

Отчёт по лабораторной работе №14

Именованные каналы

Надежда Александровна Рогожина

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	15
	Список литературы	16

Список иллюстраций

4.1	client.c	10
4.2	client2.c	11
4.3	server.c	12
4.4	common.h	12
4.5	Makefile	13
4.6	make	13
4.7	Запуск сервера с клиентами	14

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.

3 Теоретическое введение

Одним из видов взаимодействия между процессами в операционных системах является обмен сообщениями. Под сообщением понимается последовательность байтов, передаваемая от одного процесса другому. В операционных системах типа UNIX есть 3 вида межпроцессорных взаимодействий:

- общепонимание (именованные каналы, сигналы),
- System V Interface Definition (SVID — разделяемая память, очередь сообщений, семафоры) и
- BSD (сокеты)

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.

После создания файла канала процессы, участвующие в обмене данными, должны открыть этот файл либо для записи, либо для чтения. При закрытии файла сам канал продолжает существовать. Для того чтобы закрыть сам канал, нужно удалить его файл, например с помощью вызова `unlink(2)`.

Каналы представляют собой простое и удобное средство передачи данных, которое, однако, подходит не во всех ситуациях. Например, с помощью кана-

лов довольно трудно организовать обмен асинхронными сообщениями между процессами.

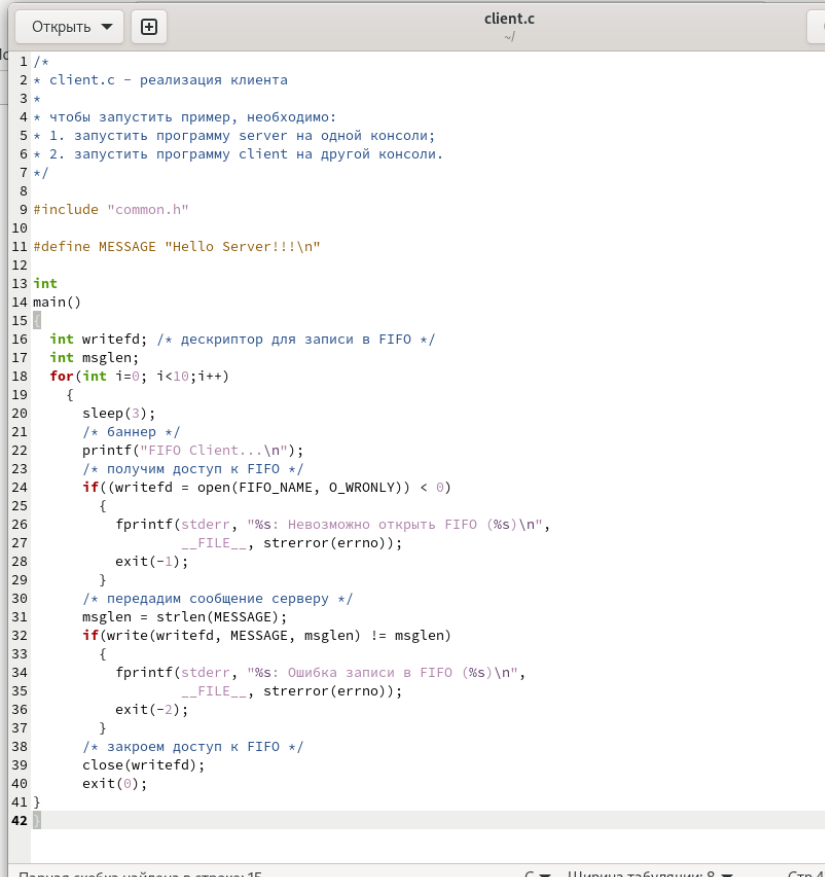
4 Выполнение лабораторной работы

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.

Первым делом составим нужные нам файлы:

- `client.c` для первого клиента (рис. 4.1).



