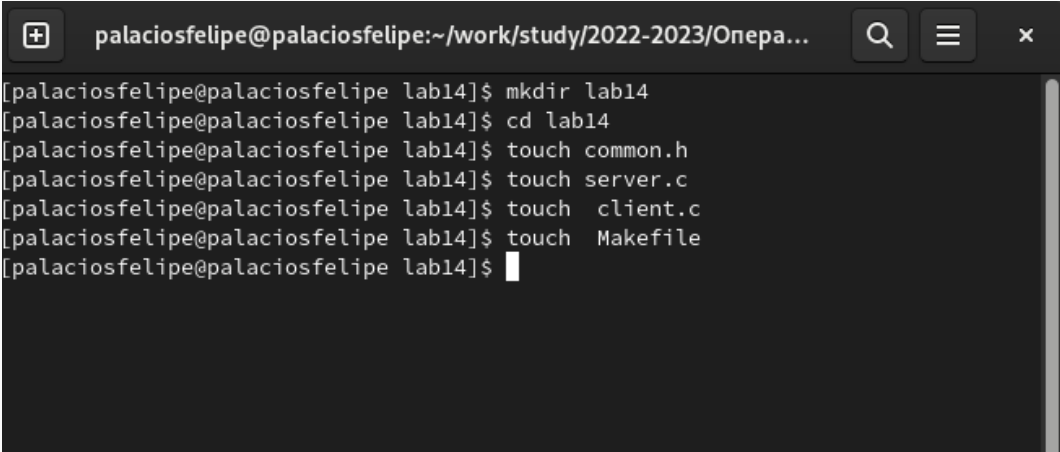
Приобретение практических навыков работы с именованными каналами.

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. Что будет в случае, если сервер завершит работу, не закрыв канал?

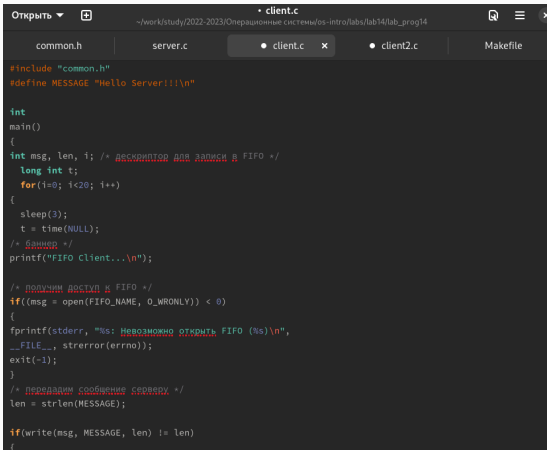
Выполнение лабораторной работы


Изучите приведённые в тексте программы `server.c` и `client.c`. Взяв данные примеры за образец, напишите аналогичные программы, внося следующие изменения:



```
palaciosfelipe@palaciosfelipe:~/work/study/2022-2023/Опера...  
[palaciosfelipe@palaciosfelipe lab14]$ mkdir lab14  
[palaciosfelipe@palaciosfelipe lab14]$ cd lab14  
[palaciosfelipe@palaciosfelipe lab14]$ touch common.h  
[palaciosfelipe@palaciosfelipe lab14]$ touch server.c  
[palaciosfelipe@palaciosfelipe lab14]$ touch client.c  
[palaciosfelipe@palaciosfelipe lab14]$ touch Makefile  
[palaciosfelipe@palaciosfelipe lab14]$
```


