

Лабораторная работа №2
студента группы ИТ – 32
Курбатовой Софьи Андреевны

Выполнение: _____ Защита _____

ИНТЕРАКТИВНЫЕ УСТРОЙСТВА ВВОДА

Цель работы: освоить методы работы с интерактивными устройствами ввода информации, научиться использовать поступающую информацию в разрабатываемых программных средствах.

Содержание работы

Вариант №10

1. Ознакомиться с теоретическим материалом.
2. Разработать алгоритм задачи, представленной в варианте № 10 (Рис. 2.1). и представить его в виде блок-схемы.
3. Реализовать разработанный алгоритм на одном из языков программирования. Для верхнего регистра использовать клавишу SHIFT.
4. В отчет занести основные сведения об аппаратному устройстве согласно заданию (клавиатура), параметры портов и интерфейсов.
5. Результаты работы программы представить в виде скриншотов.

10	Отобразить на экране двоичные коды вводимых с клавиатуры символов. Использовать символы s, f, k, w, e и реализовать по этим кодам запуск программы Проводник, указание пути к папке, создание текстового файла, удаление файла, закрытие текущего окна Проводника соответственно.
----	---

Рис. 2.1. Вариант задания

6. Сделать вывод о проделанной работе.

Ход работы

1. В процессе выполнения лабораторной работы были рассмотрены страницы 7-19 методических рекомендаций по дисциплине «Периферийное оборудование».

2. На устройстве LENOVO 81D6 Lenovo ideapad 330-15AST установлена стандартная клавиатура с интерфейсом PS/2. Подключение клавиатуры к плате производится с помощью сверхплоского кабеля (шлейфа) и соответствующих разъемов. Клавиатура является модернизацией клавиатуры AT – 101 клавиша.

Параметр	Значение	USB 3.0 5	
Touch Pad		Внутреннее обозначение	J1502
Внутреннее обозначение	J3809	Внешнее обозначение	USB 3.0 5
Внешнее обозначение	Touch Pad	Тип порта	USB
Тип порта	Mouse Port	Внутренний тип разъема	Her
Внутренний тип разъема	Her	Внешний тип разъема	Access Bus (USB)
Внешний тип разъема	PS/2	USB 2.0 1	
Broadcom BCM5762 LAN		Внутреннее обозначение	J1501
Внутреннее обозначение	U1302	Внешнее обозначение	USB 2.0 1
Внешнее обозначение	Broadcom BCM5762 LAN	Тип порта	USB
Тип порта	Network Port	Внутренний тип разъема	Her
Внутренний тип разъема	Her	Внешний тип разъема	Access Bus (USB)
Внешний тип разъема	RJ-45	Web Cam	
USB 3.0 4		Внутреннее обозначение	J3813
Внутреннее обозначение	J1500	Внешнее обозначение	Web Cam
Внешнее обозначение	USB 3.0 4	Тип порта	Other
Тип порта	USB	Внутренний тип разъема	Her
Внутренний тип разъема	Her	Внешний тип разъема	Other
Внешний тип разъема	Access Bus (USB)		

Рис. 2.2. Параметры портов: ч.1.

Параметр	Значение	Audio Line Out	
SATA 0 (HDD)		Внутреннее обозначение	J2102
Внутреннее обозначение	J1703	Внешнее обозначение	Audio Line Out
Внешнее обозначение	SATA 0 (HDD)	Тип порта	Audio Port
Тип порта	SATA	Внутренний тип разъема	Her
Внутренний тип разъема	Her	Внешний тип разъема	Mini-jack (headphones)
Внешний тип разъема	SAS/SATA Plug Receptacle	Display Port0	
SATA 1 (Cable SATA ODD)		Внутреннее обозначение	J1100
Внутреннее обозначение	J1705	Внешнее обозначение	Display Port0
Внешнее обозначение	SATA 1 (Cable SATA ODD)	Тип порта	Video Port
Тип порта	SATA	Внутренний тип разъема	Her
Внутренний тип разъема	Her	Внешний тип разъема	Other
Внешний тип разъема	SAS/SATA Plug Receptacle	Display Port1	
Audio Line In		Внутреннее обозначение	J1101
Внутреннее обозначение	J2105	Внешнее обозначение	Display Port1
Внешнее обозначение	Audio Line In	Тип порта	Video Port
Тип порта	Audio Port	Внутренний тип разъема	Her
Внутренний тип разъема	Her	Внешний тип разъема	Other
Внешний тип разъема	Mini-jack (headphones)		

Рис. 2.3. Параметры портов: ч.2.

