

Лабораторная работа №14

Именованные каналы

Бабенко Артём Сергеевич

Содержание

Цель работы	3
Теоретическое введение	4
Ход работы	5
Выводы	11
Контрольные вопросы	12

Цель работы

Приобретение практических навыков работы с именованными каналами.

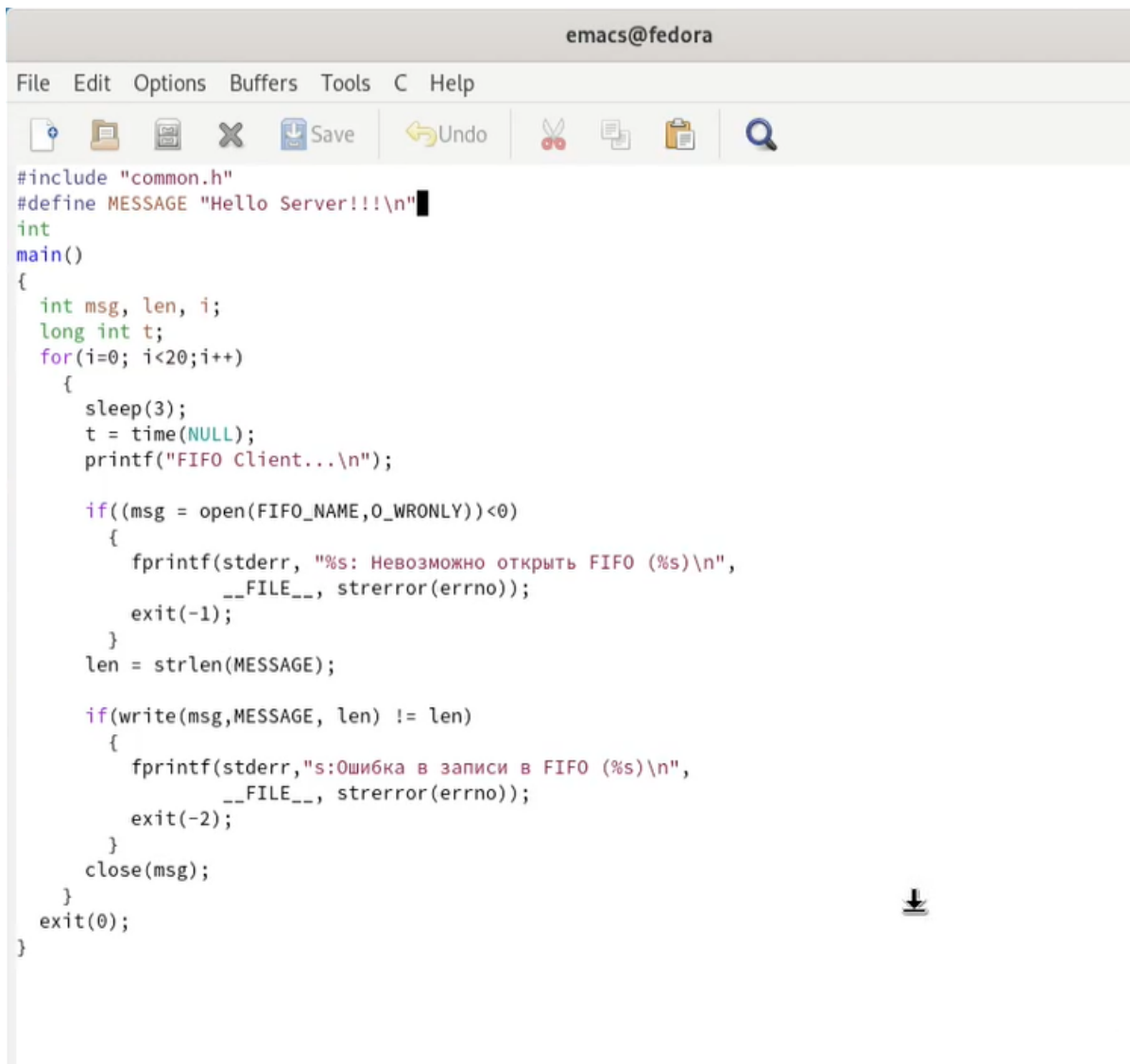
Теоретическое введение

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: общемуниксные (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 специальный файл (соответственно имя именованного канала — это имя файла). Поскольку файл находится на локальной файловой системе, данное IPC используется внутри одной системы.

Ход работы

1. Создал необходимые файлы и написал в них программы (рис.1-5).

