

Лабораторная работа No 14.

Именованные каналы

Валиева М. Р.

Российский университет дружбы народов, Москва, Россия

Информация

- Валиева Марина Русланбековна
- НБИбд-01-22
- Российский университет дружбы народов

Вводная часть

Приобретение практических навыков работы с именованными каналами.

Результаты

Элементы презентации

1. Для начала я создала необходимые файлы с помощью команды `touch` и открыла редактор `emacs` для их редактирования.


```
mrvalieva@dk3n57 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14  
$ touch common.h server.h client.c Makefile
```

Рис. 1: создание файлов

2. Далее я изменила коды программ, представленных в лабораторной работе. В файл `common.h` добавила стандартные заголовочные файлы `unistd.h` и `time.h`, необходимые для работы кодов других файлов. `Common.h` предназначен для заголовочных файлов, чтобы в остальных программах их не прописывать каждый раз.

Открыть ▾ + *common.h
~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14

```
1 #ifndef __COMMON_H__  
2 #define __COMMON_H__  
3  
4 #include <stdio.h>  
5 #include <stdlib.h>  
6 #include <string.h>  
7 #include <errno.h>  
8 #include <sys/types.h>  
9 #include <sys/stat.h>  
10 #include <fcntl.h>  
11 #include <fcntl.h>  
12 #include <unistd.h>  
13 #include <time.h>  
14  
15 #define FIFO_NAME "/tmp/fifo"  
16 #define MAX_BUFF 80  
17  
18 #endif
```

Рис. 2: изменение

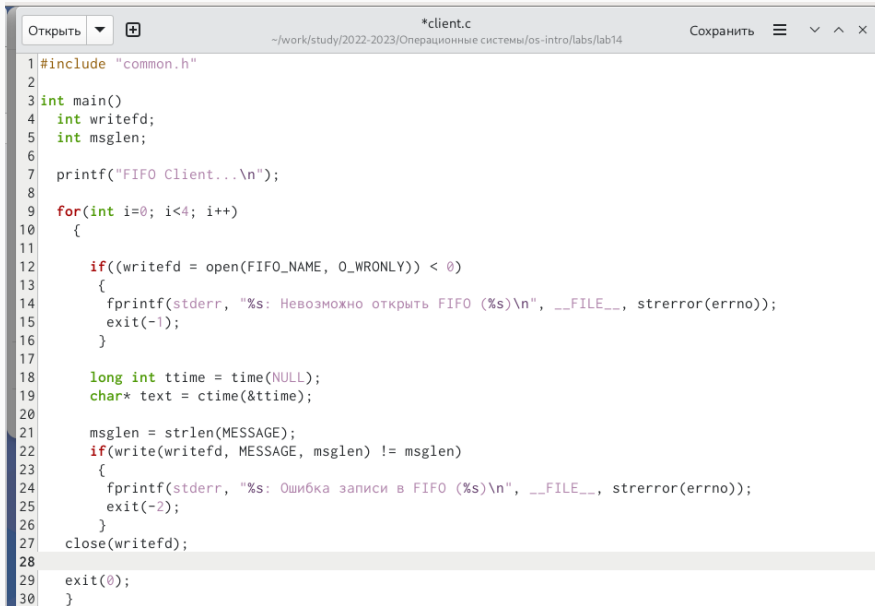
В файл `server.c` добавила цикл `while` для контроля за временем работы сервера. Разница между текущим временем `time(NULL)` и временем начала работы `clock_t start=time(NULL)` (инициализация до цикла) не должна превышать 30 секунд.

```
*server.c
~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14
Сохранить

1 #include "common.nt
2
3 int main() {
4     int readfd;
5     int n;
6     char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */
7
8     printf("FIFO Server...\n");
9
10    if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
11    {
12        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n", __FILE__, strerror(errno));
13        exit(-1);
14    }
15
16    if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
17    {
18        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n", __FILE__, strerror(errno));
19        exit(-2);
20    }
21
22    clock_t start = time(NULL);
23
24    while(time(NULL)-start < 30)
25    {
26        while((n = read(readfd, buff, MAX_BUFF)) > 0)
27        {
28            if(write(1, buff, n) != n)
29            {
30                fprintf(stderr, "%s: Ошибка вывода (%s)\n", __FILE__, strerror(errno));
31                exit(-3);
32            }
33        }
34    }
35
36    close(readfd); /* закроем FIFO */
37
38    if(unlink(FIFO_NAME) < 0)
39    {
40        fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n", __FILE__, strerror(errno));
41        exit(-4);
42    }
43
44    exit(0);
45 }
```

