

Отчет по выполнению лабораторной работы №14

Дисциплина: операционные системы

Астраханцева А. А.

7 мая 2023

Российский университет дружбы народов, Москва, Россия

- Астраханцева Анастасия Александровна
- студентка НКАбд-01-22
- Студ. билет: 1132226437
- Российский университет дружбы народов
- <https://anastasiia7205.github.io/>

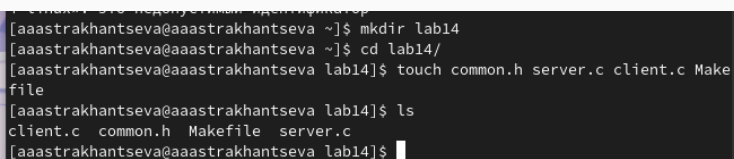


Приобретение практических навыков работы с именованными каналами.

Изучите приведённые в тексте программы `server.c` и `client.c`. Взяв данные примеры за образец, напишите 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. Что будет в случае, если сервер завершит работу, не закрыв канал?

Выполнение лабораторной работы

Создаем нужные файлы и записываем в них текст программы

A terminal window with a dark background and light-colored text. The text shows a series of commands being executed in a shell. The user is 'aaastrakhantseva' and the current directory is '~'. The commands are: 'mkdir lab14', 'cd lab14/', 'touch common.h server.c client.c Makefile', and 'ls'. The output of 'ls' shows 'client.c common.h Makefile server.c'. The prompt returns to the user's home directory.

```
[aaastrakhantseva@aaastrakhantseva ~]$ mkdir lab14
[aaastrakhantseva@aaastrakhantseva ~]$ cd lab14/
[aaastrakhantseva@aaastrakhantseva lab14]$ touch common.h server.c client.c Makefile
[aaastrakhantseva@aaastrakhantseva lab14]$ ls
client.c common.h Makefile server.c
[aaastrakhantseva@aaastrakhantseva lab14]$
```

Рис. 1: Создание файлов

Исправленный текст файла client.c

```
Открыть + client.c
~/lab14

#include "common.h"

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


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

    /* Баннер */
    for(int i = 0; i < 10; i++){
        sleep(3);
        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 */
```

Текст файла common.h

```
Открыть ▼  common.h  
~ /lab14  
● client.c
```

```
/*  
 * 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>  
  
#define FIFO_NAME "/tmp/fifo"  
#define MAX_BUFF 80
```


Исправленный текст файла server.c



```
Открыть ▾ + server.c
~/lab14
● client.c common.h

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);
}

clock_t begining=time(NULL), clock_t now=line(NULL);
/* читаем данные из FIFO и выводим на экран */
while(begining - now < 30){
    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);
        }
    }
    begining = time(NULL);
}

close(readfd); /* закроем FIFO */
```

Текст файла Makefile



```
Открыть ▾ + server.c
~lab14
client.c common.h

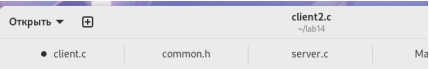
if(!mkfifo(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
{
    fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
    __FILE__, strerror(errno));
    exit(-1);
}

/* 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);
}

clock_t begining=time(NULL), clock_t now=line(NULL);
/* читаем данные из FIFO и выводим на экран */
while(begining - now < 30){
    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);
        }
    }
    begining = time(NULL);
}

close(readfd); /* закроем FIFO */
```

Текст файла client2.c



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

    /* баннер */
    for(int i = 0; i < 15; i++){
        ttime = time(NULL);
        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);
        }
        sleep(4);
    }
}
```

Далее прописываем в консоли команду **make** и получаем исполняемые файлы



client



client.c



client2



client2.c



common.h



Makefile



server



server.c



В ходе выполнения лабораторной работы №14 я приобрела практические навыки работы с именованными каналами.