

# Лабораторная работа №14

Именованные каналы

---

Салькова К. М.

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Салькова Кристина Михайловна
- студентка НБИбд-04-22
- Российский университет дружбы народов

## Вводная часть

---

- командный процессор ОС UNIX

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?

## Цель работы

---

Приобретение практических навыков работы с именованными каналами.



## Выполнение лабораторной работы

---

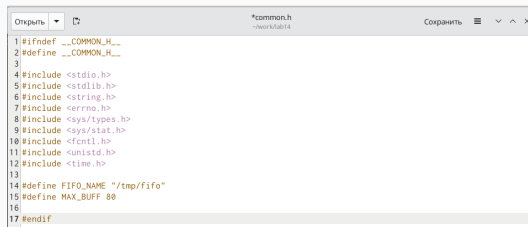
1. Создадим необходимые файлы с помощью команды «touch common.h server.c client.c Makefile» и отредактируем их

```
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. 1: создание файлов

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

A screenshot of a code editor window. The title bar shows the file name as \*common.h and the path as ~work/tab14. The editor contains C preprocessor directives for common.h. The code is as follows:

```
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: изменение common.h

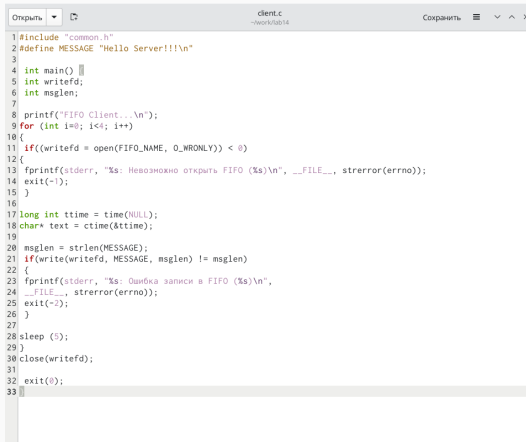
В файл `server.c` добавим цикл `while` для контроля за временем работы сервера. Разница между текущим временем `time(NULL)` и временем начала работы `clock_t start=time(NULL)` (инициализация до цикла) не должна превышать 30 секунд(рис

```
Приложение Меча gedit
Окна: 1
*server.c
1 #include "common.h"
2 int main()
3 {
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, "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, "Невозможно открыть FIFO (%s)\n", _FILE_, strerror(errno));
19        exit(-1);
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, "Ошибка вывода (%s)\n",
31                    _FILE_, strerror(errno));
32                exit(-1);
33            }
34        }
35    }
36    close(readfd); /* закроем FIFO */
37
38    if(unlink(FIFO_NAME) < 0)
39    {
40        fprintf(stderr, "Невозможно удалить FIFO (%s)\n",
41            _FILE_, strerror(errno));
42        exit(-1);
43    }
44
45    exit(0);
46 }
```

Рис. 3: 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     int writefd;
6     int msglen;
7
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, "%s: Невозможно открыть FIFO (%s)\n", __FILE__, strerror(errno));
14     exit(-1);
15 }
16
17     long int ttime = time(NULL);
18     char* text = ctime(&ttime);
19
20     msglen = strlen(MESSAGE);
21     if(write(writefd, MESSAGE, msglen) != msglen)
22 {
23     fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
24     __FILE__, strerror(errno));
25     exit(-2);
26 }
27
28     sleep (5);
29 }
30 close(writefd);
31
32     exit(0);
33 }
```

Рис. 4: изменение client.c

Makefile (файл для сборки) не изменяем

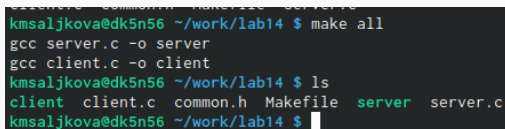


The image shows a code editor window titled "Makefile" with the path "~work/lab14". The editor contains the following text:

```
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

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

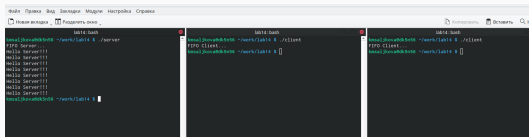


```
client.c common.h Makefile server.c  
kmsaljkova@dk5n56 ~/work/lab14 $ make all  
gcc server.c -o server  
gcc client.c -o client  
kmsaljkova@dk5n56 ~/work/lab14 $ ls  
client client.c common.h Makefile server server.c  
kmsaljkova@dk5n56 ~/work/lab14 $
```

Рис. 6: команда make all

## Задание 4 - 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 screenshot shows three terminal windows side-by-side. The leftmost window is titled 'M012 bash' and shows the command 'M012@kali:~/work/1p014\$ ./server'. It outputs 'INFO Server...' followed by eight lines of 'Hello Server!!!'. The middle window is also titled 'M012 bash' and shows the command 'M012@kali:~/work/1p014\$ ./client'. It outputs 'INFO Client...' followed by a blank line. The rightmost window is also titled 'M012 bash' and shows the command 'M012@kali:~/work/1p014\$ ./client'. It outputs 'INFO Client...' followed by a blank line.

Рис. 7: проверка

## Вывод

---

Мы приобрели практические навыки работы с именованными каналами.