

Лабораторная работы №14 “Именованные каналы”

Мажитов М.А.

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?

1. Создал файлы *common.h*, *server.c*, *client.c*, *client2.c*. Скопировал основной код из теоретической части лабораторной работы и немного подкорректировал его.(рис. 1, 2, 3, 4)

```
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
12 #define FIFO_NAME "/tmp/fifo"
13 #define MAX_BUFF 80
14
15 #endif /* __COMMON_H__ */
16
```

Figure 1: Файл common.h

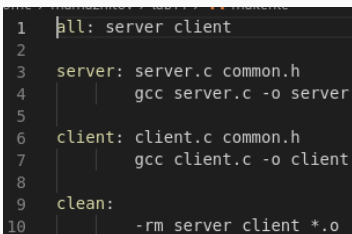
Ход работы

```
1 #include "common.h"
2 int
3 main()
4 {
5     int readfd;
6     int n;
7     char buff[MAX_BUFF];
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",
13            __FILE__, strerror(errno));
14        exit(-1);
15    }
16
17    if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
18    {
19        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
20            __FILE__, strerror(errno));
21        exit(-2);
22    }
23    clock_t now=time(NULL), start=time(NULL);
24    while(now-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",
31                    __FILE__, strerror(errno));
32            }
33        }
34        now=time(NULL);
35    }
36    printf("server timeout, %li - seconds passed\n", (now-start));
37    close(readfd);
38
39    if(unlink(FIFO_NAME) < 0)
40    {
41        fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
42            __FILE__, strerror(errno));
43        exit(-4);
44    }
45    exit(0);
46 }
```

```
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
11     for(i=0; i<20; i++)
12     {
13         sleep(3);
14         t=time(NULL);
15         printf("FIFO Client...\n");
16
17         if((msg = open(FIFO_NAME, O_WRONLY)) < 0)
18         {
19             fprintf(stderr,"%s: Невозможно открыть FIFO (%s)\n",
20                 __FILE__, strerror(errno));
21             exit(-1);
22         }
23
24         len = strlen(MESSAGE);
25
26         if(write(msg, MESSAGE, len) != len)
27         {
28             fprintf(stderr,"%s: Ошибка записи в FIFO (%s)\n",
29                 __FILE__, strerror(errno));
30             exit(-2);
31         }
32         close(msg);
33     }
34     exit(0);
35 }
```

```
1  #include "common.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
12 for(count=0; count<=5; ++count)
13 {
14     sleep(5);
15     t=(long long int) time(0);
16     sprintf(message, "%lli", t);
17     if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
18     {
19         fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
20             __FILE__, strerror(errno));
21         exit(-1);
22     }
23
24     msglen = strlen(MESSAGE);
25     if(write(writefd, MESSAGE, msglen) != msglen)
26     {
27         fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
28             __FILE__, strerror(errno));
29         exit(-2);
30     }
31
32 }
33
34 close(writefd);
35 exit(0);
36 }
```

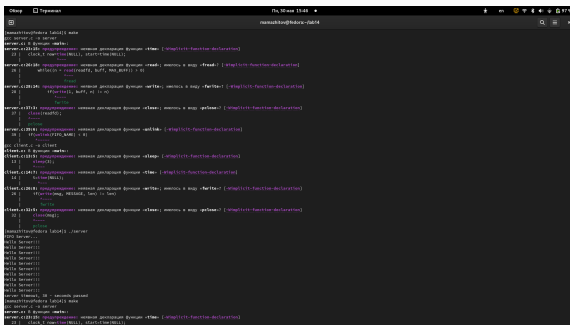
2. Создал *makefile*. (рис. 5)



```
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
9 clean:
10     -rm server client *.o
```

Figure 5: makefile

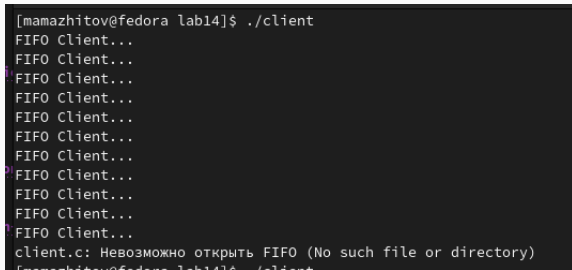
3. Запустил *makefile*. Затем запустил *server*. (рис. 6)



```
1  struct Server {int id; char *name;};
2  void start_server(int id, char *name) {
3      printf("Server ID: %d, Name: %s\n", id, name);
4  }
5  int main() {
6      struct Server s;
7      s.id = 1;
8      s.name = "Server";
9      start_server(s.id, s.name);
10     return 0;
11 }
```

Figure 6: Запуск *makefile* и *server*

4. Запустил client в отдельном окне терминала (рис. 7)



```
[mamazhitov@fedora lab14]$ ./client
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
client.c: Невозможно открыть FIFO (No such file or directory)
[mamazhitov@fedora lab14]$
```

Figure 7: Запуск client

Мы научились пользоваться именованными каналами.