

Лабораторная работа No 14. Именованные каналы

Leysan R. Abdullina

NEC-2022, 31 May

RUDN University, Moscow, Russian Federation

Лабораторная работа No 14.

Именованные каналы

Приобретение практических навыков работы с именованными каналами.

Для передачи данных между неродственными процессами можно использовать механизм именованных каналов (named pipes). Данные передаются по принципу FIFO (First In First Out) (первым записан—первым прочитан), поэтому они называются также FIFO pipes или просто FIFO. Именованные каналы отличаются от неименованных наличием идентификатора канала, который представлен как специальный файл (соответственно имя именованного канала—это имя файла). Поскольку файл находится на локальной файловой системе, данное IPC используется внутри одной системы.

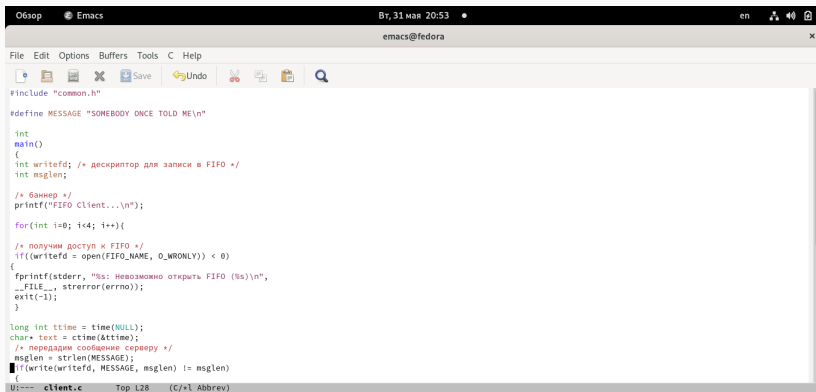
Создадим подкаталог `prog1` через команду `mkdir` и нужные нам файлы `"common.h"`, `"server.c"`, `"client.c"` и `"Makefile"` через команду `touch`

Изучим приведённые в тексте программы `server.c` и `client.c`. Взяв данные примеры за образец, напомним аналогичные программы, внеся следующие изменения:

1. Работает не 1 клиент, а несколько (например, два).
2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Используем функцию `sleep()` для приостановки работы клиента.
3. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используем функцию `clock()` для определения времени работы сервера.

Выполнение работы

- В файл `client.c` добавим 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. (скриншот 1)



```
#include "common.h"

#define MESSAGE "SOMEBODY ONCE TOLD ME\n"

int
main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;

    /* Баннер */
    printf("FIFO Client...\n");

    for(int i=0; i<4; i++){

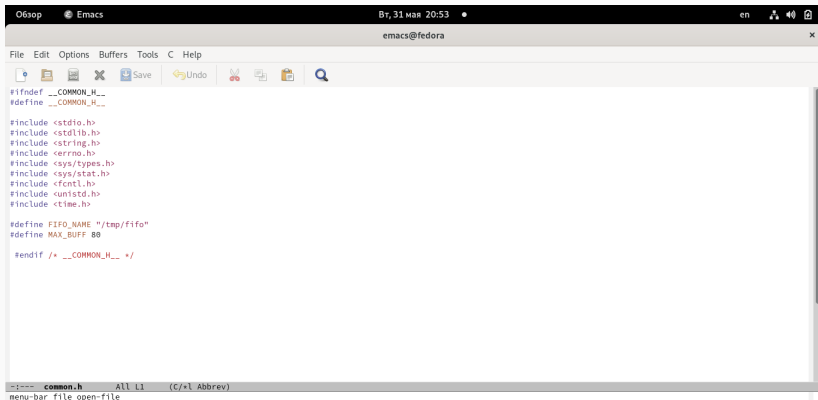
        /* получим доступ к FIFO */
        if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-1);
        }

        long int ttime = time(NULL);
        char* text = ctime(&ttime);
        /* передадим сообщение серверу */
        msglen = strlen(MESSAGE);
        if(write(writefd, MESSAGE, msglen) != msglen)
        {
            U:--- client.c      Top L28      (C/*l Abbrev)
```

