

Лабораторная работа №14

Цель работы

Приобретение практических навыков работы с именованными каналами.

Выполнение лабораторной работы

1. Работает не 1 клиент, а несколько (например, два).

```
[asmaslova@fedora 1]$ ./server
FIFO Server...
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
[asmaslova@fedora 1]$
```

рис.1 Пример работы программы

```
[asmaslova@fedora 1]$ ./client1
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
[asmaslova@fedora 1]$
```

рис.2 Пример работы программы

```

#include "common.h"

int main()
{
    int readfd; /* дескриптор для чтения из FIFO */
    int n;
    char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */

    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);
    }

    while((n = read(readfd, buff, MAX_BUFF)) > 0)
    {
        if(write(1, buff, n) != n)
        {
            fprintf(stderr, "%s: Ошибка вывода (%s)\n",
                __FILE__, strerror(errno));
            exit(-3);
        }
    }

    close(readfd);

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

    exit(0);
}

```

U:--- server.c All L1 (C/*l Abbrev)

рис.3 Текст программы

```

#include "common.h"

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

int main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;

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

    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);
}

```

U:--- client.c All LI (C/*l Abbrev)

рис.4 Текст программы

2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Используйте функцию `sleep()` для приостановки работы клиента.


```
[asmaslova@fedora 2]$ ./server
FIFO Server...
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
-----
server timeout
30 seconds passed!
-----
```

рис.5 Пример работы программы

```
[asmaslova@fedora 2]$ ./client
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
[asmaslova@fedora 2]$
```

рис.6 Пример работы программы

```

#include "common.h"

int main()
{
    int readfd; /* дескриптор для чтения из FIFO */
    int n;
    char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */

    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));
                exit(-3);
            }
        }
        now=time(NULL);
    }

    printf("\n-----\nserver timeout\n%i seconds passed!\n-----\n", now-start);
    close(readfd);

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

```

U:--- server.c Top L16 (C/*l Abbrev)

Beginning of buffer

рис.7 Текст программы

```

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));
            exit(-3);
        }
    }
    now=time(NULL);
}

printf("\n-----\nserver timeout\n%li seconds passed!\n-----\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);
}

```

U:--- server.c Bot L26 (C/*l Abbrev)

рис.8 Текст программы

```

#include "common.h"

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

int main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;
    char message[10];
    int count;
    long long int T;
    for(count=0; count<=5; ++count)
    {
        sleep(5);
        T=(long long int) time(0);
        sprintf(message,"%lli", T);
        message[9] = '\n';
        printf("FIFO Client...\n");

        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);
}

```

```

U:--- client.c All L1 (C/*l Abbrev)
Beginning of buffer

```

рис.9 Текст программы

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?

```
[asmaslova@fedora 3]$ ./server
FIFO Server...
Hello Server!!!

-----

server timeout
30 seconds passed!
-----
```

рис.10 Пример работы программы

```
[asmaslova@fedora 3]$ ./client  
FIFO Client...
```

рис.11 Пример работы программы


```
File Edit Options Buffers Tools C Help
[Icons: New, Open, Save, Print, Undo, Redo, Find, etc.]

#include "common.h"

int main()
{
    int readfd; /* дескриптор для чтения из FIFO */
    int n;
    char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */

    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));
                exit(-3);
            }
        }
        now=time(NULL);
    }

    printf("\n-----\nServer timeout!\n%i seconds passed!\n-----\n", now-start);
    close(readfd);

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

U:++- server.c FILE , strerror(errno);
Top L40 (C++1 Abbrev)

Beginning of buffer

рис.12 Текст программы

```

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));
            exit(-3);
        }
    }
    now=time(NULL);
}

printf("\n-----\nServer timeout\n%i seconds passed!\n-----\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);
}

```

U:**- **server.c** Bot L40 (C/*l Abbrev)

рис.13 Текст программы

```

#include "common.h"

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

int main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;

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

    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);
}

```

U:--- client.c All L1 (C/*l Abbrev)
Beginning of buffer

рис.14 Текст программы

Выводы

В ходе выполнения лабораторной работы я приобрела практические навыки работы с именованными каналами.