

Лабораторная работа №14

Операционные системы

Миронов Дмитрий Андреевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	12

Список иллюстраций

3.1	Процесс создания файлов	7
3.2	Изменил код <code>common.h</code>	8
3.3	Изменил код <code>server.c</code>	9
3.4	Изменил код <code>client.c</code>	10
3.5	Использование <code>make all</code>	10
3.6	Проверка работы сервера	11

Список таблиц

1 Цель работы

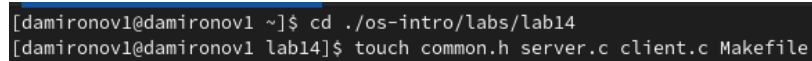
Приобретение практических навыков работы с именованными каналами.

2 Задание

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?

3 Выполнение лабораторной работы

1. Создал необходимые для работы файлы. (рис. 3.1)

A screenshot of a terminal window showing two commands being executed. The first command is `cd ./os-intro/labs/lab14` and the second command is `touch common.h server.c client.c Makefile`. The terminal output shows the current directory as `lab14` after the first command.

```
[damironov1@damironov1 ~]$ cd ./os-intro/labs/lab14
[damironov1@damironov1 lab14]$ touch common.h server.c client.c Makefile
```

Рис. 3.1: Процесс создания файлов

2. Изменил коды программ, показанных в файле лабораторной работы. (рис. 3.2)(рис. 3.3)(рис. 3.4)

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

#endif
```

Рис. 3.2: Изменил код common.h


```

int n;
char buff[MAX_BUFF];
printf("FIFO Server...\n");
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 start =time(NULL);

while(time(NULL)-start < 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);
        }
    }
}
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);
}

```

Рис. 3.3: Изменил код server.c

```

#include "common.h"
#define MESSAGE "Hello Server!!!\n"

int
main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;
    printf("FIFO Client...\n");
    for (int i = 0; i<4;i++)
    {
        if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-1);
        }
        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);
    }
    close(writefd);

    exit(0);
}

```

Рис. 3.4: Изменил код client.c

Makefile оставил в том виде, в котором он представлен.

3. После написания кодов использовал команду make all.(рис. 3.5)

```

[damironov1@damironov1 lab14]$ make all
gcc server.c -o server
gcc client.c -o client

```

Рис. 3.5: Использование make all

4. Запустил сервер и проверил его работу. (рис. 3.6)

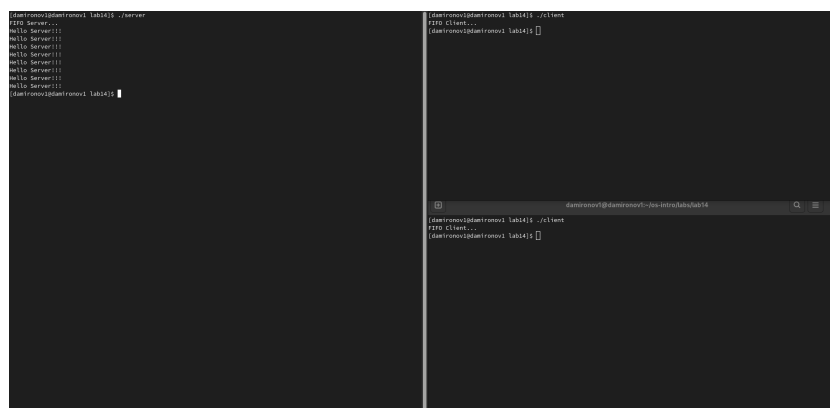


Рис. 3.6: Проверка работы сервера

Если сервер завершит работу, не закрыв канал,то, когда мы будем запускать его повторно, появится ошибка <>, так как у нас уже есть один канал.

4 Выводы

Задания были выполнены. Практические навыки по работе с именованными каналами были приобретены.