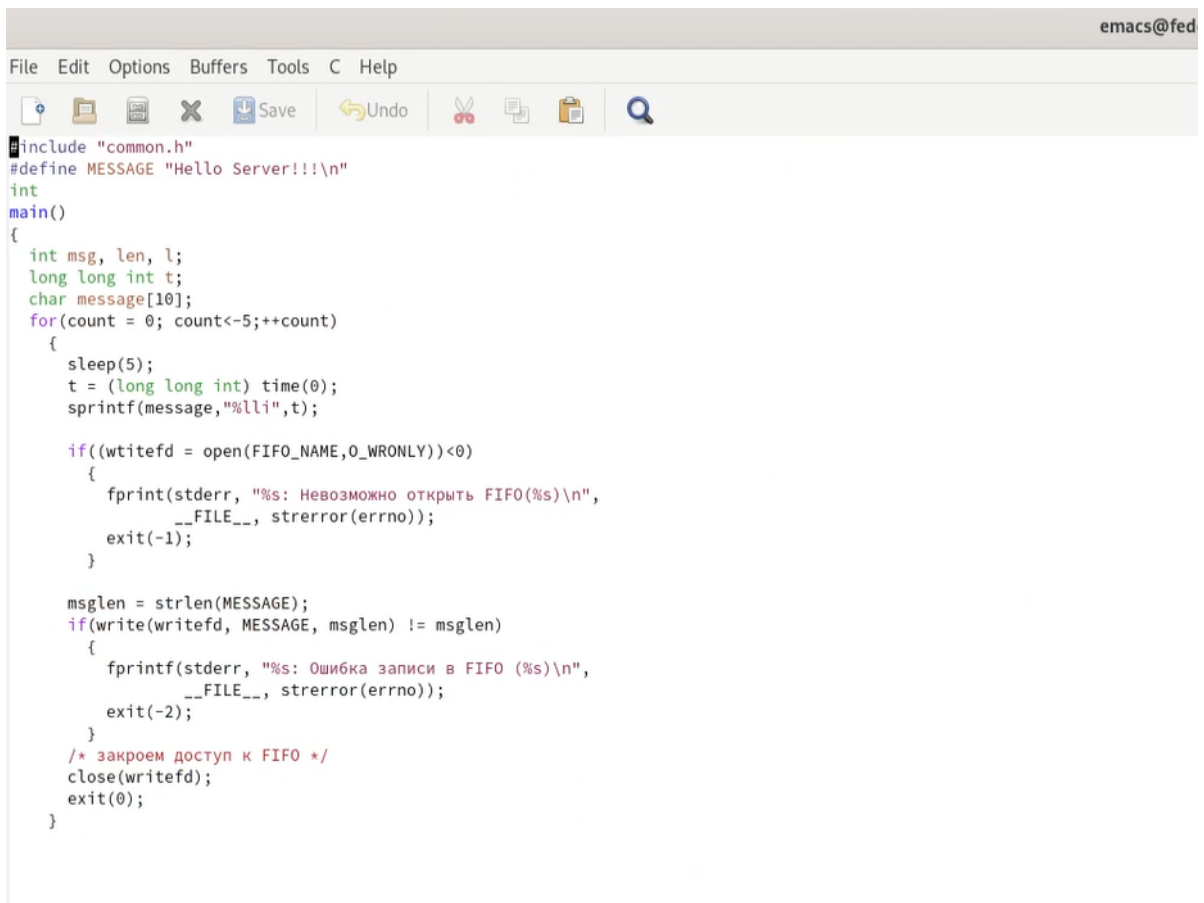


```
#include "common.h"
#define MESSAGE "Hello Server!!!\n"
int
main()
{
    int msg, len, i;
    long int t;
    for(i=0; i<20;i++)
    {
        sleep(3);
        t = time(NULL);
        printf("FIFO Client...\n");

        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);
}
```

