

Отчёт по лабораторной работе №14

Именованные каналы

Федорина Эрнест Васильевич НКНбд-01-21

Содержание

Цель работы

1

Задание

1

Теоретическое введение

2

Выполнение лабораторной работы

2

Выводы

6

Цель работы

Приобретение практических навыков работы с именованными каналами.

Задание

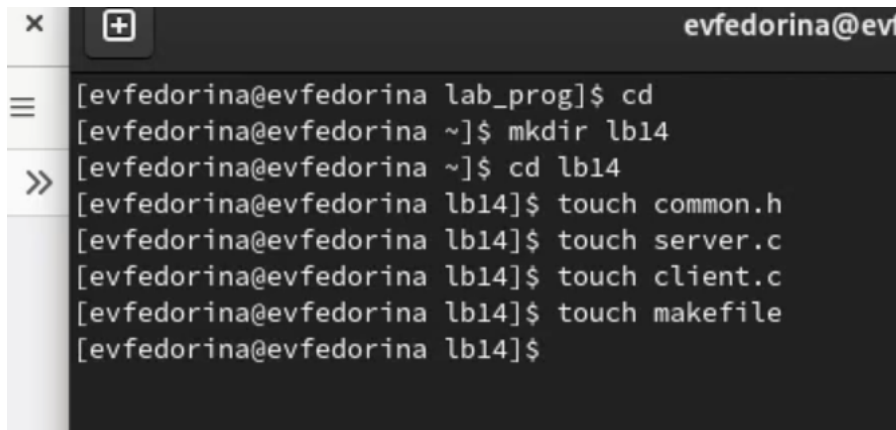
Изучите приведённые в тексте программы `server.c` и `client.c`. Взяв данные примеры за образец, напишите аналогичные программы, внося следующие изменения: 1. Работает не 1 клиент, а несколько (например, два). 2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). 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. Что будет в случае, если сервер завершит работу, не закрыв канал?

Теоретическое введение

Одним из видов взаимодействия между процессами в операционных системах является обмен сообщениями. Под сообщением понимается последовательность байтов, передаваемая от одного процесса другому. В операционных системах типа UNIX есть 3 вида межпроцессорных взаимодействий: общепонимание (именованные каналы, сигналы), System V Interface Definition (SVID — разделяемая память, очередь сообщений, семафоры) и BSD (сокеты). Для передачи данных между неродственными процессами можно использовать механизм именованных каналов (named pipes). Данные передаются по принципу FIFO (First In First Out) (первым записан — первым прочитан), поэтому они называются также FIFO pipes или просто FIFO. Именованные каналы отличаются от неименованных наличием идентификатора канала, который представлен как специальный файл (соответственно имя именованного канала — это имя файла). Поскольку файл находится на локальной файловой системе, данное IPC используется внутри одной системы.

Выполнение лабораторной работы

В домашнем каталоге создали подкаталог создали нужные файлы и подкаталоги. (рис.1)

A screenshot of a terminal window with a dark background. The window title is "evfedorina@evf". The terminal shows a series of commands and their outputs. The user starts in the "lab_prog" directory, then moves to the home directory and creates a directory named "lb14". They then move into "lb14" and create four files: "common.h", "server.c", "client.c", and "makefile".

```
x [ + evfedorina@evf
[evfedorina@evfedorina lab_prog]$ cd
[evfedorina@evfedorina ~]$ mkdir lb14
[evfedorina@evfedorina ~]$ cd lb14
[evfedorina@evfedorina lb14]$ touch common.h
[evfedorina@evfedorina lb14]$ touch server.c
[evfedorina@evfedorina lb14]$ touch client.c
[evfedorina@evfedorina lb14]$ touch makefile
[evfedorina@evfedorina lb14]$
```

{рис.1}

Запустили программы из примера, а затем модернизировали файлы client.c, common.h, server.c, как требовалось в задании(рис.2,3,4)

```
common.h [-M--] 6 L:[ 1+16 17/ 17] *(373 / 373b) <EOF>
/*
2 * common.h - заголовочный файл со стандартными определениями
3 */

#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>
#include <time.h>
#define FIFO_NAME "/tmp/fifo"
#define MAX_BUFF 80
#endif
```

{рис.2}

```
server.c [-M--] 0 L:[ 15+23 38/ 54] *(1246/1672b) 0032 0x020
char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */
/* баннер */
printf("FIFO Server...\n");
/* создаем файл FIFO с открытыми для всех
 * правами доступа на чтение и запись
 */
if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
<----->{
<----->    fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
<-----><----->    __FILE__, strerror(errno));
<----->    exit(-1);
<----->}

/* откроем FIFO на чтение */
if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
<----->{
<----->    fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
<-----><----->    __FILE__, strerror(errno));
<----->    exit(-2);
}

/* читаем данные из FIFO и выводим на экран */
while((n = read(readfd, buff, MAX_BUFF)) > 0)
<----->{
    if(write(1, buff, n) != n)
<----->{
    fprintf(stderr, "%s: Ошибка вывода (%s)\n",
    __FILE__, strerror(errno));
    exit(-3);
    }
}
close(readfd); /* закроем FIFO */

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перечитать 7Поиск 8Удалить 9МенюМС 1
```

