

Лабораторная работа No 14

Именованные каналы

Паласиос Фелипе

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	12

Список иллюстраций

3.1	Files	7
3.2	Client	8
3.3	Client2	9
3.4	Server	10
3.5	Запуск	10
3.6	Результат	11

Список таблиц

1 Цель работы

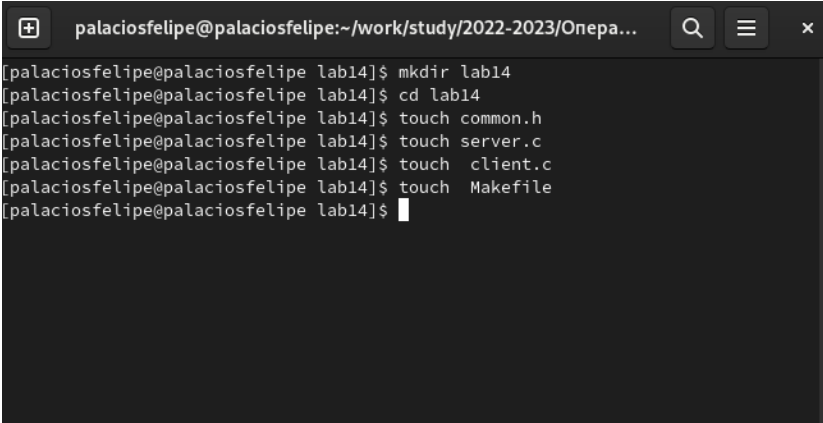
Приобретение практических навыков работы с именованными каналами.

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

3 Выполнение лабораторной работы

Изучите приведённые в тексте программы `server.c` и `client.c`. Взяв данные примеры за образец, напишите аналогичные программы, внося следующие изменения:(рис. 3.1)



```
palaciosfelipe@palaciosfelipe:~/work/study/2022-2023/Онепа...
[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]$
```

Рис. 3.1: Files

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.(рис. 3.2, 3.3),

```
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);
        /* вывод */
        printf("FIFO Client...\n");

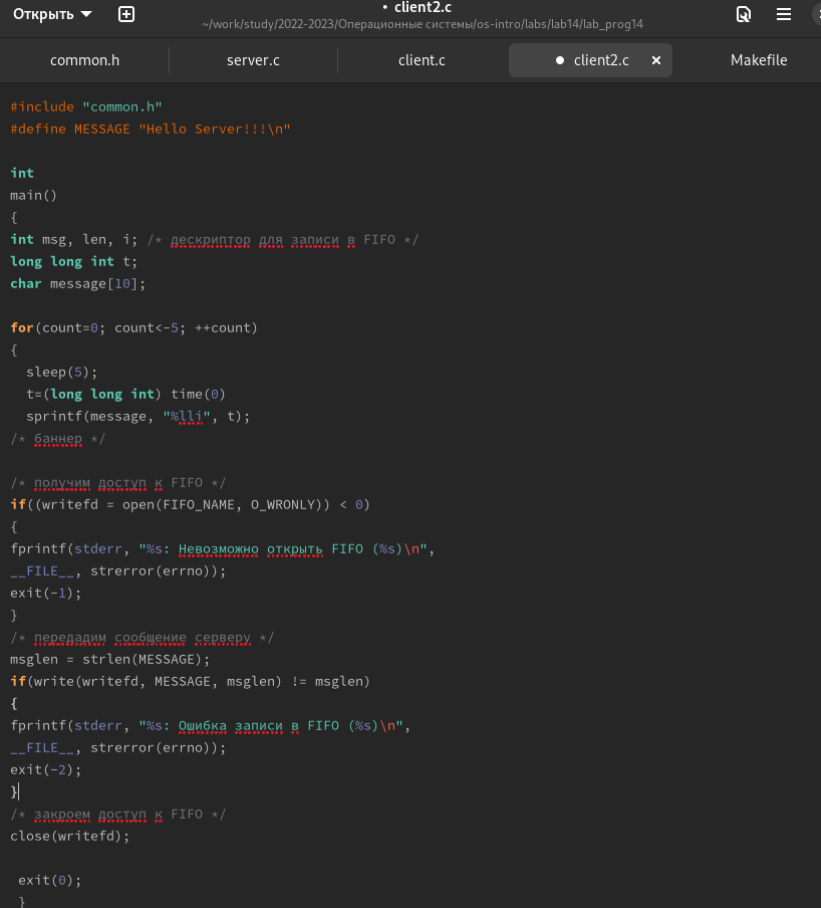
        /* получим доступ к FIFO */
        if((msg = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-1);
        }

        /* передадим сообщение серверу */
        len = strlen(MESSAGE);

        if(write(msg, MESSAGE, len) != len)
        {
            fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-2);
        }

        /* закроем доступ к FIFO */
        close(msg);
    }
    exit(0);
}
```

