

Лабораторная работа №14

Операционные системы

Кирилюк Светлана Алексеевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	10

Список иллюстраций

2.1	Создание файлов для работы	6
2.2	Скрипт common.h	6
2.3	Скрипт sever.c (1)	7
2.4	Скрипт server.c (2)	7
2.5	Скрипт client.c (1)	8
2.6	Скрипт client.c (2)	8
2.7	Скрипт Makefile	9
2.8	Компиляция файлов	9
2.9	Проверка работы файлов	9

Список таблиц

1 Цель работы

Приобретение практических навыков работы с именованными каналами.

2 Выполнение лабораторной работы

Перед началом работы я создала 4 новых файла в рабочем каталоге (рис. 2.1).

```
[sakirilyuk@fedora ~]$ touch Makefile server.c client.c common.h  
[sakirilyuk@fedora ~]$ emacs
```

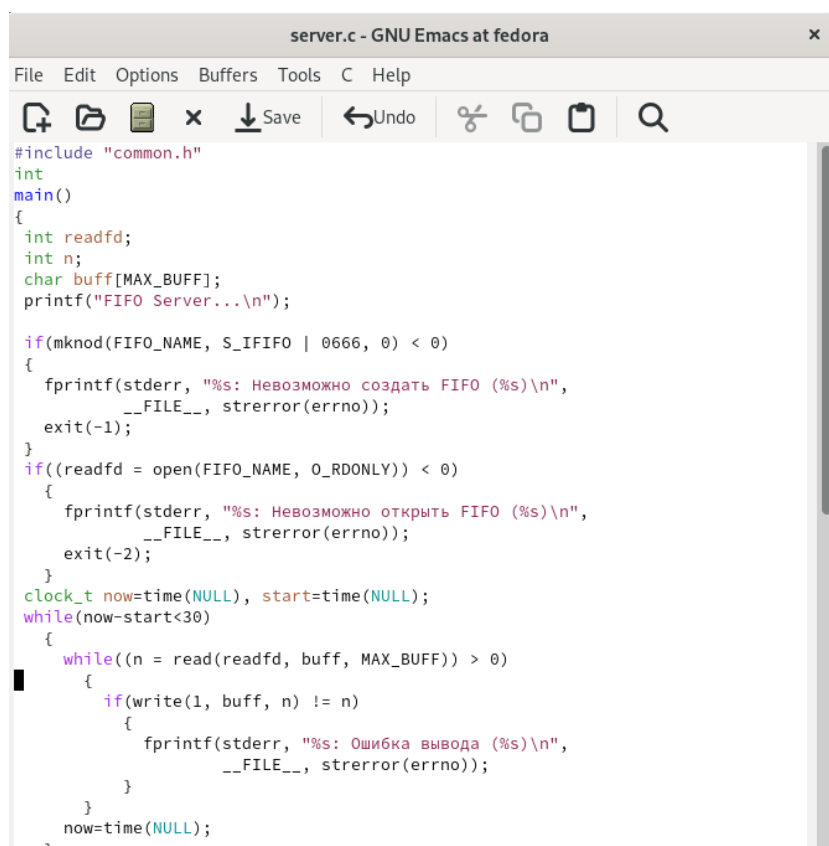
Рис. 2.1: Создание файлов для работы

Затем я перешла в файл `common.h` при помощи `emacs` и написала скрипт для него, исправив листинг (рис. 2.2).

```
#ifndef __COMMON_H__  
#define __COMMON_H__  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <errno.h>  
#include <sys/types.h>  
#include <sys/stat.h>  
#include <fcntl.h>  
#include <unistd.h>  
#include <time.h>  
  
#define FIFO_NAME "/tmp/fifo"  
#define MAX_BUFF 80  
  
#endif /* __COMMON_H__ */
```

Рис. 2.2: Скрипт `common.h`

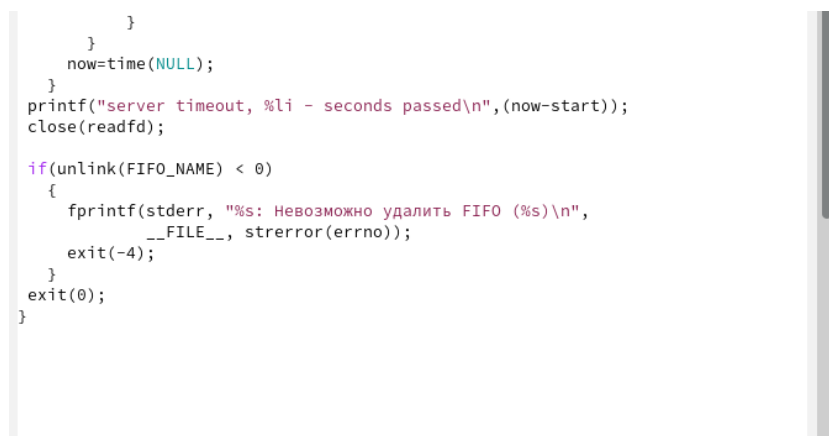
Перейдя в файл server.c, я написала ему скрипт, так же отредактировав листинг (рис. 2.3), (рис. 2.4).



```
server.c - GNU Emacs at fedora
File Edit Options Buffers Tools C Help
[Icons: Open, Save, Undo, Redo, Search]
#include "common.h"
int
main()
{
    int readfd;
    int n;
    char buff[MAX_BUFF];
    printf("FIFO Server...\n");

    if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
    {
        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
            __FILE__, strerror(errno));
        exit(-1);
    }
    if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
    {
        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
            __FILE__, strerror(errno));
        exit(-2);
    }
    clock_t now=time(NULL), start=time(NULL);
    while(now-start<30)
    {
        while((n = read(readfd, buff, MAX_BUFF)) > 0)
        {
            if(write(1, buff, n) != n)
            {
                fprintf(stderr, "%s: Ошибка вывода (%s)\n",
                    __FILE__, strerror(errno));
            }
        }
        now=time(NULL);
    }
}
```

