

Лабораторная работа №14. Именованные каналы

Парфенова Елизавета Евгеньевна

RUDN University, Moscow, Russian Federation

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?

Пишем аналогичные программы и помещаем их в папку. (рис. 1)

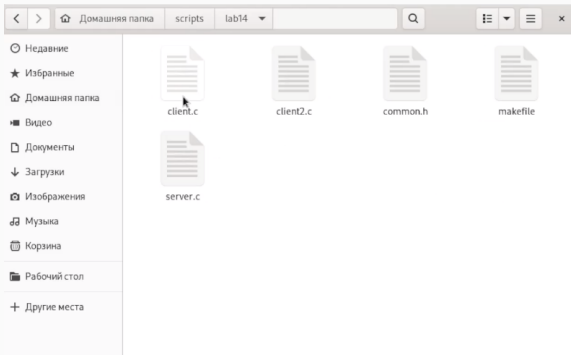
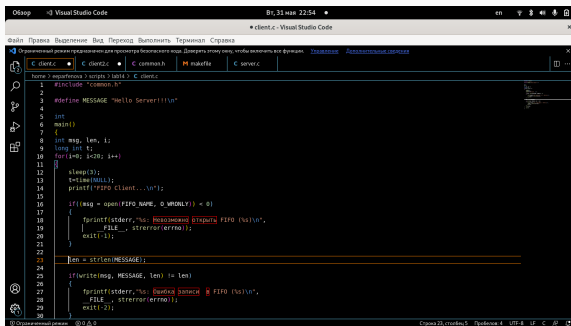


Figure 1: Папка с программами

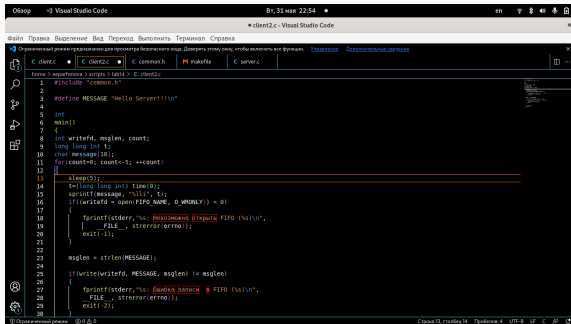
1. client.c (рис. 2)



```
1 #include "common.h"
2
3 #define MESSAGE "Hello Server!!!\n"
4
5 int
6 main()
7 {
8     int msg, len, i;
9     long int t;
10    for(i=0; i<20; i++)
11    {
12        sleep(3);
13        t=time(NULL);
14        printf("File client...\n");
15
16        if((msg = open(PIFO_NAME, O_WRONLY)) < 0)
17        {
18            fprintf(stderr, "Невозможно открыть FIFO (%s)\n",
19                    _FILE_, strerror(errno));
20            exit(-1);
21        }
22
23        len = strlen(MESSAGE);
24
25        if(write(msg, MESSAGE, len) != len)
26        {
27            fprintf(stderr, "Невозможно записать в FIFO (%s)\n",
28                    _FILE_, strerror(errno));
29            exit(-2);
30        }
```

Figure 2: client.c

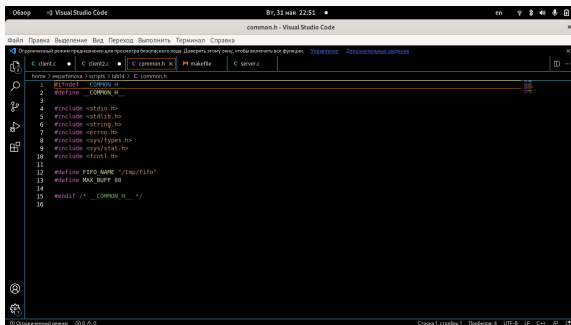
2. client2.c (рис. 3)



```
1 #include <stdio.h>
2
3 #define MESSAGE "Hello Server!!\n"
4
5 int
6 main()
7 {
8     int writefd, msglen, count;
9     long long int t;
10     char message[10];
11     for(count=0; count<3; ++count)
12     {
13         sleep(5);
14         t=long long int) time(0);
15         sprintf(message, "t=%d", t);
16         if(writefd = open(FIFO_NAME, O_WRONLY) < 0)
17         {
18             fprintf(stderr, "%s: Не удалось открыть FIFO (%s)\n",
19                     _FILE_, strerror(errno));
20             exit(-1);
21         }
22         msglen = strlen(MESSAGE);
23         if(write(writefd, MESSAGE, msglen) != msglen)
24         {
25             fprintf(stderr, "%s: Не удалось записать в FIFO (%s)\n",
26                     _FILE_, strerror(errno));
27             exit(-2);
28         }
29     }
30 }
```

Figure 3: client2.c

3. common.h (рис. 4)



The screenshot shows the Visual Studio Code editor with the file `common.h` open. The editor has a dark theme and a sidebar on the left showing the file explorer. The top status bar indicates the file is in the `common.h` project. The code in the editor is as follows:

```
1 #ifndef COMMON_H
2 #define COMMON_H
3
4 #include <stdio.h>
5 #include <stdlib.h>
6 #include <string.h>
7 #include <sys/types.h>
8 #include <sys/stat.h>
9 #include <fcntl.h>
10
11 #define PIPE_NAME "/tmp/pipe"
12 #define MAX_BUFF 80
13
14 #endif /* COMMON_H */
```

Figure 4: common.h

4. makefile (рис. 5)

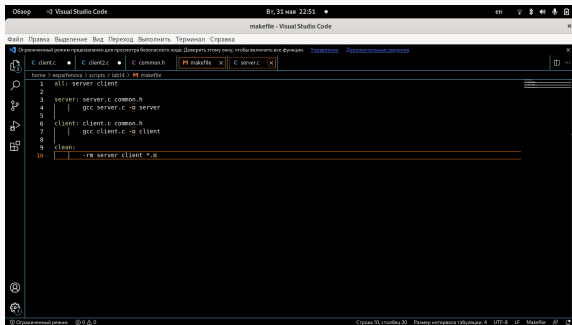


Figure 5: makefile

5. server.c (рис. 6)

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int
5 main()
6 {
7     int readfd;
8     int n;
9     char buff[MAX_BUFFER];
10    printf("FIFO Server...\n");
11
12    if (mkfifo(FIFO_NAME, S_IFIFO | 0660, 0) < 0)
13    {
14        fprintf(stderr, "Ошибка создания FIFO (%s)\n",
15                FILE_, strerror(errno));
16        exit(-1);
17    }
18
19    if (readfd = open(FIFO_NAME, O_RDONLY) < 0)
20    {
21        fprintf(stderr, "Ошибка открытия FIFO (%s)\n",
22                FILE_, strerror(errno));
23        exit(-2);
24    }
25
26    clock_t nowTime(NULL), startTime(NULL);
27    while (now start=0)
28    {
29        while (n = read(readfd, buff, MAX_BUFFER) > 0)
30        {
31            if (write(1, buff, n) != n)
32            {
33                fprintf(stderr, "Ошибка записи (%s)\n",

```

Figure 6: server.c

Следующим шагом открываем папку, в которой хранятся все файлы, в терминале и запускаем make одноименной командой. (рис. 7)

```
[eeeparfenova@fedora lab14]$ make  
gcc server.c -o server  
server.c: В функции «main»:
```

Figure 7: Запуск make

Мы приобрели практические навыки работы с именованными каналами.