

Лабораторная работа № 14

Операционные системы

Голованова Мария Константиновна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Файл common.h	9
4.2	Файл client.c	10
4.3	Файл client-2.c	11
4.4	Файл server.c (часть 1)	12
4.5	Файл server.c (часть 2)	12
4.6	Файл Makefile	13

Список таблиц

1 Цель работы

Приобретение практических навыков работы с именованными каналами.

2 Задание

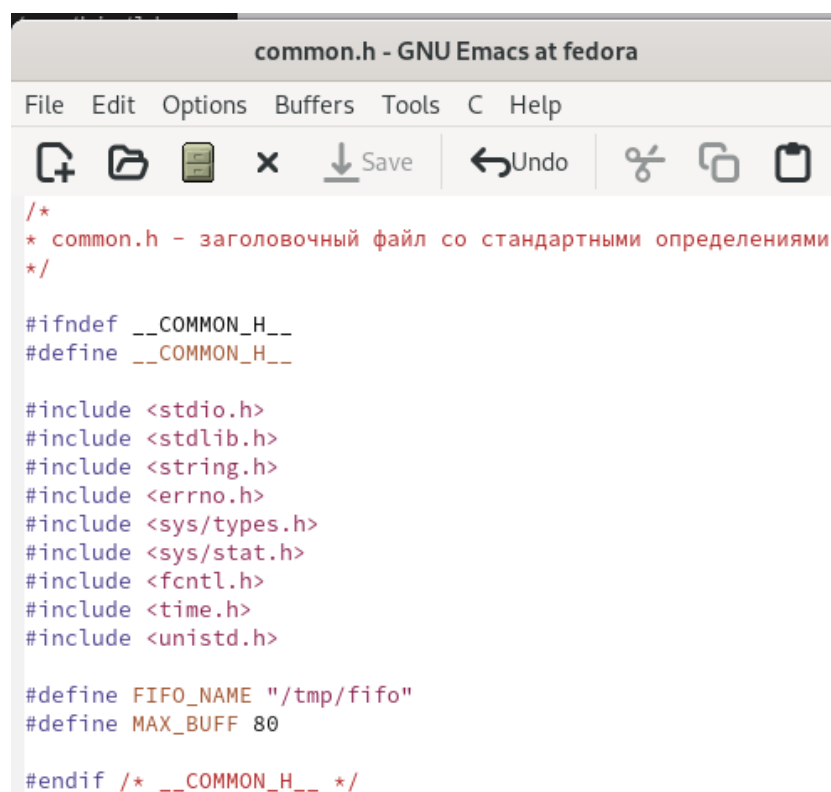
Изучите приведённые в тексте программы `server.c` и `client.c`. Взяв данные примеры за образец, напишите аналогичные программы, внося следующие изменения: 1. Работает не 1 клиент, а несколько (например, два). 2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). 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. Что будет в случае, если сервер завершит работу, не закрыв канал?

3 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: общедоступные (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.

4 Выполнение лабораторной работы

Я изучила приведённые в тексте программы `server.c` и `client.c` и, взяв данные примеры за образец, написала аналогичные программы, внося следующие изменения: 1. Работает не 1 клиент, а несколько два. 2. Клиенты передают текущее время с некоторой периодичностью раз в 15 секунд (Использовала функцию `sleep()` для приостановки работы клиента). 3. Сервер работает не бесконечно, а прекращает работу через некоторое время 40 сек. (Использовала функцию `clock()` для определения времени работы сервера). (рис. 4.1, рис. 4.2, рис. 4.3, рис. 4.4, рис. 4.5, рис. 4.6).



The image shows a screenshot of the GNU Emacs editor window titled "common.h - GNU Emacs at fedora". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "C", and "Help". The toolbar contains icons for opening, saving, undo, redo, and other editing functions. The main text area displays the following C header file content:

