

Отчёт по лабораторной работе №14

Дисциплина “Операционные системы”

Батова Ирина Сергеевна, НММбд-01-22

06 мая 2023

Российский университет дружбы народов, Москва, Россия

Вводная часть

Приобретение практических навыков работы с именованными каналами.

Основная часть

- Для начала работы создаем командой 'touch' четыре файла

```
[isbatova@fedora ~]$ touch common.h server.c client.c Makefile  
[isbatova@fedora ~]$
```

- Для корректной работы других файлов добавляем к листингу из лабораторной работы два заголовочных файла - unistd.h и time.h

```
/*  
 * 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>  
#include <time.h>  
  
#define FIF0_NAME "/tmp/fifo"
```

- Нам нужно, чтобы сервер заканчивал работу через 30 секунд. Для реализации данного действия сначала обозначаем время начало работы (`clock_t start=time(NULL)`), а затем вносим прочтение данных из FIFO и вывод их на экран под цикл `while`, который работает только пока разница между текущим временем и временем начала работы меньше 30 секунд

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);
}

clock_t start = time(NULL);

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);
}
```


- Нам нужно, чтобы клиенты передавали текущее сообщение раз в пять секунд. Для этого добавляем цикл `for`, который анализирует количество и отправляет сообщения о текущем времени, добавляем команды для генерации этих сообщений (`long int ttime=time(NULL)` и `char* text=ctime(&ttime)`), а также команду `sleep(5)`, которая останавливает работу клиента на 5 секунд

Файл 'client.c'

```
/*
 * 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");

    for(int i=0; i<4; i++)
    {
        /* 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);
            break;
        }
    }
}
```

```
long int ttime=time(NULL);
char* text=ctime(&ttime);

/* 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(5);
}

/* закроем доступ к FIFO */
close(writefd);

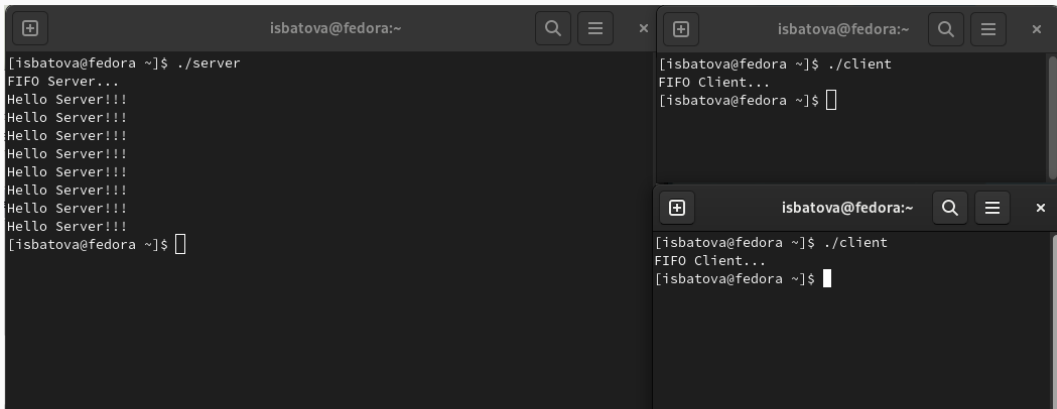
exit(0);
}
```

- В Makefile вводим листинг, соответствующий лабораторной работе, и оставляем без изменений
- Вводим команды 'make server' и 'make client' для компиляции исполняемых файлов

```
[isbatova@fedora ~]$ make server  
gcc server.c -o server  
[isbatova@fedora ~]$ make client  
gcc client.c -o client  
[isbatova@fedora ~]$
```

Запуск программы

- Открываем три окна терминала, в одном запускаем файл server, в двух других - файл client
- Каждый “клиент” вывел по четыре сообщения, а спустя тридцать секунд сервер завершил работу



The screenshot shows three terminal windows from the user isbatova@fedora. The top-left window runs the server, which prints 'FIFO Server...' and 'Hello Server!!!' eight times before exiting. The top-right window runs a client that prints 'FIFO Client...' and then waits. The bottom-right window runs another client that prints 'FIFO Client...' and then waits. The server's output indicates it received data from both clients before finishing.

```
[isbatova@fedora ~]$ ./server
FIFO Server...
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
[isbatova@fedora ~]$
```

```
[isbatova@fedora ~]$ ./client
FIFO Client...
[isbatova@fedora ~]$
```

```
[isbatova@fedora ~]$ ./client
FIFO Client...
[isbatova@fedora ~]$
```

Вывод

В ходе данной лабораторной работы я приобрела практические навыки работы с именованными каналами.