

Отчёт по лабораторной работе №14

дисциплина “Операционные системы”

Мишонков М. А.

12 мая 2023

Российский университет дружбы народов, Москва, Россия

Вводная часть

- Целью выполнения данной лабораторной работе является приобретение навыков работы с именованными каналами.

Основная часть

- Создал необходимые файлы.

```
mamishonkov@dk3n38 ~ $ cd ~/work/study/2022-2023/Операционные\ системы/os-intro/labs/lab14
mamishonkov@dk3n38 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14 $ touch common.h
mamishonkov@dk3n38 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14 $ touch server.c
mamishonkov@dk3n38 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14 $ touch client.c
mamishonkov@dk3n38 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14 $ touch Makefile
mamishonkov@dk3n38 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14 $
```

Файл “common.h”

- Изменил коды программ, данных в лабораторной работе. В файл common.h добавил 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.

```
/*  
 * 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>
```

Файл “server.c”

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

```
/*
 * 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)
```

Файл "server.c"

```
/* 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);
}
```


Файл "client.c"

- В файл client.c добавил цикл, который отвечает за количество сообщений о текущем времени (4 сообщения). С помощью команды "sleep" приостановил 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)
        {
```

```
/* 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 оставил 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

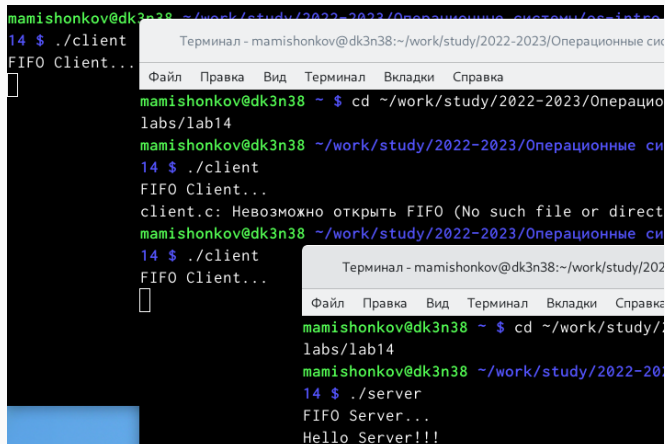
clean:
    rm server client *.o
```

- Сделал компиляцию файлов с помощью команды “make all”.

```
mamishonkov@dk3n38 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14 $ make all
gcc server.c -o server
gcc client.c -o client
[4]+  Завершён          emacs
mamishonkov@dk3n38 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14 $
```

Проверка работы

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



```
mamishonkov@dk3n38 ~ $ cd ~/work/study/2022-2023/Операционные системы/ass1/term
14 $ ./client
FIFO Client...

mamishonkov@dk3n38 ~ $ cd ~/work/study/2022-2023/Операционные системы/ass1/term
14 $ ./client
FIFO Client...
client.c: Невозможно открыть FIFO (No such file or directory)
mamishonkov@dk3n38 ~ $ cd ~/work/study/2022-2023/Операционные системы/ass1/term
14 $ ./client
FIFO Client...

mamishonkov@dk3n38 ~ $ cd ~/work/study/2022-2023/Операционные системы/ass1/term
14 $ ./server
FIFO Server...
Hello Server!!!
```

Заключение

- В ходе выполнения данной лабораторной работы приобрёл навыки работы с именованными каналами.