Figure 1: Измененный код `client.c`

Выполнение работы

- В файл `common.h` добавим 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 (скриншот 2)



```
#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>

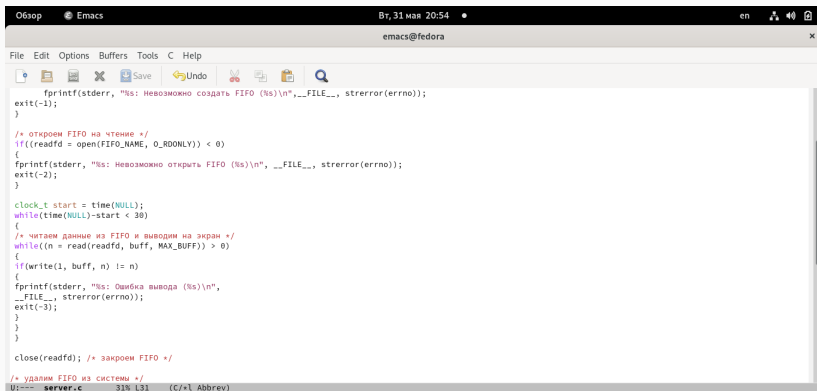
#define FIFO_NAME "/tmp/fifo"
#define MAX_BUFF 80

#endif /* __COMMON_H__ */
```

--- common.h All L1 (C/*1 Abbrev)
menu-bar file open-file

Выполнение работы

- В файл server.c добавим 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. (скриншот 3)



```
Обзор Emacs Бт, 31 мая 20:54 en emacs@fedora
File Edit Options Buffers Tools C Help

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);
}

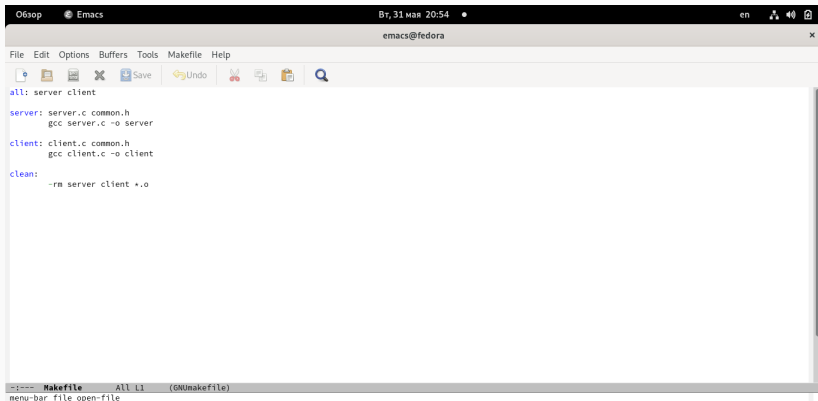
clock_t start = time(NULL);
while(time(NULL)-start < 30)
{
    /* читаем данные из FIFO и выводим на экран */
    while((n = read(readfd, buff, MAX_BUFF)) > 0)
    {
        if(write(1, buff, n) != n)
        {
            fprintf(stderr, "%s: Ошибка вывода (%s)\n",
                __FILE__, strerror(errno));
            exit(-3);
        }
    }
}

close(readfd); /* закроем FIFO */

/* удалим FIFO из системы */
U:--- server.c 31% L31 (C/*l Abbrev)
```

Figure 3: Измененный код server.c

- Makefile (файл для сборки) не изменяем (скриншот 4)



```
all: server client

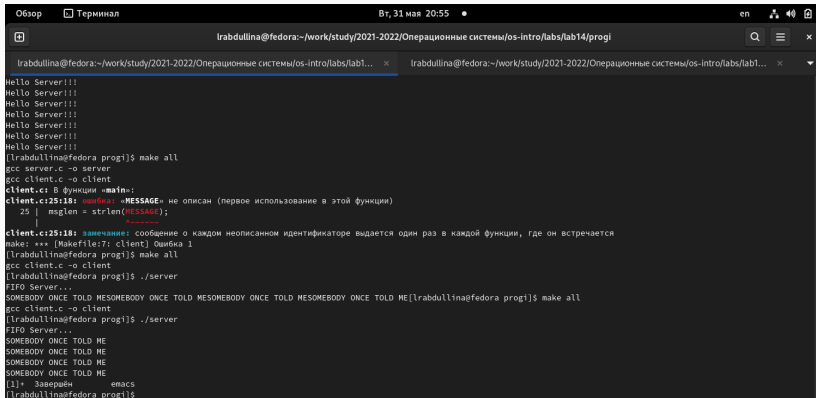
server: server.c common.h
    gcc server.c -o server

client: client.c common.h
    gcc client.c -o client

clean:
    rm server client *.o
```