Имя устройства	Класс устройства	Версия драйв...	Дата драйве...	Производитель драйвера
Стандартная клавиатура PS/2		10.0.17763.348	6-21-2006	Microsoft

Рис. 2.4. Информация об установленной клавиатуре

3. Для выполнения, определенного в варианте задания, были описаны представленные на рис. 2.2. – 2.5. алгоритмы в виде блок-схем.

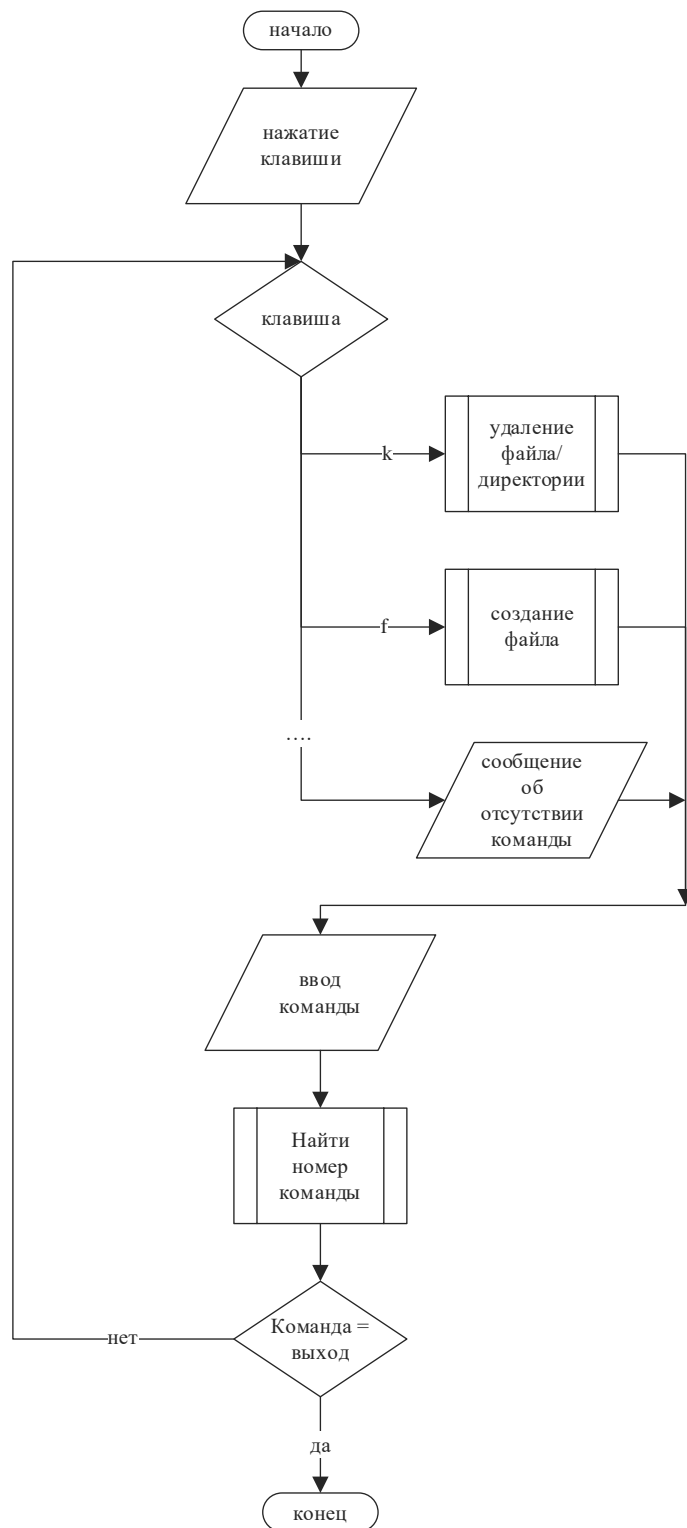


Рис. 2.5. Блок-схема алгоритма запуска функции по нажатию клавиши

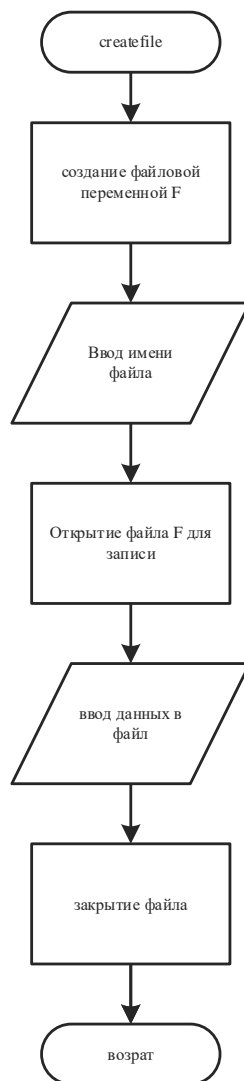


Рис. 2.6. Блок-схема алгоритма создания тестового файла

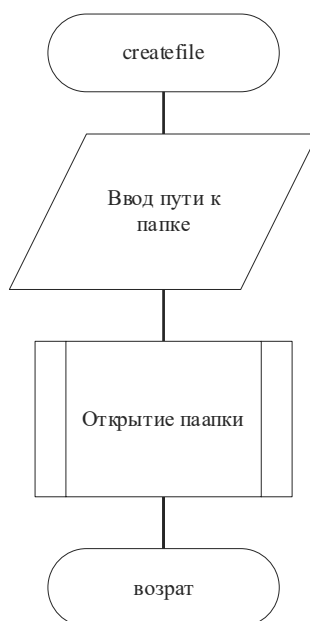


Рис. 2.7. Блок-схема алгоритма открытия папки

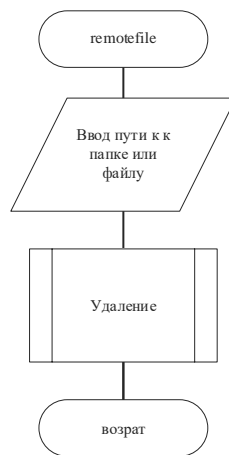


Рис. 2.8. Блок-схема алгоритма удаления папки

3. С использованием языка C/C++ и WinApi описанные в п.2. алгоритмы позволили создать консольное приложение. Разработанный код описан в листингах 1, 2, 3.

Листинг 1: Вызов функций

```

#include "keyfoo.h"
#define _CRT_SECURE_NO_WARNINGS
int main()
{
    setlocale(LC_ALL, "");
    char path[100] = { 0 }, file[100];
    while (true)
    {
        int c = _getch();
        std::cout << (char)c << keyfoo::charToBinary(c) << ' ';
        switch (keyfoo::charToBinary(c))
        {
            case 1110011:
                system("explorer");//открыть проводник
                break;

            case 1110111:
