

Лабораторная работа №14

Именованные каналы

Салькова кристина Михайловна

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 4 | Выводы | 10 |
| | Список литературы | 11 |

Список иллюстраций

| | | |
|-----|------------------------------|---|
| 3.1 | создание файлов | 7 |
| 3.2 | изменение server.c | 8 |
| 3.3 | изменение client.c | 8 |
| 3.4 | Makefile | 9 |
| 3.5 | команда make all | 9 |
| 3.6 | проверка | 9 |

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

3 Выполнение лабораторной работы

1. Создадим необходимые файлы с помощью команды «touch common.h server.c client.c Makefile» и отредактируем их

```
kmsaljkova@dk5n56 ~/work/lab14 $ touch common.h server.c client.c
kmsaljkova@dk5n56 ~/work/lab14 $ Makefile
bash: Makefile: команда не найдена
kmsaljkova@dk5n56 ~/work/lab14 $ touch Makefile
kmsaljkova@dk5n56 ~/work/lab14 $ ls
client.c  common.h  Makefile  server.c
```

Рис. 3.1: создание файлов

2. Далее меняем коды программ, представленных в тексте лабораторной 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



```
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 #include <unistd.h>
12 #include <time.h>
13
14 #define FIFO_NAME "/tmp/fifo"
15 #define MAX_BUFF 80
16
17 #endif
```

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(рис

```

1 #include <common.h>
2 int main()
3 {
4     int readfd;
5     int n;
6     char buf[FIFO_SIZE];
7     printf("FIFO Server...\n");
8     if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
9     {
10         fprintf(stderr, "Невозможно открыть FIFO (%s)\n", __FILE__, strerror(errno));
11         exit(-1);
12     }
13     if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
14     {
15         fprintf(stderr, "Невозможно открыть FIFO (%s)\n", __FILE__, strerror(errno));
16         exit(-1);
17     }
18     clock_t start = time(NULL);
19     while((time(NULL) - start) < 10)
20     {
21         while((n = read(readfd, buf, FIFO_SIZE)) > 0)
22         {
23             if(write(writefd, buf, n) != n)
24             {
25                 fprintf(stderr, "Ошибка записи (%s)\n",
26                     __FILE__, strerror(errno));
27                 exit(-1);
28             }
29         }
30         close(readfd); // закрываем FIFO
31         if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
32         {
33             fprintf(stderr, "Невозможно открыть FIFO (%s)\n",
34                 __FILE__, strerror(errno));
35             exit(-1);
36         }
37     }
38     exit(0);
39 }

```

Рис. 3.2: изменение server.c

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.

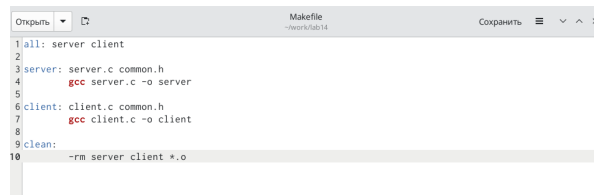
```

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

```

Рис. 3.3: изменение client.c

Makefile (файл для сборки) не изменяем



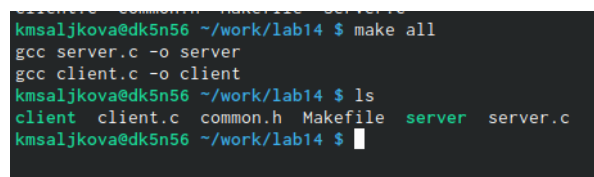
```

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

```

Рис. 3.4: Makefile

3. После написания кодов,используя команду «make all», скомпилируем необходимые файлы



```

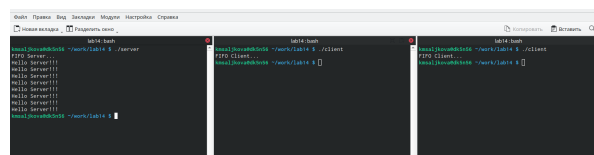
kmsaljkova@dk5n56 ~/work/lab14 $ make all
gcc server.c -o server
gcc client.c -o client
kmsaljkova@dk5n56 ~/work/lab14 $ ls
client  client.c  common.h  Makefile  server  server.c
kmsaljkova@dk5n56 ~/work/lab14 $

```

Рис. 3.5: команда make all

4. Далее проверим работу написанного кода.

Откроем 3 консоли (терминала) и запустил: в первом терминале – «./server», в остальных двух – «./client». В результате каждый терминал-клиент вывел по 4 сообщения. Спустя 30 секунд работа сервера была прекращена. Программа работает корректно



The image shows three terminal windows. The leftmost window is running the server program, which outputs 'PPTD Server' four times. The middle and right windows are running the client program, which outputs 'PPTD Client' four times each. The server window shows a prompt to press Ctrl+C to stop the server.

Рис. 3.6: проверка

4 Выводы

Мы приобрели практические навыки работы с именованными каналами.

Список литературы