

Презентация к лабораторной работе №15

Тема:

Именованные каналы

Российский Университет Дружбы Народов

Факультет Физико-Математических и Естественных Наук

Дисциплина: *Операционные системы*

Студент: Мухамедияр Адиль

Группа: НКНбд-01-20

Москва, 2021г.

Цель работы

Приобретение практических навыков работы с именованными каналами.

Введение

- Идею программных каналов и значок вертикальной черты как их обозначение придумал Douglas McIlroy, один из авторов ранних командных оболочек. Он обратил внимание на то, сколько времени уходит на обработку вывода одной программы в качестве ввода другой. Его идеи были внедрены в жизнь, когда в 1973 Ken Thompson добавил программные каналы в операционную систему Юникс. Идея была со временем позаимствована другими ОС, такими как DOS, OS/2, Microsoft Windows, и BeOS, часто даже с тем же обозначением.
 - Английское название именованного канала - named pipe или FIFO (File In, File Out - файл пришел, файл ушел). Именованные каналы служат в основном для межпроцессного взаимодействия, когда различные процессы в системе обмениваются информацией. Тема это сложная и большая, заслуживающая отдельной статьи. Поэтому в данной работе я только вкратце коснусь ее.
-

Ход работы:

- Изучил приведённые в тексте программу *common.h* и взял данный пример за образец.

File Edit Options Buffers Tools C Help



Save



Undo

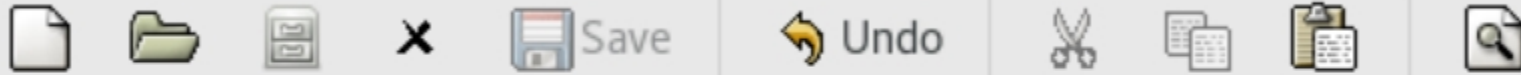


```
#ifndef __COMMON_H__
#define __COMMON_H__
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

#define FIFO_NAME
#define MAX_BUFF
"/tmp/fifo"
80
#endif /* __COMMON_H__ */
```

- Изучил приведённые в тексте программу *server.c* и взял данный пример за образец.

File Edit Options Buffers Tools C Help



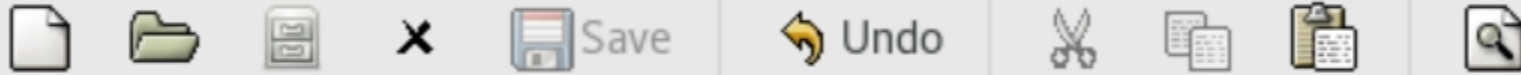
```
#include "common.h"
int main()
{
int readfd;
int n;
char buff[MAX_BUFF];
printf("FIFO Server...\n");

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);
}
```

- Продолжение кода файла *server.c*.

File Edit Options Buffers Tools C Help



```
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);
    }
}
```

- Изучил приведённые в тексте программу *client.c* и взял данный пример за образец.

File Edit Options Buffers Tools C Help



```
#include "common.h"
#define MESSAGE "Hello Server!!!\n"
int
main()
{
    int writefd;
    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",
```

- Продолжение кода файла *client.c*.

File Edit Options Buffers Tools C Help



```
fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
__FILE__, strerror(errno));
exit(-2);
}
```

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

2. Написал аналогичные программы, внося следующие изменения:

- В коде файла *common.h* добавил библиотеку *time.h*, для использования таких функций, как *sleep* и *clock*.

File Edit Options Buffers Tools C Help



```
#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 <time.h>

#define FIFO_NAME "/tmp/fifo"
#define MAX_BUFF 80
#endif /* __COMMON_H__ */
```

- В код файла *server.c* так же ввел некоторые изменения.

File Edit Options Buffers Tools C Help



```
#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);
    }
    int start = time(NULL);
    while (time(NULL) - start <= 30){
        if((readfd=open(FIFO_NAME, O_RDONLY))<0)
        {
            fprintf(stderr,"%s: Невозможно открыть FIFO (%s)\n",
                FILE , strerror(errno));
```

- Продолжение кода файла *server.c*. Сервер работает не бесконечно, а прекращает работу через некоторое время.

File Edit Options Buffers Tools C Help



```
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);
}
```

- В коде файла *client.c* использовал функцию `sleep()` для приостановки работы клиента. Так же приравнял переменную *ttime* к нулю, задав его в нутри цикла.

File Edit Options Buffers Tools C Help



```
#include "common.h"
#define MESSAGE "Hello Server!!!\n"

int
main()
{
    int writefd;
    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);
    }
    int i;
    for (i=0; i<10; i++){
        sleep(10);
        long ttime = time(NULL);
        msglen = strlen(ctime(&ttime));
        if(write(writefd, ctime(&ttime), msglen)!=msglen)
```

- Продолжение кода файла *client.c*.

client.c - emacs@amukhamediyar.localdomain

File Edit Options Buffers Tools C Help

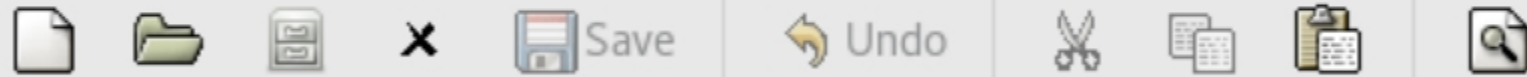


```
if(write(writefd, ctime(&ttime), msglen)!=msglen)
{
    fprintf(stderr,"%s: Ошибка записи в FIFO (%s)\n",
        __FILE__, strerror(errno));
    exit(-2);
}
close(writefd);
exit(0);
}
```

- С помощью команды *make* запустил *makefile*.

makefile - emacs@amukhamediyar.localdomain

File Edit Options Buffers Tools Makefile Help



```
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
```

- Проверяю, запустил коды в файлах *server.c* и *client.c*

```
[amukhamediyar@amukhamediyar lab_15]$ ./server
FIFO Server...
server.c: Невозможно создать FIFO (File exists)
[amukhamediyar@amukhamediyar lab_15]$ █
```

- В случае, если сервер завершит работу, не закрыв канал, файл FIFO не удалится, поэтому его в следующий раз создать будет нельзя и вылезет ошибка, следовательно, работать ничего не будет.

```
amukhamediyar@amukh...  ×  amukhamediyar@amukh...  ×

[amukhamediyar@amukhamediyar lab_15]$ ./client
FIFO Client...
█
```

Вывод

Приобрел практические навыки работы с именованными каналами.

Библиография

[Программные каналы в Linux](#)

[Введение в каналы и именованные каналы в Linux](#)