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.




```
Открыть ▾  • client.c
~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14/lab_prog14

common.h | server.c | • client.c x | • client2.c | Makefile

#include "common.h"
#define MESSAGE "Hello Server!!!\n"

int
main()
{
    int msg, len, i; /* дескриптор для записи в FIFO */
    long int t;
    for(i=0; i<20; i++)
    {
        sleep(3);
        t = time(NULL);
        /* баннер */
        printf("FIFO Client...\n");

        /* получим доступ к FIFO */
        if((msg = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-1);
        }
        /* передадим сообщение серверу */
        len = strlen(MESSAGE);
        if(write(msg, MESSAGE, len) != len)
        {
```

```
Открыть ▾  • client2.c ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14/lab_prog14
common.h server.c client.c • client2.c x Makefile

#include "common.h"
#define MESSAGE "Hello Server!!!\n"

int
main()
{
    int msg, len, i; /* описание для записи в FIFO */
    long long int t;
    char message[10];

    for(count=0; count<5; ++count)
    {
        sleep(5);
        t=(long long int) time(0);
        sprintf(message, "%lli", t);
        /* бегун */

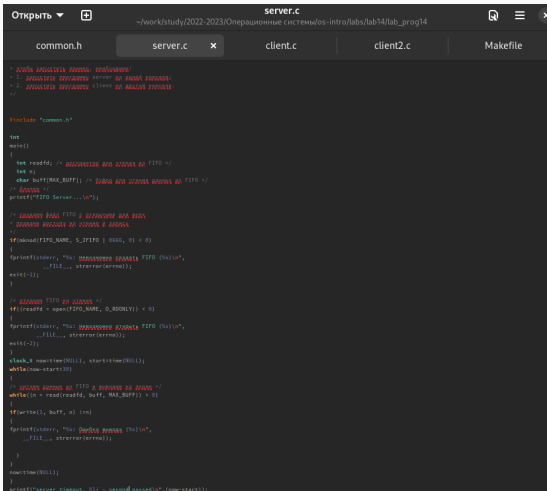
        /* получим доступ к FIFO */
        if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-1);
        }

        /* передадим сообщение серверу */
        msglen = strlen(MESSAGE);
        if(write(writefd, MESSAGE, msglen) != msglen)
        {
            fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-2);
        }

        /* закрыв доступ к FIFO */
        close(writefd);

        exit(0);
    }
}
```

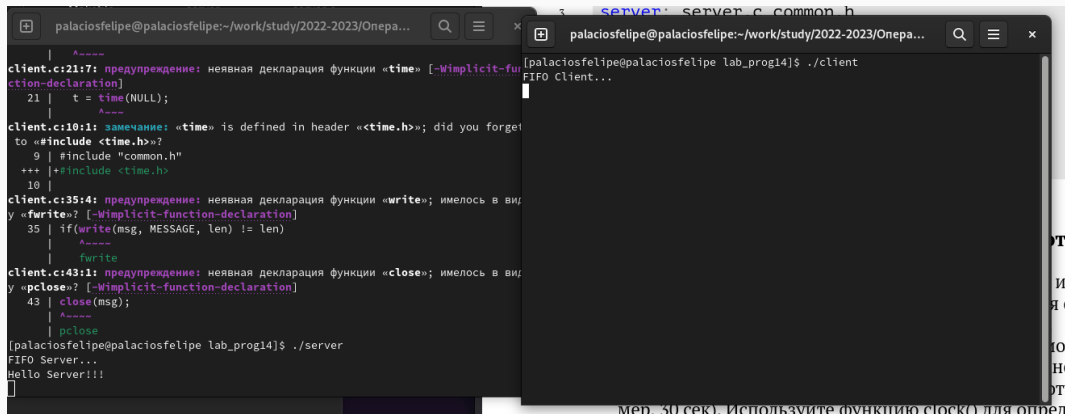
3. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию `clock()` для определения времени работы сервера.



```
server.c
~/workstudy/2022-2023/Операционные системы/os-intro/labs/lab14/lab_prog14

common.h  server.c x  client.c  client2.c  Makefile

/* A simple server program: server.c */
1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <unistd.h>
4. #include <sys/types.h>
5. #include <sys/stat.h>
6. #include <fcntl.h>
7. #include <time.h>
8. #include "common.h"
9.
10. int
11. main()
12. {
13.     int readfd; /* READ fd to READ from FIFO */
14.     int n;
15.     char buff[MAX_BUFF]; /* Buffer to hold data read from FIFO */
16.     /* OPEN */
17.     printf("FIFO Server...\n");
18.
19.     /* OPEN FIFO & SETMODE FOR READ
20.      * OPENMODE FOR READ & WRITE
21.      */
22.     if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
23.     {
24.         printf(stderr, "No: %s\n", strerror(errno));
25.         _FILE_, strerror(errno);
26.         exit(-1);
27.     }
28.
29.     /* OPEN FIFO TO READ */
30.     if(readfd = open(FIFO_NAME, O_RDONLY) < 0)
31.     {
32.         printf(stderr, "No: %s\n", strerror(errno));
33.         _FILE_, strerror(errno);
34.         exit(-2);
35.     }
36.     clock_t now_time(NULL), start_time(NULL);
37.     while(now_time - start_time < 30)
38.     {
39.         /* READ FROM FIFO & WRITE TO FILE */
40.         while(n = read(readfd, buff, MAX_BUFF) > 0)
41.         {
42.             if(write(1, buff, n) != n)
43.             {
44.                 printf(stderr, "No: %s\n", strerror(errno));
45.                 _FILE_, strerror(errno);
46.             }
47.         }
48.         now_time(NULL);
49.     }
50.     printf("Server timeout. 30 - second passed\n", now_time);
51. }
```



