

Лабораторная работа №14

Желдакова Виктория Алексеевна

3 июня 2022 г.

Российский университет дружбы народов

Именованные каналы

Приобретение практических навыков работы с именованными каналами.

Выполнение лабораторной работы

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.

Скопировали содержимое файлов `common.h` и `Makefile` из лабораторной работы.

Выполнение лабораторной работы

```
vazheldakova@fedora:~/work/os/server — /usr/lib...
#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);
    }
    printf("server timeout, %li - second 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);
}
```

Выполнение лабораторной работы

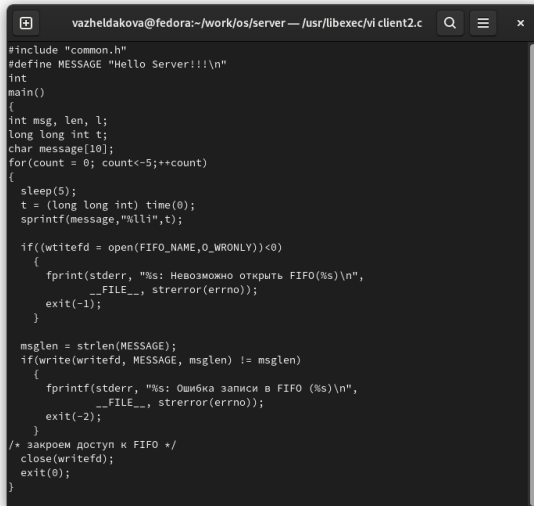
```
vazheldakova@fedora:~/work/os/server — /usr/lib...
#include "common.h"
#define MESSAGE "Hello Server!!!\n"
int
main()
{
    int msg, len, i;
    long int t;
    for(i=0; i<20;i++)
    {
        sleep(3);
        t = time(NULL);
        printf("FIFO client...\n");

        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 в записи в FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-2);
        }
        close(msg);
    }
    exit(0);
}
```

"client.c" 32L, 619B

Выполнение лабораторной работы



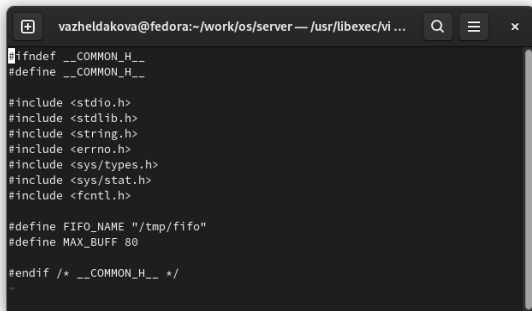
```
#include "common.h"
#define MESSAGE "Hello Server!!!\n"
int
main()
{
    int msg, len, l;
    long long int t;
    char message[10];
    for(count = 0; count<5;++count)
    {
        sleep(5);
        t = (long long int) time(0);
        sprintf(message,"%lli",t);

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

        msglen = strlen(MESSAGE);
        if(write(wtfefd, MESSAGE, msglen) != msglen)
        {
            fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-2);
        }
    }
    /* закроем доступ к FIFO */
    close(wtfefd);
    exit(0);
}
```

Figure 3: Содержимое client2.c

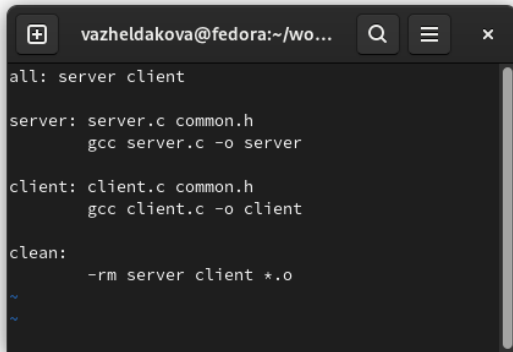
Выполнение лабораторной работы

A screenshot of a terminal window with a dark background. The window title bar shows the user 'vazheldakova@fedora' and the current directory path '~ /work/os/server'. The terminal displays the content of a file named 'common.h'. The code includes several standard C headers, defines constants for a FIFO name and buffer size, and uses a guard to prevent multiple inclusions.

```
+ vazheldakova@fedora:~/work/os/server — /usr/libexec/vi ...  
#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>  
  
#define FIFO_NAME "/tmp/fifo"  
#define MAX_BUFF 80  
  
#endif /* __COMMON_H__ */
```

Figure 4: Содержимое common.h

Выполнение лабораторной работы

A terminal window with a dark background and light text. The window title bar shows a plus icon, the user and host 'vazheldakova@fedora:~/wo...', a search icon, a menu icon, and a close icon. The terminal content is a Makefile with targets 'all', 'server', 'client', and 'clean'. The 'all' target depends on 'server' and 'client'. The 'server' target depends on 'server.c' and 'common.h' and runs 'gcc server.c -o server'. The 'client' target depends on 'client.c' and 'common.h' and runs 'gcc client.c -o client'. The 'clean' target runs '-rm server client *.o'. There are two blue cursor icons at the bottom.

```
all: server client

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

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

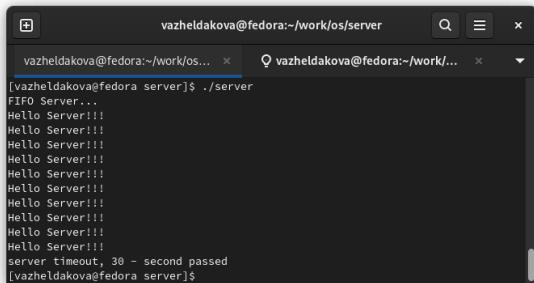
clean:
    -rm server client *.o

~
~
```

Figure 5: Содержимое Makefile

Выполнение лабораторной работы

Запустили make, server в одном окне и client в другом (рис. 6)

A terminal window titled 'vazheldakova@fedora:~/work/os/server' with search and menu icons. It contains two tabs: 'vazheldakova@fedora:~/work/os/...' and 'vazheldakova@fedora:~/work/...'. The active tab shows the command '[vazheldakova@fedora server]\$./server' and its output: 'FIFO Server...', followed by ten 'Hello Server!!!' messages, and finally 'server timeout, 30 - second passed'. The prompt returns to '[vazheldakova@fedora server]\$'.

```
[vazheldakova@fedora server]$ ./server
FIFO Server...
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
server timeout, 30 - second passed
[vazheldakova@fedora server]$
```

Figure 6: Работа программы

Сообщения передевалась с периодичностью в несколько секунд, а по истечении 30 секунд - канал закрылся, и вывелось сообщение об этом.

Что будет в случае, если сервер завершит работу, не закрыв канал?

При завершении программы каналы автоматически закрываются.

Приобретение практических навыков работы с именованными каналами.