

Именованные каналы

Лабораторная работа №14

Козлова Нонна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	10
	Список литературы	11

Список иллюстраций

2.1	Пользуюсь командой touch	6
2.2	Добавляю стандартные заголовочные файлы	6
2.3	Добавляю цикл while для контроля времени работы	7
2.4	Добавляю цикл, отвечающий за количество сообщений о текущем времени	8
2.5	Использую команду make all	8

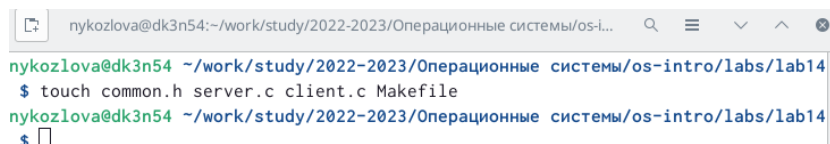
Список таблиц

1 Цель работы

Приобретение практических навыков работы с именованными каналами.

2 Выполнение лабораторной работы

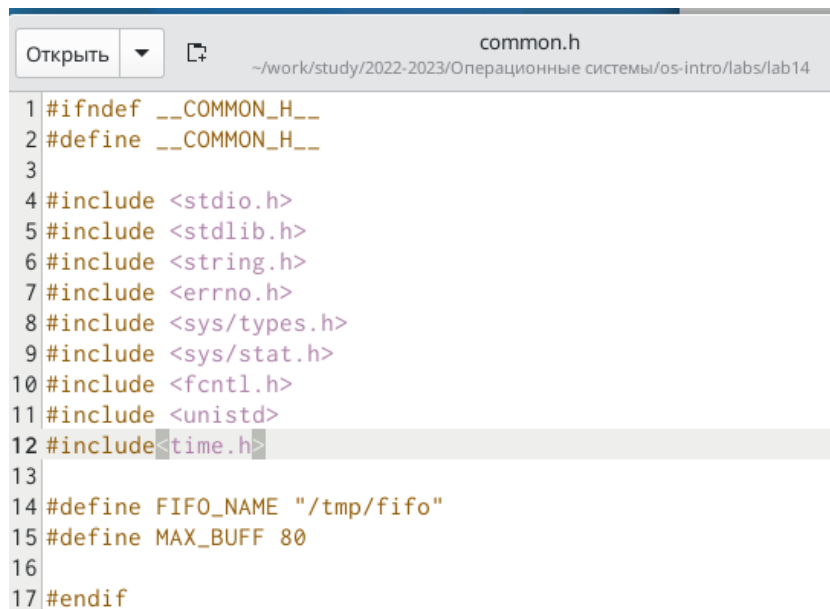
1. Создаю необходимые файлы. (рис. 2.1).



```
nykozlova@dk3n54:~/work/study/2022-2023/Операционные системы/os-i...
nykozlova@dk3n54 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14
$ touch common.h server.c client.c Makefile
nykozlova@dk3n54 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14
$
```

Рис. 2.1: Пользуюсь командой touch

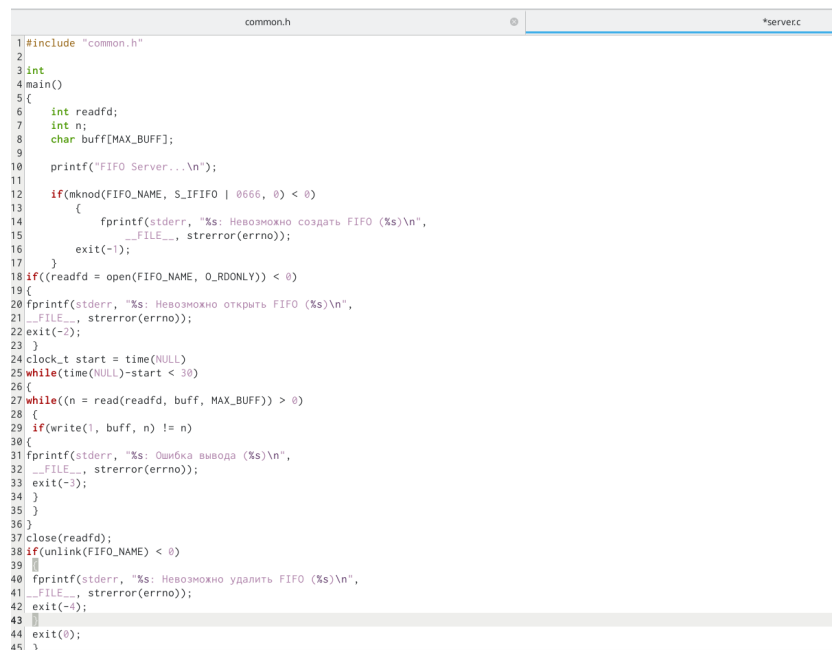
2. Меняю код программы, представленный в лабораторной работе в файле common.h. (рис. 2.2).



```
Открыть common.h
~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14
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>
12 #include <time.h>
13
14 #define FIFO_NAME "/tmp/fifo"
15 #define MAX_BUFF 80
16
17 #endif
```