Рис.1. Листинг client.c

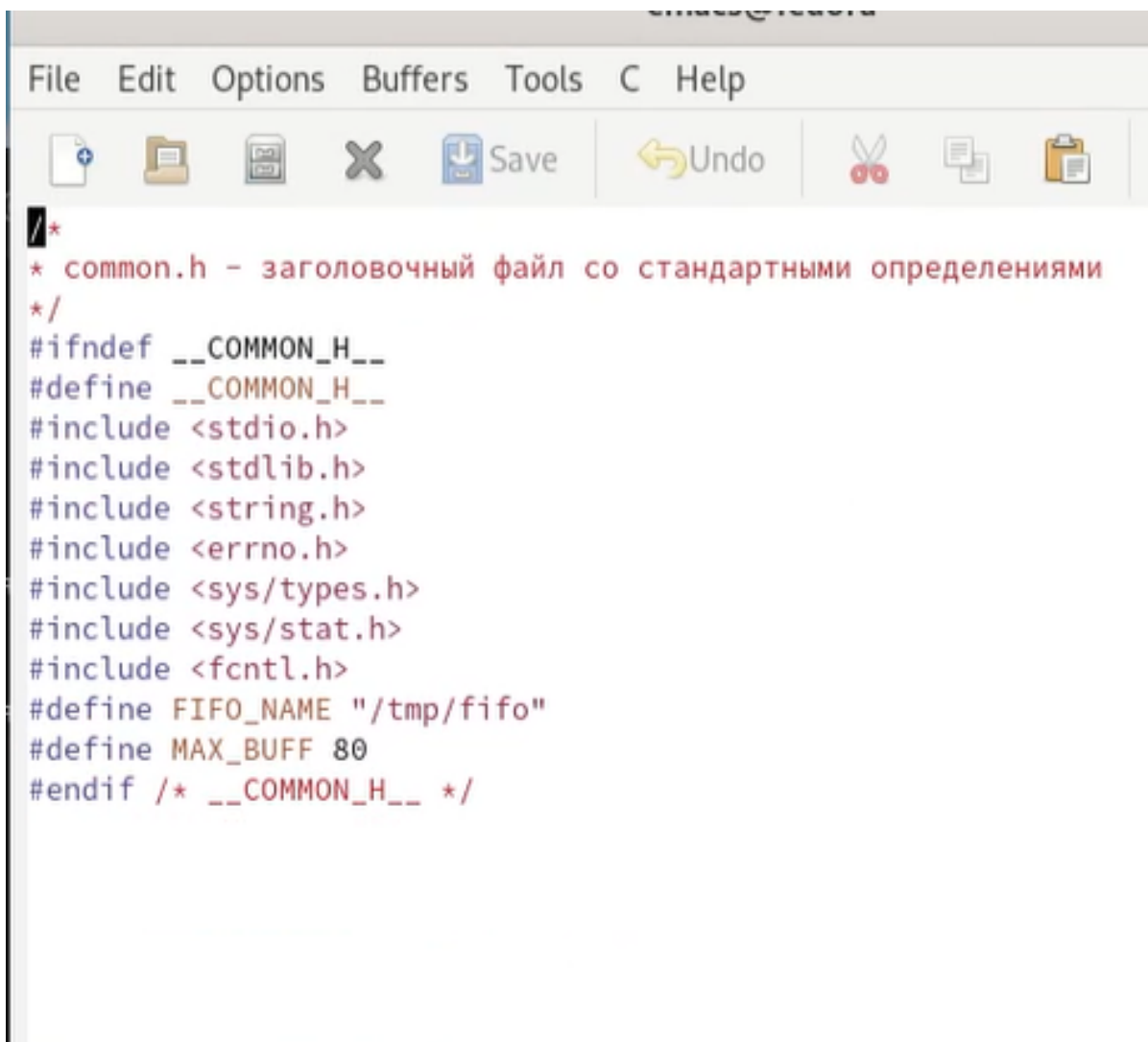


```
#include "common.h"
#define MESSAGE "Hello Server!!!\n"
int
main()
{
    int msg, len, l;
    long long int t;
    char message[10];
    for(count = 0; count<-5;++count)
    {
        sleep(5);
        t = (long long int) time(0);
        sprintf(message,"%lli",t);

        if((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);
        }
        /* закроем доступ к FIFO */
        close(writefd);
        exit(0);
    }
}
```

Рис.2. Листинг client1.c



```
/*
 * common.h - заголовочный файл со стандартными определениями
 */
#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__ */
```

Рис.3. Листинг common.h

```
emacs@fedora
File Edit Options Buffers Tools C Help
[Icons: Save, Undo, Cut, Copy, Paste, Find]

#include "common.h"
int
main()
{
    int readfd; /* Прочитать данные с FIFO */
    int n;
    char buff[MAX_BUFF]; /* Буфер для чтения данных с FIFO */
    /* Печать сообщения о запуске сервера */
    printf("FIFO Server...\n");

    if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
    {
        fprintf(stderr, "%s: Не удалось создать FIFO (%s)\n",
            __FILE__, strerror(errno));
        exit(-1);
    }
    /* Создать FIFO */
    if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
    {
        fprintf(stderr, "%s: Не удалось 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: Не удалось записать данные (%s)\n",
                    __FILE__, strerror(errno));
            }
        }
        now=time(NULL);
    }
    printf("server timeout, %li - second passed\n", (now-start));
    close(readfd); /* Закрыть FIFO */
    /* Удалить FIFO */
    if(unlink(FIFO_NAME) < 0)
    {

```

Рис.4. Листинг server.c

Рис.6. Проверка работы программы

Выводы

Я приобрёл практические навыки работы с именованными каналами.

Контрольные вопросы

1. Именованные каналы отличаются от неименованных наличием идентификатора канала, который представлен как специальный файл (соответственно имя именованного канала — это имя файла). Поскольку файл находится на локальной файловой системе, данное IPC используется внутри одной системы.
2. Да, с помощью команды `pipe`.
3. Да, с помощью команды `mkfifo`.