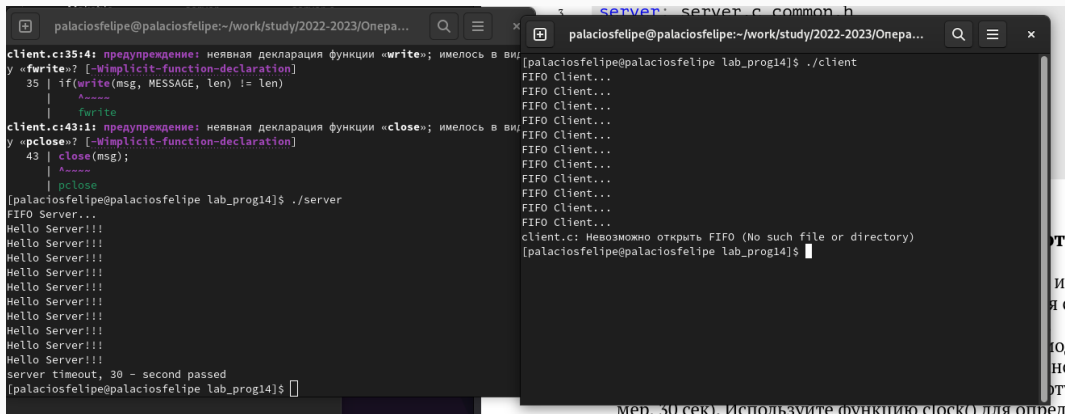
```
palaciosfelipe@palaciosfelipe:~/work/study/2022-2023/Onepa...
client.c:21:7: предупреждение: неявная декларация функции «time» [-Wimplicit-fu
ction-declaration]
 21 |     t = time(NULL);
    |           ^
client.c:10:1: замечание: «time» is defined in header «<time.h>»; did you forget
to «#include <time.h>»?
   9 | #include "common.h"
+++ |+#include <time.h>
  10 |
client.c:35:4: предупреждение: неявная декларация функции «write»; имелось в вид
у «fwrite»? [-Wimplicit-function-declaration]
   35 | if(write(msg, MESSAGE, len) != len)
    |    ^~~~~
client.c:43:1: предупреждение: неявная декларация функции «close»; имелось в вид
у «pclose»? [-Wimplicit-function-declaration]
   43 | close(msg);
    |     ^~~~~
[palaciosfelipe@palaciosfelipe lab_prog14]$ ./server
FIFO Server...
Hello Server!!!

[palaciosfelipe@palaciosfelipe lab_prog14]$ ./client
FIFO Client...
```

меб. 30 сек). Используйте функцию clock() для определ

Рис. 5: Запуск

Результаты



```
client.c:35:4: предупреждение: неявная декларация функции «write»; имелось в виду
у «fwrite»? [-Wimplicit-function-declaration]
  35 | if(write(msg, MESSAGE, len) != len)
      |      ^~~~~
      |      fwrite
client.c:43:1: предупреждение: неявная декларация функции «close»; имелось в виду
у «pclose»? [-Wimplicit-function-declaration]
  43 | close(msg);
      |      ^~~~~
      |      pclose
[palaciosfelipe@palaciosfelipe lab_prog14]$ ./server
FIFO Server...
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
server timeout, 30 - second passed
[palaciosfelipe@palaciosfelipe lab_prog14]$
```

```
server.c: common.h
[palaciosfelipe@palaciosfelipe lab_prog14]$ ./client
FIFO Client...
FIFO Client...
FIFO 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)
[palaciosfelipe@palaciosfelipe lab_prog14]$
```

мед. 30 сек). Используйте функцию clock() для определ

Рис. 6: Результат

Приобрел практические навыки работы с именованными каналами