

Российский университет дружбы народов

Факультет физико-математический и естественных наук

Кафедра прикладной информатики и теории вероятностей

Лабораторная работа №14 “Именованные каналы”

Дисциплина

Операционные системы

Студент

Олейник Анастасия Игоревна

Группа

НБИбд-03-21

Отчёт

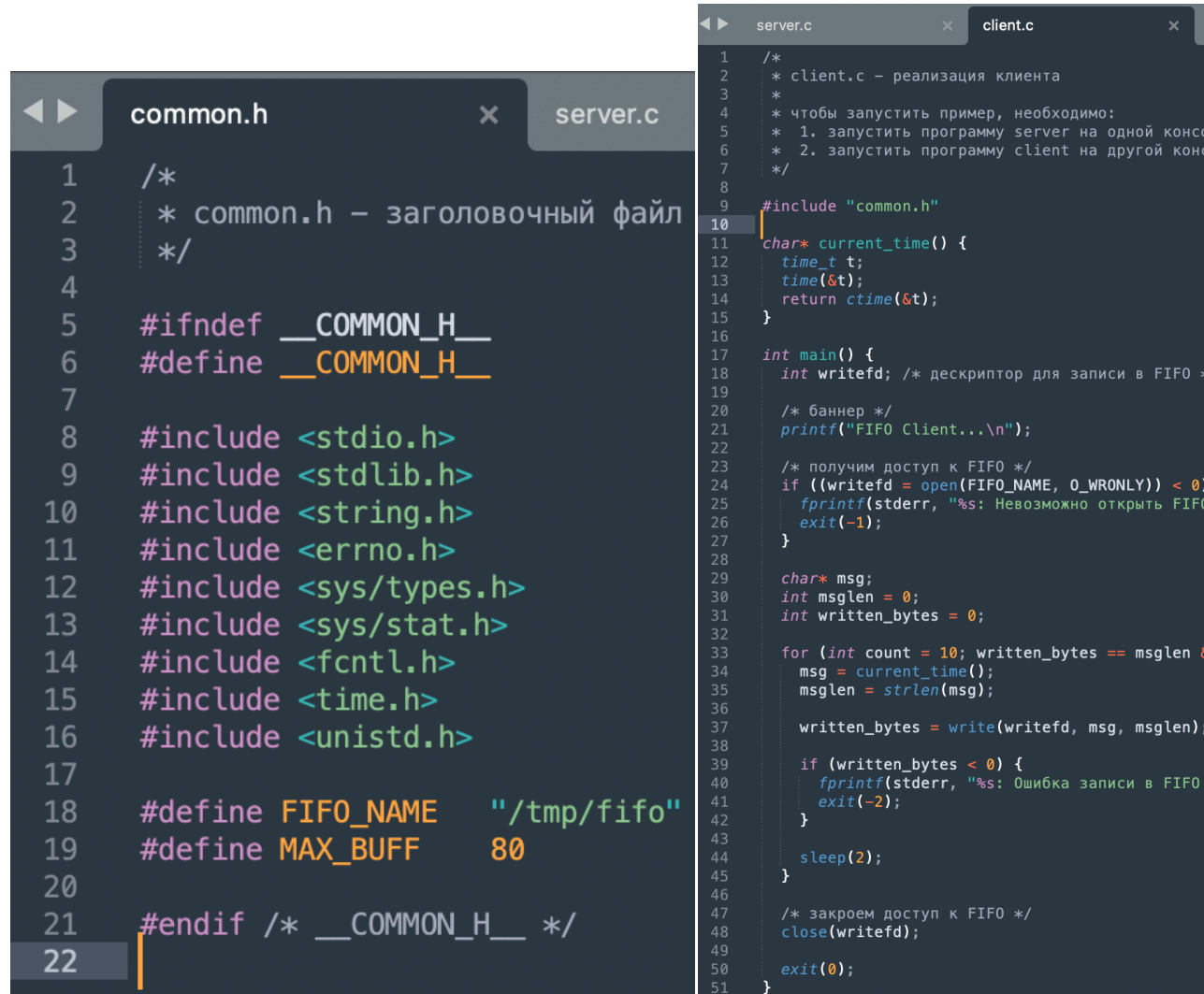
1. Работает не 1 клиент, а несколько:

