

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ
НАРОДОВ**

**Факультет физико-математических и естественных
наук**

**Фундаментальная Информатика и Информационные
технологии**

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 14

дисциплина: операционные системы

Узор-Ежикеме Чинечелум А.

НФИбд-03-21

1032215263

Цель работы

Приобретение практических навыков работы с именованными каналами.

Последовательность выполнения работы

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?

Взяв примеры в примечаниях в качестве образца, я написал аналогичные программы, внося следующие изменения:

- Работает не 1 клиент, а несколько.

```
#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>

#define FIFO_NAME      "/tmp/fifo"
#define MAX_BUFF      80

#endif/* __COMMON_H__ */
~
~
```

```

#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);
}

```

- Клиенты передают текущее время с некоторой периодичностью

```

#include "common.h"

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

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

        /* 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);
    }
}

```

Создал makefile:

```

makefile      [----] 0 L:[ 1+ 7 8/
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

```

```
cuzorezhikeme@dk8n59 ~ $ ./server
FIFO Server...
Hello Server!!!
cuzorezhikeme@dk8n59 ~ $ ./server
FIFO Server...
Hello Server!!!
cuzorezhikeme@dk8n59 ~ $ ./client
FIFO Client...
cuzorezhikeme@dk8n59 ~ $ ./client
FIFO Client...
client.c: Невозможно открыть FIFO (No such file or directory)
cuzorezhikeme@dk8n59 ~ $ ./client
FIFO Client...
—
```

Контрольные вопросы

1. В чем ключевое отличие именованных каналов от неименованных?
-
2. Возможно ли создание неименованного канала из командной строки?
- Да, возможно
3. Возможно ли создание именованного канала из командной строки?
- Да, возможно
4. Опишите функцию языка C, создающую неименованный канал.
- Для того, чтобы создать неназванный канал, используется функция, вызываемая т.е. `pipe()`.
5. Опишите функцию языка C, создающую именованный канал.
- Для создания файла FIFO/именованного канала используется функция, вызываемая, например, *mkfifo*.
6. Что будет в случае прочтения из `fifo` меньшего числа байтов, чем находится в канале?
Большого числа байтов?
- Нет, меньшее числа байтов.
7. Аналогично, что будет в случае записи в `fifo` меньшего числа байтов, чем позволяет буфер? Большого числа байтов?
- Да.

8. Могут ли два и более процессов читать или записывать в канал?

- Да, несколько процессов могут считывать данные из канала (или записывать в него)

9. Опишите функцию `strerror`.

- Функция `strerror()` возвращает указатель на строку, которая описывает код ошибки, переданный в аргументе `errnum`, возможно, используя часть `LC_MESSAGES` текущей локали для выбора соответствующего языка.