Рис. 3: изменение

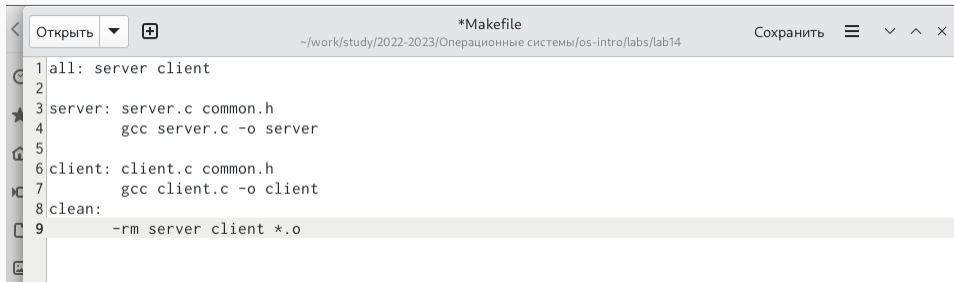
В файл `client.c` добавила цикл, который отвечает за количество сообщений о текущем времени (4 сообщения), которое получается в результате выполнения команд, и команду `sleep(5)` для приостановки работы клиента на 5 секунд.



```
1 #include "common.h"
2
3 int main()
4 {
5     int writefd;
6     int msglen;
7
8     printf("FIFO Client...\n");
9
10    for(int i=0; i<4; i++)
11    {
12        if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
13        {
14            fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n", __FILE__, strerror(errno));
15            exit(-1);
16        }
17
18        long int ttime = time(NULL);
19        char* text = ctime(&ttime);
20
21        msglen = strlen(MESSAGE);
22        if(write(writefd, MESSAGE, msglen) != msglen)
23        {
24            fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n", __FILE__, strerror(errno));
25            exit(-2);
26        }
27        close(writefd);
28    }
29    exit(0);
30 }
```

Рис. 4: изменение

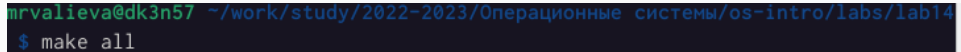
Makefile (файл для сборки) не изменяла



```
1 all: server client
2
3 server: server.c common.h
4     gcc server.c -o server
5
6 client: client.c common.h
7     gcc client.c -o client
8 clean:
9     -rm server client *.o
```

Рис. 5: неизменение

3. После написания кодов, я, используя команду «make all», скомпилировала необходимые файлы

A terminal window with a dark background. The prompt is 'mrvalieva@dk3n57' in green, followed by the path '~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14' in blue. The command '\$ make all' is entered in white.

```
mrvalieva@dk3n57 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14
$ make all
```

Рис. 6: компиляция

```
int val1, val2, val3, val4;  
$ ./server
```

Рис. 7: компиляция

```
mrval1eva@dk3n5 /  
$ ./client
```

Рис. 8: компиляция

Далее я проверила работу написанного кода. Открыла 3 консоли (терминала) и запустила: в первом терминале – «./server», в остальных двух – «./client». В результате каждый терминал-клиент вывел по 4 сообщения. Спустя 30 секунд работа сервера была прекращена. Программа работает корректно.

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

Спасибо за внимание!