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

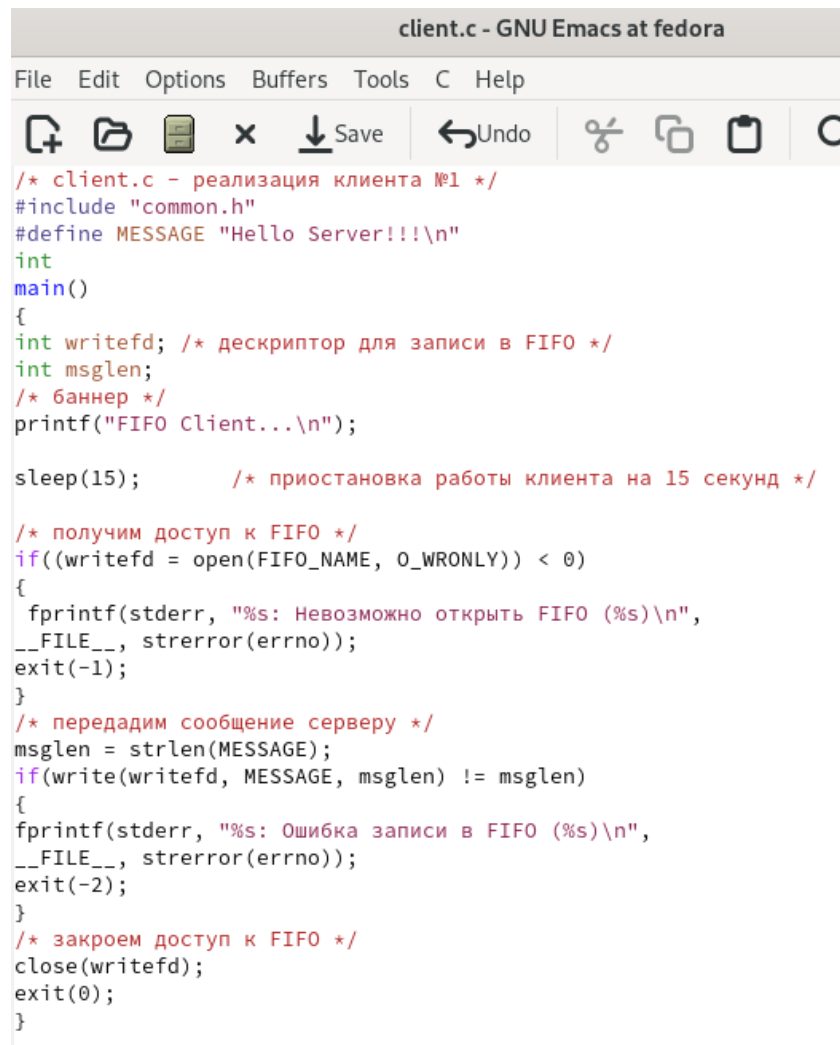
#ifndef __COMMON_H__
#define __COMMON_H__

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <time.h>
#include <unistd.h>

#define FIFO_NAME "/tmp/fifo"
#define MAX_BUFF 80

#endif /* __COMMON_H__ */
```

Рис. 4.1: Файл common.h



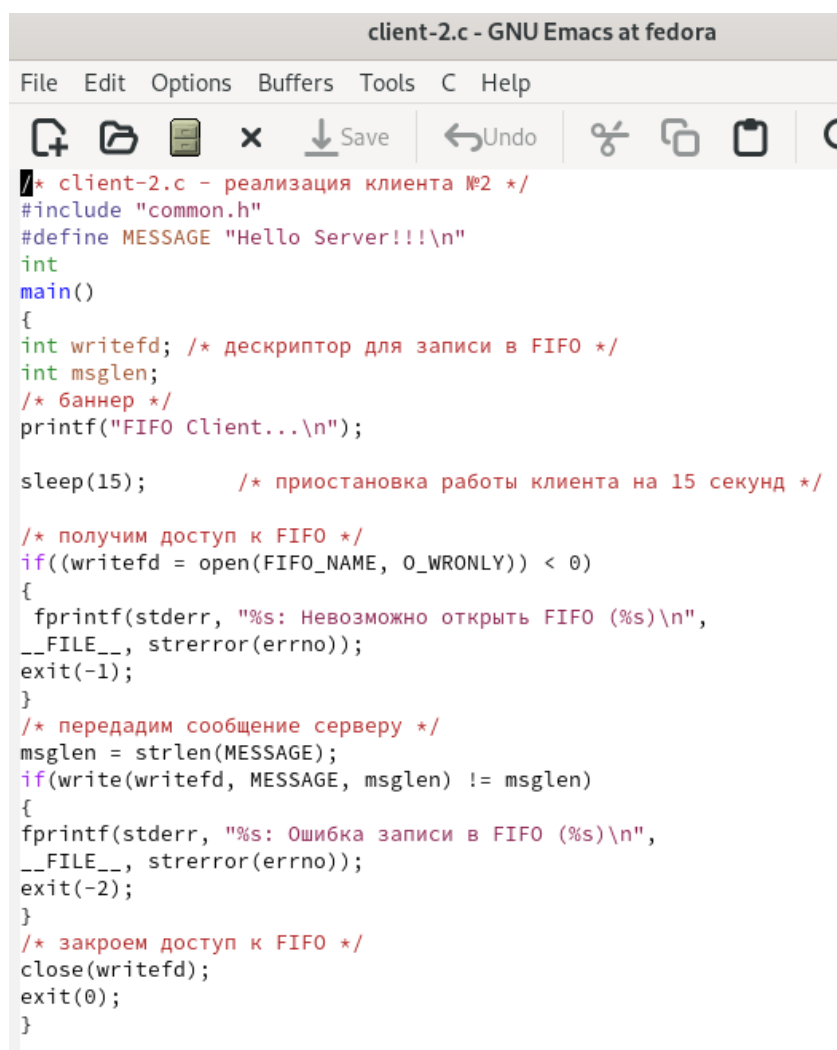
```
client.c - GNU Emacs at fedora
File Edit Options Buffers Tools C Help
[Icons: New, Open, Save, Close, Save All, Undo, Cut, Copy, Paste]