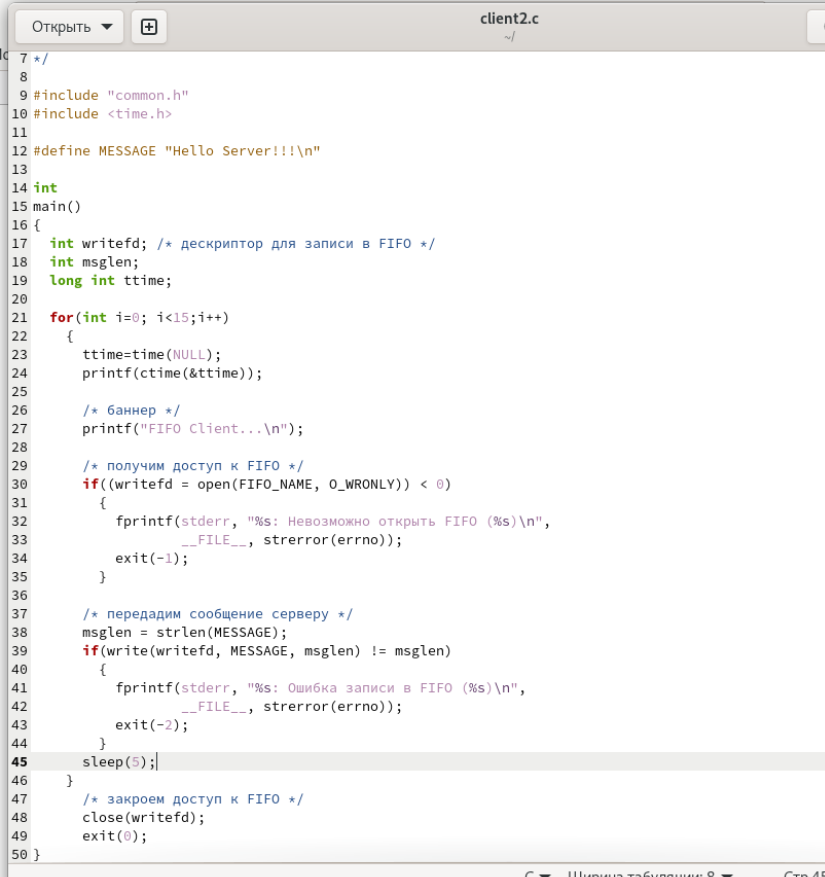
```
1 /*
2 * client.c - реализация клиента
3 *
4 * чтобы запустить пример, необходимо:
5 * 1. запустить программу server на одной консоли;
6 * 2. запустить программу client на другой консоли.
7 */
8
9 #include "common.h"
10 #define MESSAGE "Hello Server!!!\n"
11
12 int
13 main()
14 {
15     int writefd; /* дескриптор для записи в FIFO */
16     int msglen;
17     for(int i=0; i<10;i++)
18     {
19         sleep(3);
20         /* баннер */
21         printf("FIFO Client...\n");
22         /* получим доступ к FIFO */
23         if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
24         {
25             fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
26                     __FILE__, strerror(errno));
27             exit(-1);
28         }
29         /* передадим сообщение серверу */
30         msglen = strlen(MESSAGE);
31         if(write(writefd, MESSAGE, msglen) != msglen)
32         {
33             fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
34                     __FILE__, strerror(errno));
35             exit(-2);
36         }
37         /* закроем доступ к FIFO */
38         close(writefd);
39         exit(0);
40     }
41 }
42
```

Парная скобка найдена в строке: 15

С Ширина табуляции: 8 Стр 42

Рис. 4.1: client.c

- client2.c для второго клиента (рис. 4.2).



```
7 */
8
9 #include "common.h"
10 #include <time.h>
11
12 #define MESSAGE "Hello Server!!!\n"
13
14 int
15 main()
16 {
17     int writefd; /* дескриптор для записи в FIFO */
18     int msglen;
19     long int ttime;
20
21     for(int i=0; i<15;i++)
22     {
23         ttime=time(NULL);
24         printf(ctime(&ttime));
25
26         /* баннер */
27         printf("FIFO Client...\n");
28
29         /* получим доступ к FIFO */
30         if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
31         {
32             fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
33                     __FILE__, strerror(errno));
34             exit(-1);
35         }
36
37         /* передадим сообщение серверу */
38         msglen = strlen(MESSAGE);
39         if(write(writefd, MESSAGE, msglen) != msglen)
40         {
41             fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
42                     __FILE__, strerror(errno));
43             exit(-2);
44         }
45         sleep(5);
46     }
47     /* закроем доступ к FIFO */
48     close(writefd);
49     exit(0);
50 }
```

