

Лабораторная работа №14

Операционные системы

Савурская П.А.

26 апреля 2023

Российский университет дружбы народов, Москва, Россия

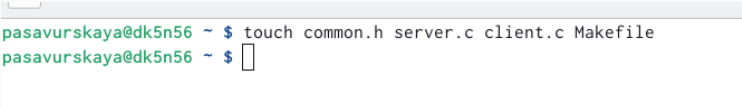
- Савурская Полина
- 1132222827
- НБИбд-04-22
- Российский университет дружбы народов

Приобретение практических навыков работы с именованными каналами.

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?

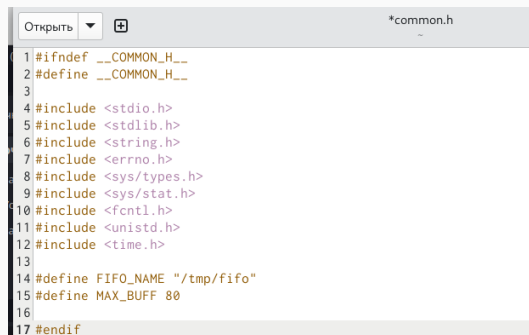
Создаю файлы common.h, server.c, client.c, makefile

A terminal window with a light gray title bar. The prompt is 'pasavurskaya@dk5n56 ~ \$'. The command 'touch common.h server.c client.c Makefile' has been entered and executed. The prompt is now 'pasavurskaya@dk5n56 ~ \$' followed by a cursor.

```
pasavurskaya@dk5n56 ~ $ touch common.h server.c client.c Makefile
pasavurskaya@dk5n56 ~ $
```

Рис. 1: создаю файлы

Изменяю файл common.h

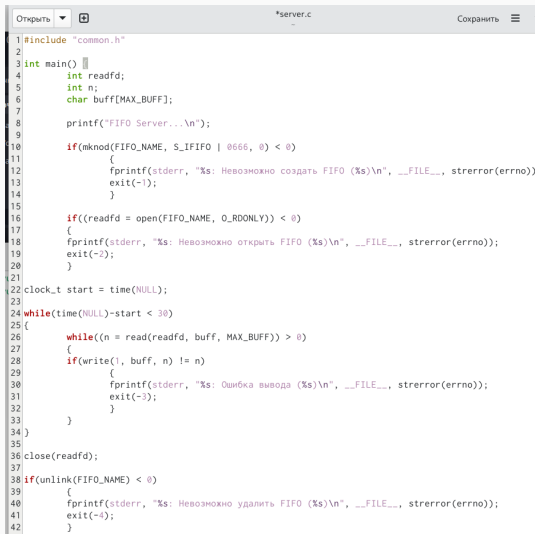
A screenshot of a code editor window. The title bar at the top shows a button labeled 'Открыть' (Open), a dropdown arrow, a plus icon, and the filename '*common.h'. The editor area contains C preprocessor code for a header file. The code includes several standard library headers and defines constants. Line numbers 1 through 17 are visible on the left side of the code.

```
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
```

Рис. 2: common.h

Выполнение лабораторной работы. Шаг 3.

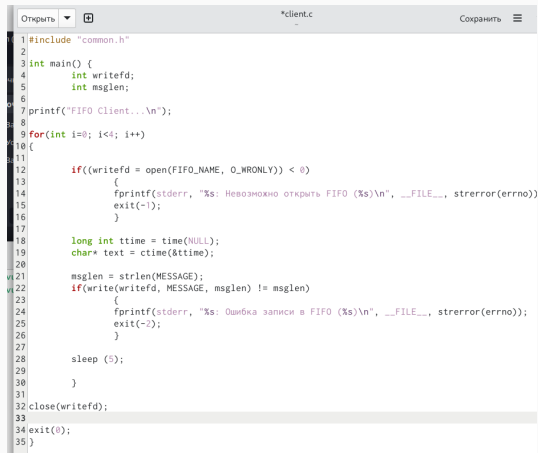
Изменяю файл server.c



```
1#include "common.h"
2
3int main() {
4    int readfd;
5    int n;
6    char buff[MAX_BUFF];
7
8    printf("FIFO Server...\n");
9
10   if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
11   {
12       fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n", __FILE__, strerror(errno));
13       exit(-1);
14   }
15
16   if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
17   {
18       fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n", __FILE__, strerror(errno));
19       exit(-2);
20   }
21
22   clock_t start = time(NULL);
23
24   while(time(NULL)-start < 30)
25   {
26       while((n = read(readfd, buff, MAX_BUFF)) > 0)
27       {
28           if(write(1, buff, n) != n)
29           {
30               fprintf(stderr, "%s: Ошибка вывода (%s)\n", __FILE__, strerror(errno));
31               exit(-3);
32           }
33       }
34   }
35   close(readfd);
36
37   if(unlink(FIFO_NAME) < 0)
38   {
39       fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n", __FILE__, strerror(errno));
40       exit(-4);
41   }
42 }
```

Выполнение лабораторной работы. Шаг 4.

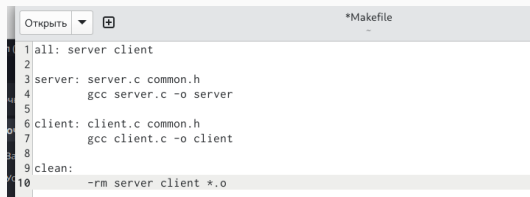
Изменяю файл client.c



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

Рис. 4: client.c

Изменяю файл makefile



The image shows a text editor window with a title bar containing a dropdown menu with the word "Открыть" (Open) and a plus icon, and a file name "*Makefile". The editor contains the following text:

```
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. 5: makefile

В терминале пишем make all.

```
pasavurskaya@dk5n56 ~ $ make all
gcc client.c -o client
client.c: В функции «main»:
client.c:21:25: ошибка: «MESSAGE» не описан (первое использование в этой функции
)
   21 |             msglen = strlen(MESSAGE);
      |                               ^~~~~~
client.c:21:25: замечание: сообщение о каждом неопisanном идентификаторе выдается
я один раз в каждой функции, где он встречается
make: *** [Makefile:7: client] Ошибка 1
pasavurskaya@dk5n56 ~ $
```

Рис. 6: make all

Появляется исполняемый файл client

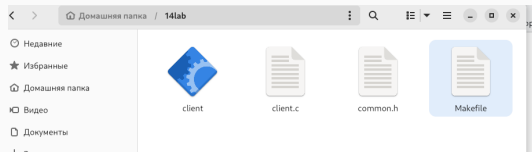
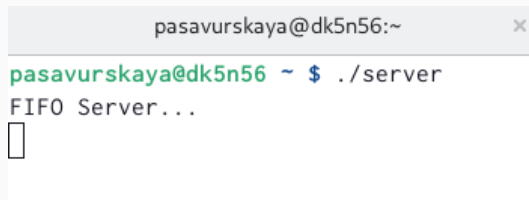


Рис. 7: client

Пишем ./server

A terminal window with a title bar 'pasavurskaya@dk5n56:~' and a close button. The prompt is 'pasavurskaya@dk5n56 ~ \$'. The command './server' has been entered and executed, resulting in the output 'FIFO Server...'. A cursor is visible on the line following the output.

```
pasavurskaya@dk5n56:~  
pasavurskaya@dk5n56 ~ $ ./server  
FIFO Server...  
█
```

Рис. 8: ./server

Я приобрела практические навыки работы с именованными каналами.