Рис. 3.2: Client



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

int
main()
{
    int msg, len, i; /* дескриптор 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);
        /* БАННЕР */

        /* 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);
        }
        /* закроем доступ к FIFO */
        close(writefd);

        exit(0);
    }
}
```

Рис. 3.3: Client2

3. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию clock() для определения времени работы сервера. (рис. 3.4, 3.5, 3.6)

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

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

/*
 * 1. Аппаратные ресурсы: процессор
 * 2. Аппаратные ресурсы: сервер на локальной машине
 */

#include "common.h"

int
main()
{
    int readfd; /* READ-только для FIFO */
    int n;
    char buff[MAX_BUFF]; /* Buffer size MAX_BUFF for FIFO */
    /*
     * 1. Создаем FIFO
     */
    if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
    {
        fprintf(stderr, "mknod: %s\n",
            __FILE__, strerror(errno));
        exit(-1);
    }

    /*
     * 2. Открываем FIFO для чтения
     */
    if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
    {
        fprintf(stderr, "open: %s\n",
            __FILE__, strerror(errno));
        exit(-2);
    }

    clock_t nowtime(NULL), starttime(NULL);
    while(nowtime < starttime + 30)
    {
        /*
         * 3. Читаем данные из FIFO и выводим их на stdout
         */
        while((n = read(readfd, buff, MAX_BUFF)) > 0)
        {
            if(write(1, buff, n) != n)
            {
                fprintf(stderr, "write: %s\n",
                    __FILE__, strerror(errno));
            }
        }
        nowtime(NULL);
    }

    printf("server timeout, %d - second passed\n", nowtime - starttime);
    close(readfd); /* Close FIFO */

    /*
     * 4. Удаляем FIFO
     */
    if(unlink(FIFO_NAME) < 0)
    {
        fprintf(stderr, "unlink: %s\n",
            __FILE__, strerror(errno));
        exit(-4);
    }
    exit(0);
}
```

Рис. 3.4: Server

```
palaciosfelipe@palaciosfelipe:~/work/study/2022-2023/Onepa...
client.c:21:17: предупреждение: неявная декларация функции «time» [-Wimplicit-fu
tion-declaration]
21 |     t = time(NULL);
    |               ^
client.c:10:11: предупреждение: «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 ви
у «write» [-Wimplicit-function-declaration]
35 | if(write(msg, MESSAGE, len) != len)
    |    ^~~~~
client.c:43:11: предупреждение: неявная декларация функции «close»; имелось в ви
у «close» [-Wimplicit-function-declaration]
43 | close(msg);
    |     ^~~~~
[palaciosfelipe@palaciosfelipe lab_prog14]$ ./server
FIFO Server...
Hello Server!!!

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

Рис. 3.5: Запуск

```
client.c:35:11: предупреждение: неявная декларация функции «write»; имелось в виду «fwrite» [-Wimplicit-function-declaration]
35 | if(write(msg, MESSAGE, len) != len)
    | ^~~~~~
    | fwrite
client.c:43:11: предупреждение: неявная декларация функции «close»; имелось в виду «pclose» [-Wimplicit-function-declaration]
43 | close(msg);
    | ^~~~~~
    | pclose
[palaciosfelipe@palaciosfelipe lab_prog14]$ ./server
FIPO 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]$

[palaciosfelipe@palaciosfelipe lab_prog14]$ ./client
FIPO Client...
FIPO Client...
FIPO Client...
FIPO Client...
FIPO Client...
FIPO Client...
FIPO Client...
FIPO Client...
client.c: Не удалось открыть FIFO (No such file or directory)
[palaciosfelipe@palaciosfelipe lab_prog14]$
```

Рис. 3.6: Результат

4 Выводы

Приобрел практические навыки работы с именованными каналами