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

Makefile
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
11
```

2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять

секунд). Используя функцию sleep() для приостановки работы клиента:



```
common.h
1  /*
2   * common.h - заголовочный файл
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 #include <unistd.h>
17
18 #define FIFO_NAME    "/tmp/fifo"
19 #define MAX_BUFF     80
20
21 #endif /* __COMMON_H__ */
22

server.c
1  /*
2   * server.c - реализация сервера
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
11 char* current_time() {
12     time_t t;
13     time(&t);
14     return ctime(&t);
15 }
16
17 int main() {
18     int writefd; /* дескриптор для записи в FIFO */
19
20     /* баннер */
21     printf("FIFO Client...\n");
22
23     /* получим доступ к FIFO */
24     if ((writefd = open(FIFO_NAME, O_WRONLY)) < 0) {
25         fprintf(stderr, "%s: Невозможно открыть FIFO\n", FIFO_NAME);
26         exit(-1);
27     }
28
29     char* msg;
30     int msglen = 0;
31     int written_bytes = 0;
32
33     for (int count = 10; written_bytes == msglen; count++) {
34         msg = current_time();
35         msglen = strlen(msg);
36
37         written_bytes = write(writefd, msg, msglen);
38
39         if (written_bytes < 0) {
40             fprintf(stderr, "%s: Ошибка записи в FIFO\n", FIFO_NAME);
41             exit(-2);
42         }
43
44         sleep(2);
45     }
46
47     /* закроем доступ к FIFO */
48     close(writefd);
49
50     exit(0);
51 }
```

```

server.c  client.c  Makefile  common.h
11  int main() {
12      clock_t start;
13      start = clock();
14
15      int readfd; /* дескриптор для чтения из FIFO */
16      int n;
17      char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */
18
19      /* баннер */
20      printf("FIFO Server...\n");
21
22      /*
23       * создаем файл FIFO с открытыми для всех
24       * правами доступа на чтение и запись
25       */
26
27      if (mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0) {
28          fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n", __FILE__, strerror(errno));
29          exit(-1);
30      }
31
32      /* откроем FIFO на чтение */
33      if ((readfd = open(FIFO_NAME, O_RDONLY)) < 0) {
34          fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n", __FILE__, strerror(errno));
35          exit(-2);
36      }
37
38      /* читаем данные из FIFO и выводим на экран */
39      while ((n = read(readfd, buff, MAX_BUFF)) > 0) {
40          if (write(1, buff, n) != n) {
41              fprintf(stderr, "%s: Ошибка вывода (%s)\n", __FILE__, strerror(errno));
42              exit(-3);
43          }
44
45          clock_t current = clock();
46          double duration = (double)(current - start) / 100.0;
47          printf("Reading timestamp: %f\n", duration);
48
49          if (duration > MAX_WORK_TIME) {
50              break;
51          }
52      }
53
54      close(readfd); /* закроем FIFO */
55
56      /* удалим FIFO из системы */
57      if (unlink(FIFO_NAME) < 0) {
58          fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n", __FILE__, strerror(errno));
59          exit(-4);
60      }
61      exit(0);
62  }

```

3. Сервер работает небесконечно, а прекращает работу через некоторое время(например, 30 сек). Используя функцию clock() для определения времени работы сервера.

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

The screenshot displays a code editor with three tabs: `common.h`, `server.c`, and `client.c`. The `common.h` file contains the following code:

```
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 #include <unistd.h>
17
18 #define FIFO_NAME "/tmp/fifo"
19 #define MAX_BUFF 80
20 #define MAX_WORK_TIME 10
21
22 #endif /* __COMMON_H__ */
23
```

Below the editor, there are three terminal windows:

- Terminal 1 (top left):** Shows the execution of `./client` and the output `FIFO Client...`.
- Terminal 2 (bottom left):** Shows the execution of `./client` and the output `FIFO Client...`.
- Terminal 3 (right):** Shows the execution of `./server` and the output `FIFO Server...`. It then displays a series of timestamps and readings, indicating the server is running and processing data.

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

```
[anastasia@Air-Anastasia lab14 % ./server  
FIFO Server...
```

запустить сервер (последний скриншот) `server.c: Невозможно создать FIFO (File exists)`