

Лабораторная работа №14

Коняева Марина НФИбд-01-21

01.06.2022

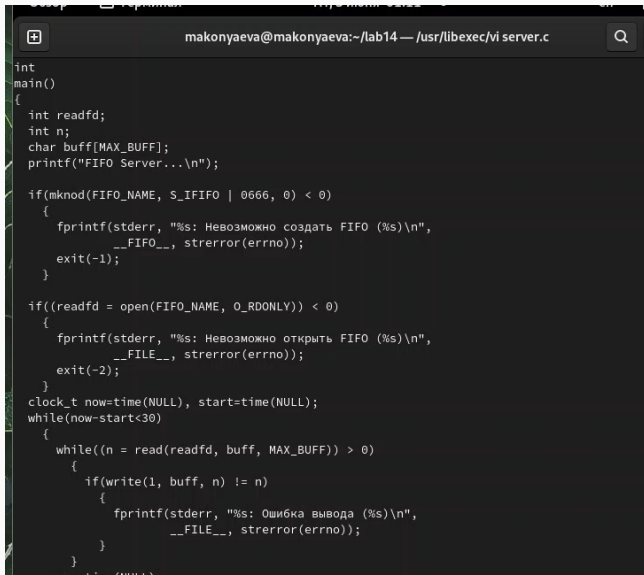
Именованные каналы

- Приобретение практических навыков работы с именованными каналами.

Одним из видов взаимодействия между процессами в операционных системах является обмен сообщениями. Под сообщением понимается последовательность байтов, передаваемая от одного процесса другому. В операционных системах типа UNIX есть 3 вида межпроцессорных взаимодействий: общеюниксовые (именованные каналы, сигналы), System V Interface Definition (SVID — разделяемая память, очередь сообщений, семафоры) и BSD (сокеты). Для передачи данных между неродственными процессами можно использовать механизм именованных каналов (named pipes). Данные передаются по принципу FIFO (First In First Out) (первым записан — первым прочитан), поэтому они называются также FIFO pipes или просто FIFO. Именованные каналы отличаются от неименованных наличием идентификатора канала, который представлен как

Выполнение лабораторной работы

1. Скрипт 1-4 (изображение 1.1-4)



```
makonyaeva@makonyaeva:~/lab14 — /usr/libexec/vi server.c
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",
            __FIFO__, 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));
            }
        }
    }
}
```

```
makonyaeva@makonyaeva:~/lab14 — /usr/libexec/vi client.c
#include "common.h"

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

int
main()
{
    int msg, len, i;
    long int t;

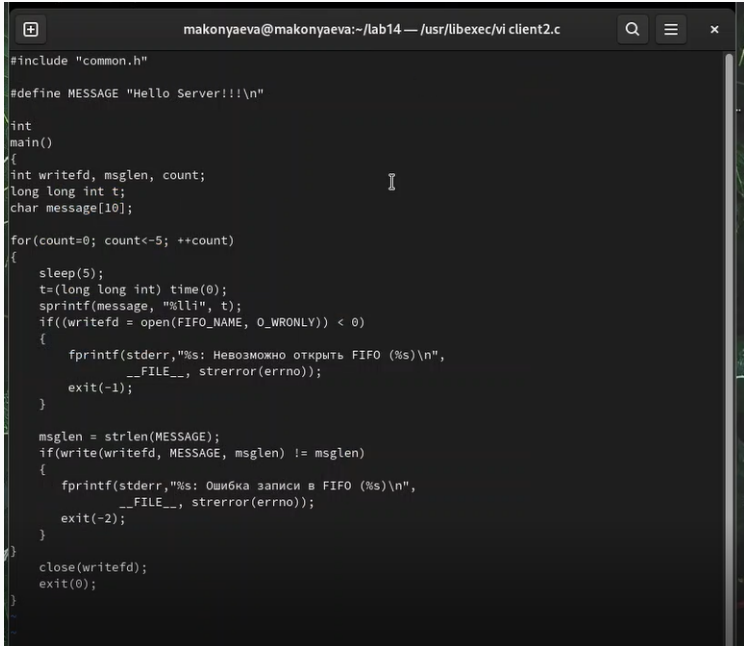
    for(i=0; i<20; i++)
    {
        sleep(3);
        t=time(NULL);
        printf("FIFO Client...\n");

        if((msg = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr,"%s: Невозможно открыть FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-1);
        }

        len = strlen(MESSAGE);

        if(write(msg, MESSAGE, len) != len)
        {
            fprintf(stderr,"%s: Ошибка записи в FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-2);
        }
        close(msg);
    }
    exit(0);
}
```

Изображение 1.2



The image shows a terminal window with a dark background. The title bar at the top reads "makonyaeva@makonyaeva:~/lab14 — /usr/libexec/vi client2.c". The terminal contains C code for a script that attempts to write to a FIFO. The code includes a header "common.h", defines a message "Hello Server!!!\n", and sets up a main function. It declares variables for file descriptors, time, and a message buffer. A loop runs 6 times, each time sleeping for 5 seconds, getting the current time, formatting it into a string, and attempting to open and write to a FIFO named "FIFO_NAME". If the FIFO cannot be opened or the write fails, it prints an error message to stderr and exits with a non-zero status. After the loop, it closes the file descriptor and exits with status 0.

```
#include "common.h"

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

int
main()
{
    int writefd, msglen, count;
    long long int t;
    char message[10];

    for(count=0; count<=5; ++count)
    {
        sleep(5);
        t=(long long int) time(0);
        sprintf(message, "%lli", t);
        if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr,"%s: Невозможно открыть FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-1);
        }

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

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

Изображение 1.3 Скрипт 3

A screenshot of a terminal window with a dark background. The title bar at the top shows the user 'makonyaeva@makonyaeva' and the current directory '~ /lab14'. The terminal content displays a Makefile with targets for 'server' and 'client', each depending on 'common.h' and compiled with 'gcc'. A 'clean' target is also present to remove the executables and object files. The window includes standard UI elements like a plus icon on the left and a search icon on the right.

```
makonyaeva@makonyaeva:~/lab14 — /usr/libexec/vi Makefile

all: server client

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

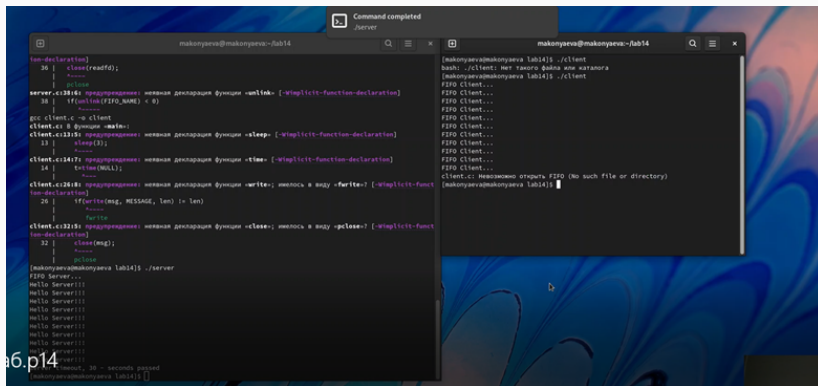
client: client.c common.h
    gcc client.c -o client

clean:
    -rm server client *.o

~
~
~
~
~
```

Изображение 1.4 Скрипт 4

2. Запустим в разных консолях (изображение 2.1)



Изображение 2.1 Запустим в разных консолях

В ходе данной лабораторной работы приобрели практических навыков работы с именованными каналами, а также ответили на контрольные вопросы.