/* client.c - реализация клиента №1 */
#include "common.h"
#define MESSAGE "Hello Server!!!\n"
int
main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;
    /* баннер */
    printf("FIFO Client...\n");

    sleep(15); /* приостановка работы клиента на 15 секунд */

    /* получим доступ к FIFO */
    if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
    {
        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
            __FILE__, strerror(errno));
        exit(-1);
    }
    /* передадим сообщение серверу */
    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. 4.2: Файл client.c



```
/* client-2.c - реализация клиента №2 */
#include "common.h"
#define MESSAGE "Hello Server!!!\n"
int
main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;
    /* баннер */
    printf("FIFO Client...\n");

    sleep(15);          /* приостановка работы клиента на 15 секунд */

    /* получим доступ к FIFO */
    if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
    {
        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
            __FILE__, strerror(errno));
        exit(-1);
    }
    /* передадим сообщение серверу */
    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. 4.3: Файл client-2.c

```
server.c - GNU Emacs at fedora
File Edit Options Buffers Tools C Help
[Icons: Open, Save, Undo, Redo, Search]

/*server.c - реализация сервера */
#include "common.h"
int
main()
{
    int readfd; /* дескриптор для чтения из FIFO */
    int n;
    char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */
    /* баннер */
    printf("FIFO Server...\n");

    /* создаем файл FIFO с открытыми для всех правами доступа на чтение и запись */
    if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
    {
        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
            __FILE__, strerror(errno));
        exit(-1);
    }
    /* откроем FIFO на чтение */
    if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
    {
        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
            __FILE__, strerror(errno));
        exit(-2);
    }
    /* читаем данные из FIFO и выводим на экран */
    while((n = read(readfd, buff, MAX_BUFF)) > 0)
    {
        if(write(1, buff, n) != n)
        {
            U:--- server.c          Top L29      (C/*l Abbrev)
```

Рис. 4.4: Файл server.c (часть 1)

```
server.c - GNU Emacs at fedora
File Edit Options Buffers Tools C Help
[Icons: Open, Save, Undo, Redo, Search]

        if(write(1, buff, n) != n)
        {
            fprintf(stderr, "%s: Ошибка вывода (%s)\n",
                __FILE__, strerror(errno));
            exit(-3);
        }
        close(readfd); /* закроем FIFO */
        /* удалим FIFO из системы */
        if(unlink(FIFO_NAME) < 0)
        {
            fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-4);
        }
        exit(0);
    }
}
```

Рис. 4.5: Файл server.c (часть 2)

```
Makefile - GNU Emacs at fedora
File Edit Options Buffers Tools Makefile Help
[Icons: New, Open, Save, Close, Save, Undo, Cut, Copy, Paste]
all: server client client-2

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

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

client-2: client-2.c common.h
gcc client-2.c -o client-2

clean:
-rm server client client-2 *.o
█
```

Рис. 4.6: Файл Makefile

5 Выводы

Я приобрела практические навыки работы с именованными каналами.

Список литературы