

# Лабораторная работа №15

Колосова Кристина Александровна

## Цель работы

Приобретение практических навыков работы с именованными каналами.

## Выполнение лабораторной работы

Для передачи данных между неродственными процессами можно использовать механизм именованных каналов (named pipes). Данные передаются по принципу FIFO (First In First Out)

Файлы именованных каналов создаются функцией `mkfifo`

```
int mkfifo(const char *pathname, mode_t mode);
```

Первый параметр — имя файла, идентифицирующего канал, второй параметр — маска прав доступа к файлу.

После создания файла канала процессы, участвующие в обмене данными, должны открыть этот файл либо для записи, либо для чтения. При закрытии файла сам канал продолжает существовать. Для того чтобы закрыть сам канал, нужно удалить его файл, например с помощью вызова `unlink`

Каналы представляют собой простое и удобное средство передачи данных, которое, однако, подходит не во всех ситуациях. Например, с помощью каналов довольно трудно организовать обмен асинхронными сообщениями между процессами.

Изучила приведённые в тексте программы `server.c` и `client.c`. Взяв данные примеры за образец, написала 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.

```
GNU nano 4.8 common.h
#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 /* __COMMON_H__ */
```

Рис. 1. Файл common.h

```
kakolosova@LAPTOP-HOEP161O: ~
GNU nano 4.8 server.c
#include "common.h"
int main(){
int readfd;
int n;
char buff[MAX_BUFF];
printf("FIFO Server...\n");
if(mknod(FIFO_NAME, S_IFIFO | 0666, 0)){
    fprintf(stderr, "%s: unable to create FIFO (%s)\n", __FILE__, strerror(errno));
    exit(-1);
}
int start=time(NULL);
while(time(NULL)-start<=30){
    if((readfd=open(FIFO_NAME, O_RDONLY))<0){
        fprintf(stderr, "%s: unable to open FIFO (%s)\n", __FILE__, strerror(errno));
        exit(-2);
    }
    while((n=read(readfd, buff, MAX_BUFF))>0){
        if(write(1, buff, n)!=n){
            fprintf(stderr, "%s: Input error(%s)\n", __FILE__, strerror(errno));
            exit(-3);
        }
    }
}
close(readfd);
if(unlink(FIFO_NAME)<0){
    fprintf(stderr, "%s: Unable to delete FIFO (%s)", __FILE__, strerror(errno));
    exit(-4);
}
exit(0);
}
```

Рис. 2. Файл server.c

```
kakolosova@LAPTOP-HOEP1610: ~
GNU nano 4.8 client.c
#include "common.h"
#include <unistd.h>
#define MESSAGE "Hello, server!\n"
int main(){
int writefd;
int msglen;
printf("FIFO Client...\n");
if((writefd=open(FIFO_NAME, O_WRONLY))<0){
    fprintf(stderr, "%s: Unable to open FIFO (%s)\n", __FILE__, strerror(errno));
    exit(-1);
}
int i;
for (i=0; i<5; i++){
    sleep(5);
    long ttime=time(NULL);
    msglen=strlen(ctime(&ttime));
    if(write(writefd, ctime(&ttime), msglen)!=msglen){
        fprintf(stderr, "%s: Write error FIFO (%s)\n", __FILE__, strerror(errno));
        exit(-2);
    }
}
close(writefd);
exit(0);
}
```

Рис. 3. Файл client.c

Результат:

```
kakolosova@LAPTOP-HOEP1610:~$ ./server
FIFO Server...
```

Рис. 4. Вызов server

```
kakolosova@LAPTOP-HOEP1610:~$ make
gcc client.c -o client
kakolosova@LAPTOP-HOEP1610:~$ ./client
FIFO Client...
```

Рис. 5. Вызов client

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

- при закрытии сервера до закрытия каналов ничего не произошло - они автоматически закрываются при завершении программы. Но закрывать их необходимо. Если этого не делать – некорректное завершение приведет к проблемам.

## Выводы

Приобрела практические навыки работы с именованными каналами.

## Библиография

[c - make: Nothing to be done for `all' - Stack Overflow](#)

[c - Sleep | warning implicit declaration of function `sleep'? - Stack Overflow](#)

[Что такое каналы \(pipe\) в Linux? Как работает перенаправление каналов? \(andreyex.ru\)](#)