Figure 4: Код Makefile

Далее мы компилировали наши файлы через команду `make all` (скриншот 5)



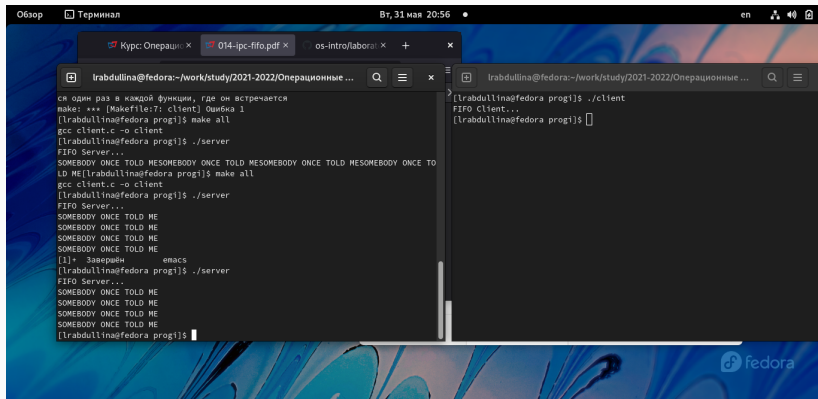
```
Обзор Терминал Вт, 31 мая 20:55 en
lrabduullina@fedora:~/work/study/2021-2022/Операционные системы/os-intro/labs/lab14/progi
lrabduullina@fedora:~/work/study/2021-2022/Операционные системы/os-intro/labs/lab14/progi
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
[lrabduullina@fedora progi]$ make all
gcc server.c -o server
gcc client.c -o client
client.c: В функции «main»:
client.c:25:18: ошибка: «MESSAGE» не описан (первое использование в этой функции)
   25 |     msglen = strlen(MESSAGE);
      |                      ^~~~~~
client.c:25:18: замечание: сообщение о каждом неопisanном идентификаторе выдается один раз в каждой функции, где он встречается
make: *** [Makefile:7: client] Ошибка 1
[lrabduullina@fedora progi]$ make all
gcc client.c -o client
[lrabduullina@fedora progi]$ ./server
FIFO Server...
SOMEBODY ONCE TOLD MESOMEBODY ONCE TOLD MESOMEBODY ONCE TOLD MESOMEBODY ONCE TOLD ME[lrabduullina@fedora progi]$ make all
gcc client.c -o client
[lrabduullina@fedora progi]$ ./server
FIFO Server...
SOMEBODY ONCE TOLD ME
SOMEBODY ONCE TOLD ME
SOMEBODY ONCE TOLD ME
SOMEBODY ONCE TOLD ME
[1]+  Завершён      emacs
[lrabduullina@fedora progi]$
```

Figure 5: Компиляция

Выполнение работы

После чего открываем 2 терминала и проверяем работу наших программ. Вводим в первый терминал `./server`, во второй - `./client`

Программа останавливается после 30 секунд, следовательно работают успешно.(скриншот 6)



```
lrabdu...@fedora:~/work/study/2021-2022/Операционные ...  
с я один раз в каждой функции, где он встречается  
make: *** [Makefile:7: client] Ошибка 1  
[lrabdu...@fedora prog]$ make all  
gcc client.c -o client  
[lrabdu...@fedora prog]$ ./server  
FIFO Server...  
SOMEbody ONCE TOLD MESOMEbody ONCE TOLD MESOMEbody ONCE TOLD MESOMEbody ONCE TO  
LD ME [lrabdu...@fedora prog]$ make all  
gcc client.c -o client  
[lrabdu...@fedora prog]$ ./server  
FIFO Server...  
SOMEbody ONCE TOLD ME  
SOMEbody ONCE TOLD ME  
SOMEbody ONCE TOLD ME  
SOMEbody ONCE TOLD ME  
SOMEbody ONCE TOLD ME  
[!]+ Завершен emacs  
[lrabdu...@fedora prog]$ ./server  
FIFO Server...  
SOMEbody ONCE TOLD ME  
SOMEbody ONCE TOLD ME  
SOMEbody ONCE TOLD ME  
SOMEbody ONCE TOLD ME  
[lrabdu...@fedora prog]$  
  
lrabdu...@fedora:~/work/study/2021-2022/Операционные ...  
[lrabdu...@fedora prog]$ ./client  
FIFO client...  
[lrabdu...@fedora prog]$
```

В ходе лабораторной работы мы приобрели практические навыки работы с именованными каналами