

Отчёт по лабораторной работе №14

Именованные каналы

Кочина Д. С.

10 мая 2023

Российский университет дружбы народов, Москва, Россия

Вводная часть

Целью данной лабораторной работы является приобретение практических навыков работы с именованными каналами.

Основная часть

- Для начала изучили материал лабораторной работы. Далее на основе примеров напишем аналогичные программы, но с изменениями.
- Создадим необходимые файлы для работы.

```
dskochina@dk3n31 ~ $ touch common.h server.c client.c Makefile  
dskochina@dk3n31 ~ $ emacs
```

- Затем изменим коды программ, данных в лабораторной работе. В файл common.h добавим стандартные заголовочные файлы: “unistd.h”, “time.h”. Это необходимо для работы других файлов. Этот файл является заголовочным, чтобы в остальных программах не прописывать одно и то же каждый раз.

Файл common.h

```
/*  
 * common.h - заголовочный файл со стандартными определениями  
 */  
#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 <unistd.h>  
#include <time.h>  
  
#define FIFO_NAME "/tmp/fifo"  
#define MAX_BUFF 80  
  
#endif /* __COMMON_H__ */
```

- Затем в файл server.c добавляем цикл “while” для контроля за временем работы сервера. Разница между текущим временем и началом работы не должна превышать 30 секунд.

Файл server.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"

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(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);
    }
}
```

```
/* 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);
}

/* начало отсчёта времени */
clock_t start = time(NULL);

/* цикл работает пока с момента начала отсчёта времени прошло меньше 30 секунд */
while(time(NULL)-start < 30)
{
    /* 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 */

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

- В файл client.c добавим цикл, который отвечает за количество сообщений о текущем времени (4 сообщения). С помощью команды “sleep” приостановим работу клиента на 5 секунд.

```
/*
 * 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"

#define MESSAGE "Hello Server!!!\n"

int main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;

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

    /* цикл отвечающий за отправку сообщения о текущем времени */
    for(int i=0; i<4; i++)
    {
        /* 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);
            break;
        }

        /* текущее время */
        long int ttime=time(NULL);
        char* text=ctime(&ttime);
```

```
/* текущее время */
long int ttime=time(NULL);
char* text=ctime(&ttime);

/* 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);
}
/* приостановка работы клиента на 5 секунд */
sleep(5)
}

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

- Makefile оставили без изменений.

```
all: server client

server: server.c common.h
    gcc server.c -o server

client: client.c common.h
    gcc client.c -o client

clean:
```

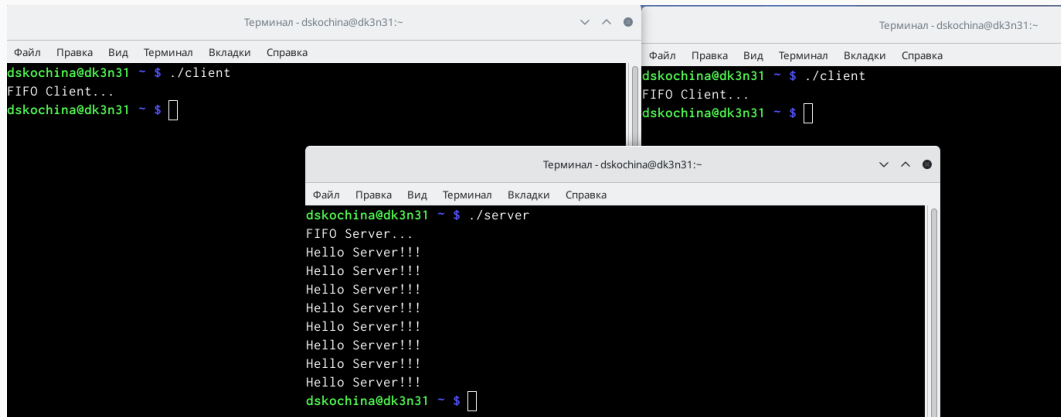
```
rm server client *.o
```

- Далее делаем компиляцию файлов с помощью команды “make all”.

```
dskochina@dk3n31 ~ $ make all  
gcc client.c -o client  
dskochina@dk3n31 ~ $
```

- Затем открываем три терминала для проверки работы наших файлов. В первом пишем “./server”, а в остальных “./client”. В результате каждый терминал вывел по 4 сообщения, а по истечении 30 секунд работа сервера была завершена. Всё работает верно.

Проверка



```
Терминал - dskochina@dk3n31:~  
Файл  Правка  Вид  Терминал  Вкладки  Справка  
dskochina@dk3n31 ~ $ ./client  
FIFO Client...  
dskochina@dk3n31 ~ $  
  
Терминал - dskochina@dk3n31:~  
Файл  Правка  Вид  Терминал  Вкладки  Справка  
dskochina@dk3n31 ~ $ ./client  
FIFO Client...  
dskochina@dk3n31 ~ $  
  
Терминал - dskochina@dk3n31:~  
Файл  Правка  Вид  Терминал  Вкладки  Справка  
dskochina@dk3n31 ~ $ ./server  
FIFO Server...  
Hello Server!!!  
Hello Server!!!  
Hello Server!!!  
Hello Server!!!  
Hello Server!!!  
Hello Server!!!  
Hello Server!!!  
Hello Server!!!  
dskochina@dk3n31 ~ $
```

- Проверим длительность работы сервера. Вводим команду “./server” в одном терминале. Он завершил свою работу через 30 секунд. Если сервер завершит свою работу, не закрыва канал, то при повторном запуске появится ошибка “Невозможно создать FIFO”, так как у нас уже есть один канал.

```
dskochina@dk3n31 ~ $ ./server
FIFO Server...
server.c: Невозможно создать FIFO (File exists)
dskochina@dk3n31 ~ $
```

Заключение

В ходе выполнения данной лабораторной работы я приобрела практические навыки работы с именованными каналами.