Лабораторная работа №14

Кочкарев “sakochkarev” Станислав

RUDN University

Цель работы

Приобретение практических навыков работы с именованными каналами.

Задание

Написать программы по примеру приведенных со следующими изменениями:

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

Первым делом мы взяли за основу приведенные тексты программ и создали файлы с данными текстами.

common.h | server.c | client.c

```
1   //
2   //#ifndef LAB14_COMMON_H
3   //#define LAB14_COMMON_H
4   //
5   //#endif //LAB14_COMMON_H
6   //
7   /*
8    * common.h - заголовочный файл со стандартными определениями
9    */
10  #ifndef __COMMON_H__
11  #define __COMMON_H__
12
13  #include <stdio.h>
14  #include <stdlib.h>
15  #include <string.h>
16  #include <errno.h>
17  #include <sys/types.h>
18  #include <sys/stat.h>
19  #include <fcntl.h>
20  #include <unistd.h>
21
22  #define FIFO_NAME       "/Users/squidass/tmp/fifo"
23  #define MAX_BUFF        80
```

common.h | server.c | client.c

```c
/*
 * server.c - реализация сервера
 *
 * чтобы запустить пример, необходимо:
 *  1. запустить программу server на одной консоли;
 *  2. запустить программу client на другой консоли.
 */
#include "common.h"
#include "time.h"

int main() {
    int readfd; /* дескриптор для чтения из FIFO */
    int n;
    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 (mkfifo(FIFO_NAME, 0666) < 0) {
        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n", __FILE__, strerror( errnum: 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( errnum: errno));
        exit(-2);
    }

    clock_t start_time = time(NULL);

/* читаем данные из FIFO и выводим на экран */
    while ((n = read(readfd, buff, MAX_BUFF)) > 0 && time(NULL) - start_time < 30) {
```

common.h | server.c | client.c

```c
/*
 * client.c - реализация клиента
 *
 * чтобы запустить пример, необходимо:
 *  1. запустить программу server на одной консоли;
 *  2. запустить программу client на другой консоли.
 */
#include "common.h"
#include "time.h"

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( errnum: errno));
        exit(-1);
    }

    while (1) {
        /* передадим сообщение серверу */
        time_t rawtime;
        struct tm *timeinfo;
        time(&rawtime);
        timeinfo = localtime(&rawtime);
        char *message = asctime(timeinfo);
        msglen = strlen( s: message);
        if (write( fd: writefd, buf: message, nbyte: msglen) != msglen) {
            fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
                    __FILE__, strerror( errnum: errno));
            exit(-2);
        }
// fprintf(stdout, "test %ld", clock());
```

После этого мы начали работать с программой client.c.

Там мы добавили второе приведенное изменение – отправку текущего времени (timestamp) каждые 5 секунд. Для приостановки работы клиента была использована функция `sleep()`.

```c
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( errnum: errno));
        exit(-1);
    }

    while (1) {
        /* передадим сообщение серверу */
        time_t rawtime;
        struct tm *timeinfo;
        time(&rawtime);
        timeinfo = localtime(&rawtime);
        char *message = asctime(timeinfo);
        msglen = strlen( s: message);
        if (write( fd: writefd, buf: message, nbyte: msglen) != msglen) {
            fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
                    __FILE__, strerror( errnum: errno));
            exit(-2);
        }
//        fprintf(stdout, "test %ld", clock());
        sleep(5);
    }
    /* закроем доступ к FIFO */
    close(writefd);
    exit(0);
}
```

Далее были произведены изменения в файле `server.c`. В нем были добавлены изменения из 3-его задания – прекращение работы по истечению 30 секунд. Для этого была использована функция `time()`.

```c
    if (mkfifo(FIFO_NAME, 0666) < 0) {
        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n", __FILE__, strerror( errnum: 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( errnum: errno));
        exit(-2);
    }

    clock_t start_time = time(NULL);

/* читаем данные из FIFO и выводим на экран */
    while ((n = read(readfd, buff, MAX_BUFF)) > 0 & time(NULL) - start_time < 30) {
        if (write( fd: 1, buf: buff, nbyte: n) != n) {
            fprintf(stderr, "%s: Ошибка вывода (%s)\n", __FILE__, strerror( errnum: errno));
            exit(-3);
        }
    }
    close(readfd); /* закроем FIFO */
    /* удалим FIFO из системы */
    if (unlink(FIFO_NAME) < 0) {
        fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n", __FILE__, strerror( errnum: errno));
        exit(-4);
    }
    exit(0);
}
```

Во время работы над файлами изменения были протестированы.

В результате этих тестов было выяснено, что если сервер завершит работу не закрыв канал, то файл канала останется и при следующем запуске сервера он выдаст ошибку.

```
sakochkarev@sakochkarev [21:07:10] [~/work/study/2021-2022/Операционные системы/os-intro/labs/lab14] [master *]
-> % ./server
FIFO Server...
server.c: Невозможно создать FIFO (File exists)
```

После выполнения всех заданий были произведены финальные тесты, которые подтвердили работоспособность и корректность выполнения всех заданий.

```
sakochkarev@sakochkarev [21:14:04] [~/work/study/2021-2022/Операционные системы/os-intro/labs/lab14] [master *]
-> % ./server
FIFO Server...
Mon May 30 21:16:41 2022
Mon May 30 21:16:43 2022
Mon May 30 21:16:46 2022
Mon May 30 21:16:48 2022
Mon May 30 21:16:51 2022
Mon May 30 21:16:53 2022
Mon May 30 21:16:56 2022
Mon May 30 21:16:58 2022
Mon May 30 21:17:01 2022
Mon May 30 21:17:03 2022
Mon May 30 21:17:06 2022
Mon May 30 21:17:08 2022
sakochkarev@sakochkarev [13:38:35] [~/work/study/2021-2022/Операционные системы/os-intro/labs/lab14] [master *]
```

```
sakochkarev@sakochkarev [21:14:09] [~/work/study/2021-2022/Операционные системы/os-intro/labs/lab14] [master *]
-> % ./client
FIFO Client...
sakochkarev@sakochkarev [13:39:16] [~/work/study/2021-2022/Операционные системы/os-intro/labs/lab14] [master *]
```

Выводы

По выполнении данной лабораторной работы мы приобрели практические навыки работы с именованными каналами.