Рис. 2.2: Добавляю стандартные заголовочные файлы

3. Меняю код программы, представленный в лабораторной работе в файле server.c. (рис. 2.3).



```
1 #include "common.h"
2
3 int
4 main()
5 {
6     int readfd;
7     int n;
8     char buff[MAX_BUFF];
9
10    printf("FIFO Server...\n");
11
12    if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
13    {
14        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
15            __FILE__, strerror(errno));
16        exit(-1);
17    }
18    if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
19    {
20        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
21            __FILE__, strerror(errno));
22        exit(-2);
23    }
24    clock_t start = time(NULL)
25    while(time(NULL)-start < 30)
26    {
27        while((n = read(readfd, buff, MAX_BUFF)) > 0)
28        {
29            if(write(1, buff, n) != n)
30            {
31                fprintf(stderr, "%s: Ошибка вывода (%s)\n",
32                    __FILE__, strerror(errno));
33                exit(-3);
34            }
35        }
36    }
37    close(readfd);
38    if(unlink(FIFO_NAME) < 0)
39    {
40        fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
41            __FILE__, strerror(errno));
42        exit(-4);
43    }
44    exit(0);
45 }
```

Рис. 2.3: Добавляю цикл while для контроля времени работы

4. Меняю код программы, представленный в лабораторной работе в файле client.c. (рис. 2.4).

```

common.h  *server.c  client.c
1 #include "common.h"
2
3 #define MESSAGE "Hello Server!!!\n"
4
5 int
6 main()
7 {
8     int writefd;
9     int msglen;
10
11     printf("FIFO Client...\n");
12
13     for(int i=0; i<4; i++)
14     {
15         if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
16         {
17             printf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
18                __FILE__, strerror(errno));
19             exit(-1);
20         }
21
22         long int ttime = time(NULL);
23         char* text = ctime(&ttime);
24
25         msglen = strlen(MESSAGE);
26         if(write(writefd, MESSAGE, msglen) != msglen)
27         {
28             fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n", __FILE__, strerror(errno));
29             exit(-2);
30         }
31         sleep (5);
32     }
33
34     close(writefd);
35
36     exit(0);
37 }

```

Рис. 2.4: Добавляю цикл, отвечающий за количество сообщений о текущем времени

5. Makefile оставляю неизменным, далее компилирую сразу все файлы(рис. 2.5).

```

nykozlova@dk3n54 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14 $ make all
gcc server.c -o server
gcc client.c -o client
client.c: В функции «main»:
client.c:18:10: предупреждение: в передаче аргумента 1 «printf»: несовместимый тип указателя [-Wincompatible-pointer-types]
18 |     printf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
    |           ^~~~~~
    |           |
    |           FILE *
In file included from common.h:4,
   from client.c:1:
/usr/include/stdio.h:356:43: замечание: ожидался тип «const char * restrict», но аргумент имеет тип «FILE *»
356 | extern int printf (const char *__restrict __format, ...);
    |
nykozlova@dk3n54 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab14 $

```

Рис. 2.5: Использую коанду make all

6. Проверяю работу написанного кода

3 Выводы

В ходе лабораторной работы я приобрела практические навыки работы с именнованными каналами.

Список литературы