{рис.3}

```

client.c      [-M--]  5 L:[  5+14  19/ 35] *(610 /1087b) 0010 0x00A  [*]
* 1. запустить программу server на одной консоли;
* 2. запустить программу client на другой консоли.
*/
#include "common.h"
#define MESSAGE "Hello Server!!!\n"
int
main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;
    /* баннер */
    printf("FIFO Client...\n");
    /* 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.4}

Скомпилировали модернизированные программы с помощью gcc(рис.5)

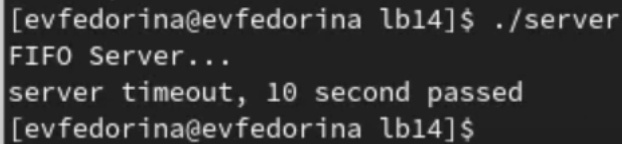
```

[evfedorina@evfedorina lb14]$ gcc client.c -o client
client.c: В функции «main»:
client.c:20:9: предупреждение: неявная декларация функции «sleep» [-Wimplicit-function-declaration]
   20 |         sleep(5);
      |         ^~~~~
client.c:35:12: предупреждение: неявная декларация функции «write»; имелось в
client.c:35:12: предупреждение: [implicit-function-declaration]
   35 |         if(write(writefd, MESSAGE, msglen) != msglen)
      |            ^~~~~
client.c:43:5: предупреждение: неявная декларация функции «close»; имелось в
client.c:43:5: предупреждение: [implicit-function-declaration]
   43 |         close(writefd);
      |         ^~~~~
client.c:43:5: предупреждение: [implicit-function-declaration]
client.c:43:5: предупреждение: [implicit-function-declaration]
[evfedorina@evfedorina lb14]$ gcc server.c -o server

```

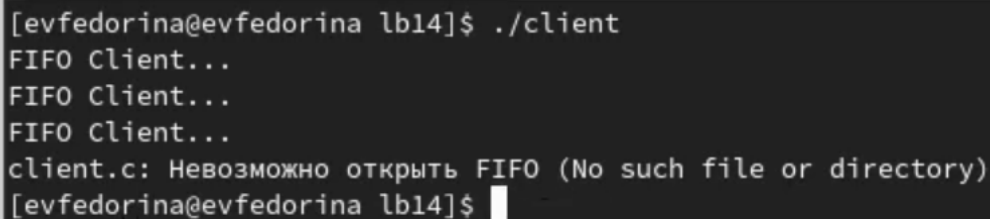
{рис.5}

Запустили server и client в разных консолях, а client дублировали в третьей консоли. Как можно увидеть, все условия выполняются.(рис.6,7,8)



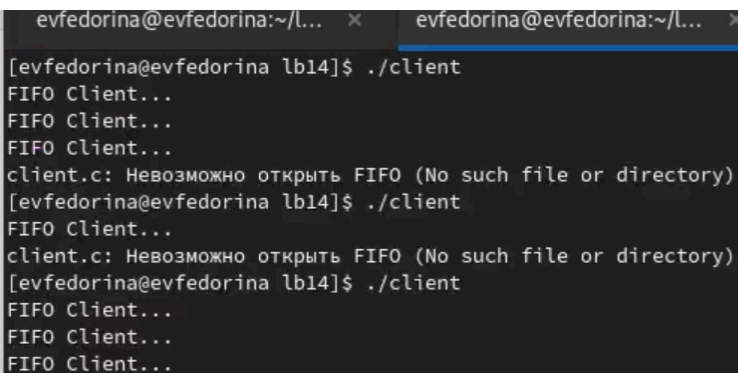
```
[evfedorina@evfedorina lb14]$ ./server
FIFO Server...
server timeout, 10 second passed
[evfedorina@evfedorina lb14]$
```

{рис.6}



```
[evfedorina@evfedorina lb14]$ ./client
FIFO Client...
FIFO Client...
FIFO Client...
client.c: Невозможно открыть FIFO (No such file or directory)
[evfedorina@evfedorina lb14]$
```

{рис.7}



```
evfedorina@evfedorina:~/l... x  evfedorina@evfedorina:~/l... x  evfedorina@evfedorina:~/l...
[evfedorina@evfedorina lb14]$ ./client
FIFO Client...
FIFO Client...
FIFO Client...
client.c: Невозможно открыть FIFO (No such file or directory)
[evfedorina@evfedorina lb14]$ ./client
FIFO Client...
client.c: Невозможно открыть FIFO (No such file or directory)
[evfedorina@evfedorina lb14]$ ./client
FIFO Client...
FIFO Client...
FIFO Client...
```

{рис.8}

Выводы

Приобрели практические навыки работы с именованными каналами.