С ▾ Ширина табуляции: 8 ▾ Стр 45,

Рис. 4.2: client2.c

- Файл самого сервера (рис. 4.3).

```

1  char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */
2
3  /* баннер */
4  printf("FIFO Server...\n");
5
6  /* создаем файл FIFO с открытыми для всех
7   * правами доступа на чтение и запись
8   */
9  if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0){
10     fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
11             __FILE__, strerror(errno));
12     exit(-1);
13 }
14
15 /* откроем FIFO на чтение */
16 if((readfd = open(FIFO_NAME, O_RDONLY)) < 0){
17     fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
18             __FILE__, strerror(errno));
19     exit(-2);
20 }
21
22 /* Заведем переменные обозначающие время начала работы сервера и текущее время */
23 clock_t now=time(NULL), start=time(NULL);
24
25 while(now-start<30){
26     /* читаем данные из FIFO и выводим на экран */
27     while((n = read(readfd, buff, MAX_BUFF)) > 0)
28     {
29         if(write(1, buff, n) != n){
30             fprintf(stderr, "%s: Ошибка вывода (%s)\n",
31                     __FILE__, strerror(errno));
32             exit(-3);
33         }
34     }
35     now=time(NULL);
36 }
37 printf("Время работы сервера вышло, %li - сек. прошло\n", (now-start));
38 close(readfd); /* закроем FIFO */
39
40 /* удалим FIFO из системы */
41 if(unlink(FIFO_NAME) < 0)
42 {
43     fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
44             __FILE__, strerror(errno));
45     exit(-4);
46 }
47 exit(0);
48 }

```

Рис. 4.3: server.c

- Заголовочный файл со стандартными определениями (рис. 4.4).

```

1  /*
2   * common.h - заголовочный файл со стандартными определениями
3   */
4
5  #ifndef __COMMON_H__
6  #define __COMMON_H__
7
8  #include <stdio.h>
9  #include <stdlib.h>
10 #include <string.h>
11 #include <errno.h>
12 #include <sys/types.h>
13 #include <sys/stat.h>
14 #include <fcntl.h>
15 #include <time.h>
16
17 #define FIFO_NAME "/tmp/fifo"
18 #define MAX_BUFF 80
19
20 #endif /* __COMMON_H__ */

```

Рис. 4.4: common.h

- Makefile (рис. 4.5).

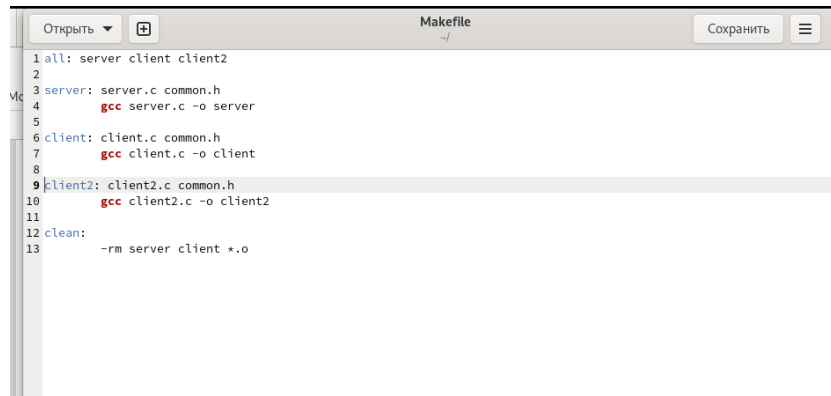


Рис. 4.5: Makefile

После создания всех файлов запустим команду make (рис. 4.6).



Рис. 4.6: make

Команда выполнена успешно, и затем я открыла 3 терминала и запустила сервер и двух клиентов (рис. 4.7).

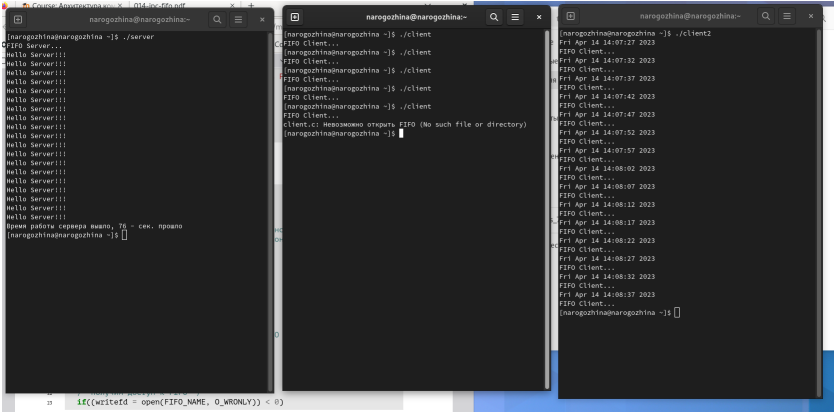


Рис. 4.7: Запуск сервера с клиентами

Через некоторое время (-76 секунд) сервер завершил работу, и в тот же момент, я попыталась запустить первого клиента - именно из-за этого выдало “ошибку подключения”.

5 Выводы

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

Список литературы

1. Руководство по выполнению лабораторной работы №14