Лабораторная работа №13

Дисциплина - операционные системы

Волгин Иван Алексеевич

Содержание

Список иллюстраций

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, 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). Используйте функцию sleep() для приостановки работы клиента. 3. Сервер работает не бесконечно, а прекращает работу через некоторое время (напри- мер, 30 сек). Используйте функцию clock() для определения времени работы сервера. Что будет в случае, если сервер завершит работу, не закрыв канал?

3 Выполнение лабораторной работы

Сперва я создал нужные файлы (рис. 3.1).

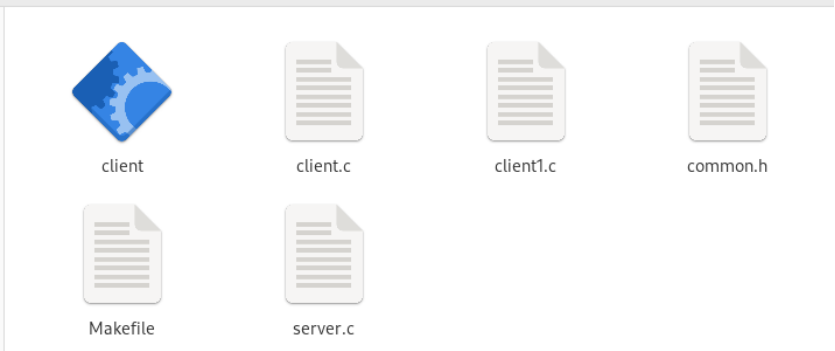


Рис. 3.1: Созданные файлы

Затем приступил к выполнению работы.

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. (рис. 3.2) (рис. 3.3)
3. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию clock() для определения времени работы сервера. (рис. 3.4)

Так же был создан Makefile (рис. 3.5)

Что будет в случае, если сервер завершит работу, не закрыв канал?

При компиляции и выполнении файлов возникли ошибки, которые я так и не смог решить.

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

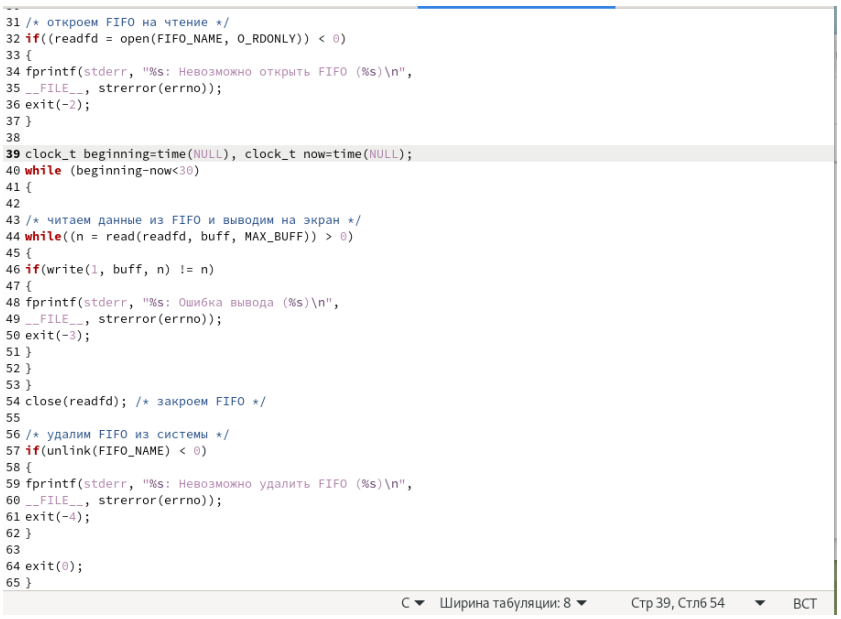
С ▾ Ширина табуляции: 8 ▾ Стр 46, Стлб 15 ▾ ВСТ

Рис. 3.2: Код программы

```
 1 #include "common.h"
 2 #include <time.h>
 3 #define MESSAGE "Hello Server!!!\n"
 4
 5 int
 6 main()
 7 {
 8 int writefd; /* дескриптор для записи в FIFO */
 9 int msglen;
10 long int ttime;
11
12 for(int i=0; i<15; i++)
13 {
14 ttime=time(NULL)
15 printf(ctime(&ttime));
16 /* баннер */
17 printf("FIFO Client...\n");
18
19 /* получим доступ к FIFO */
20 if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
21 {
22 fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
23 __FILE__, strerror(errno));
24 exit(-1);
25 }
26
27 /* передадим сообщение серверу */
28 msglen = strlen(MESSAGE);
29 if(write(writefd, MESSAGE, msglen) != msglen)
30 {
31 fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
32 __FILE__, strerror(errno));
33 exit(-2);
34 }
35 sleep(4);
```

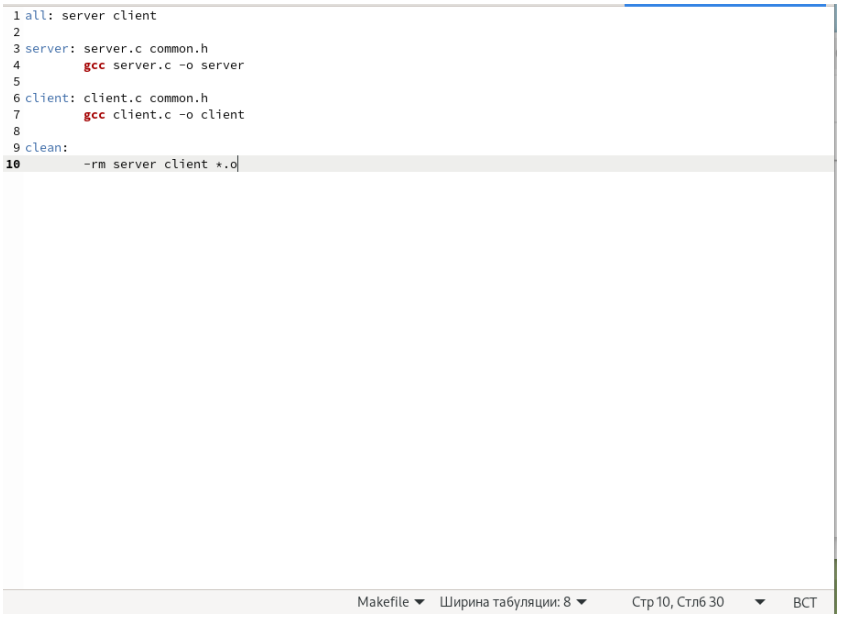
С ▾ Ширина табуляции: 8 ▾ Стр 1, Стлб 1 ▾ ВСТ

Рис. 3.3: Код программы



```
/* 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 beginning=time(NULL), clock_t now=time(NULL);
while (beginning-now<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 */
if(unlink(FIFO_NAME) < 0)
{
fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
__FILE__, strerror(errno));
exit(-4);
}

exit(0);
}
```


Рис. 3.4: Код программы



```
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
```


Рис. 3.5: Makefile

4 Выводы

В процессе выполнения этой лабораторной работы я приобрел практические навыки работы с именованными каналами.