

# Отчёт по лабораторной работе №13

## Именованные каналы

Архипов Юрий Денисович

### Цель работы

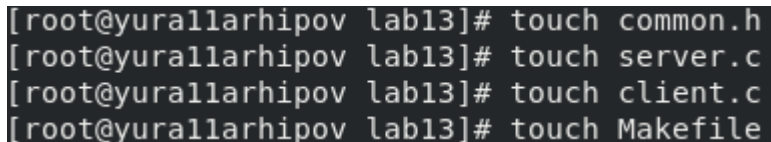
Приобретение практических навыков работы с именованными каналами.

### Ход работы

Изучить приведённые в тексте программы server.c и client.c. Взяв данные примеры за образец, написать аналогические 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). Использовать функцию sleep() для приостановки работы клиента.
3. Сервер работает не бесконечно, а прекратит работу через некоторое время (например, 30 сек). Использовать функцию clock() для определения времени работы сервера. Что будет в случае, если сервер завершит работу, не закрыв канал?

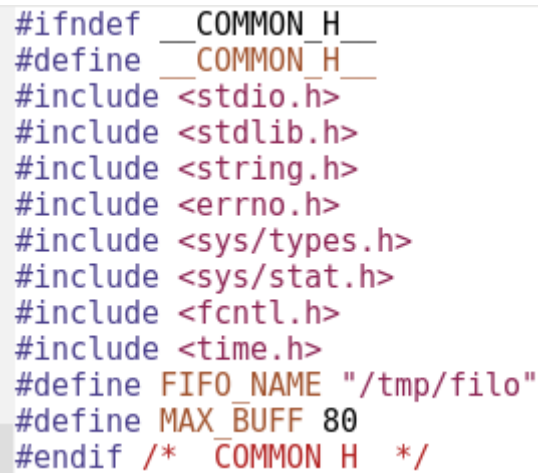
- 1) Создание файлов.



```
[root@yurallarhipov lab13]# touch common.h
[root@yurallarhipov lab13]# touch server.c
[root@yurallarhipov lab13]# touch client.c
[root@yurallarhipov lab13]# touch Makefile
```

Рис.1

- 2) Написание кодов в файлах.



```
#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/filo"
#define MAX_BUFF 80
#endif /* __COMMON_H__ */
```

Рис.2 common.h

Рис.3 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: Невозможно создать 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: Невозможно открыть 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: Ошибка вывода (%s)\n",__FILE__, strerror(errno));
                exit(-3);
            }
        }
        close(readfd);
        if(unlink(FIFO_NAME) < 0){
            fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",__FILE__,strerror(errno));
            exit(-4);
        }
    }
    exit(0);
}
```

Рис.4 client.c

```
#include "common.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: невозможно открыть FIFO (%s)\n",__FILE__,strerror(errno));
        exit(-1);
    }
    int i;
    for(i = 0; i < 5; i++){
        long ttime = time(NULL);
        msglen = strlen(ctime(&ttime));
        if(write(writefd,ctime(&ttime),msglen) !=msglen){
            fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",__FILE__,strerror(errno));
            exit(-2);
        }
    }
    close(writefd);
    exit(0);
}
```

```
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
```

Рис.5 Makefile

```

ser[root@yurallarhipov lab13]# ./server
[yurallarhipov@yurallarhipov lab13]# ./client
FIFO Server...
FIFO Client...
client.c: невозможно открыть FIFO (Permission denied)
[yurallarhipov@yurallarhipov lab13]$

```

Рис.6

Если не закрыть канал при завершении работы, могут случиться различные ошибки.

## Вывод

Я приобрёл практические навыки работы с именованными каналами.