Рис. 2.3: Скрипт sever.c (1)

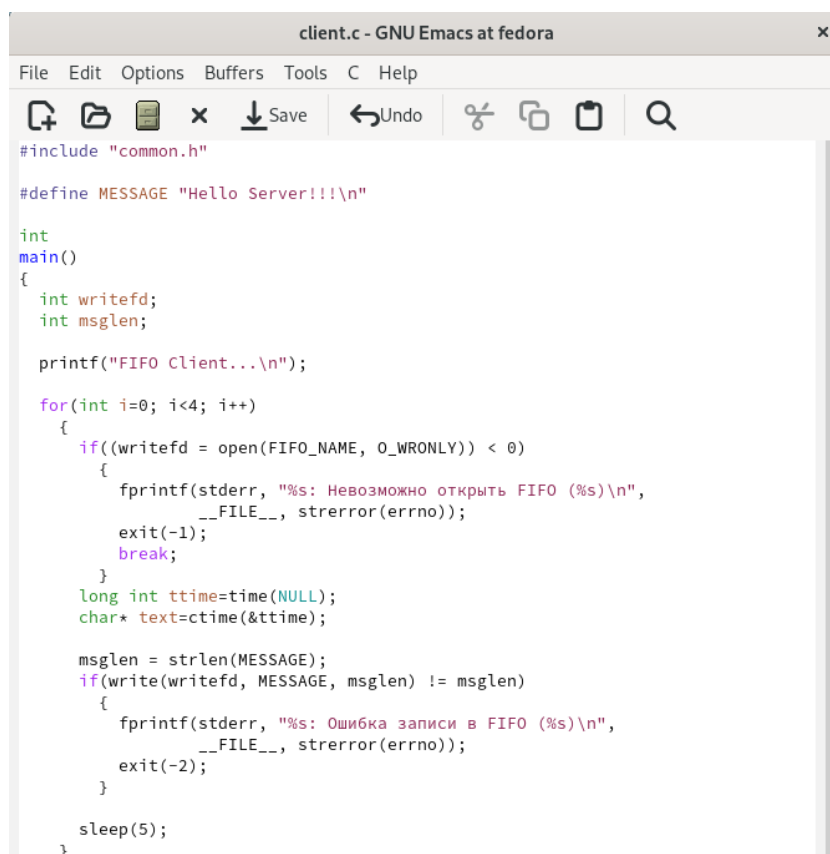


```
    }
    now=time(NULL);
}
printf("server timeout, %li - seconds passed\n", (now-start));
close(readfd);

if(unlink(FIFO_NAME) < 0)
{
    fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
        __FILE__, strerror(errno));
    exit(-4);
}
exit(0);
}
```

Рис. 2.4: Скрипт server.c (2)

Аналогично написала листинг для файла client.c (рис. 2.5), (рис. 2.6).

A screenshot of the GNU Emacs editor window titled 'client.c - GNU Emacs at fedora'. The window shows the C source code for a client program. The code includes a header file 'common.h', defines a message 'Hello Server!!!\n', and implements a main function. The main function opens a FIFO named 'FIFO_NAME' for writing. It then enters a loop where it writes the message to the FIFO four times, with a 5-second delay between each write. Error handling is provided for both opening the FIFO and writing to it. The code is as follows:

```
#include "common.h"

#define MESSAGE "Hello Server!!!\n"

int
main()
{
    int writefd;
    int msglen;

    printf("FIFO Client...\n");

    for(int i=0; i<4; i++)
    {
        if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-1);
            break;
        }
        long int ttime=time(NULL);
        char* text=ctime(&ttime);

        msglen = strlen(MESSAGE);
        if(write(writefd, MESSAGE, msglen) != msglen)
        {
            fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-2);
        }

        sleep(5);
    }
}
```

Рис. 2.5: Скрипт client.c (1)

A continuation of the GNU Emacs editor window showing the final part of the client.c code. The code continues from the previous block, showing the closing of the FIFO and the program's termination. The code is as follows:

```
        exit(-2);
    }

    sleep(5);
}

close(writefd);
exit(0);
}
```

Рис. 2.6: Скрипт client.c (2)

Листинг для Makefile просто переписала (рис. 2.7).


```
all: server client

server: server.c common.h
    gcc server.c -o server

client: client.c common.h
    gcc client.c -o client

clean:
    rm server client *.o
```

Рис. 2.7: Скрипт Makefile

Затем я выполнила компиляцию исполняемых файлов при помощи команд (рис. 2.8).

```
[sakirilyuk@fedora ~]$ make server
gcc server.c -o server
[sakirilyuk@fedora ~]$ make client
gcc client.c -o client
[sakirilyuk@fedora ~]$
```

Рис. 2.8: Компиляция файлов

После чего я проверила работу файлов (рис. 2.9).

```
[sakirilyuk@fedora ~]$ ./server
FIFO Server...
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
server timeout, 31 - seconds passed
[sakirilyuk@fedora ~]$
```

```
[sakirilyuk@fedora ~]$ ./client
FIFO Client...
[sakirilyuk@fedora ~]$
```

Рис. 2.9: Проверка работы файлов

3 Выводы

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