                keyfoo::chdir();//открыть папку по указанию ее пути
                break;

            case 1100110:
                keyfoo::filecreate();//создать файл с текстом
                break;
            case 1101011:
                keyfoo::removefile();//удалить папку по ее адресу и файл в текущей папке
                break;
            case 1100101:
                {
                    //keyfoo::GetProcessList();
                    HWND hwnd;
                    hwnd = FindWindow("CabinetWClass", NULL);//Ищем проводник <br>
                    //std::cout << hwnd;
                    if (hwnd!=NULL)//Проводник открыт!
                    {
                        //SendMessage(hwnd,WM_DESTROY,NULL,NULL);
                        //SendMessage(hwnd,WM_NCDESTROY,NULL,NULL);
                        SendMessage(hwnd, WM_SYSCOMMAND, SC_CLOSE, 0);
                    }
                }break;
        }
    }
}

```

```
#pragma once
#include <windows.h>
#include <string>
#include <fstream>
#include <stdio.h>
#include <direct.h> //удаление файлов и папок
#include <shellapi.h> //чтобы открыть указанную папку ShellExecute
#include <TlHelp32.h>
#include <stdio.h>
#include <iostream>
#include <conio.h>

namespace keyfoo {
void chdir(); //открытие папки по указанному пути
void filecreate();
void removefile(); // Удалить существующий файл, общая функция
int charToBinary(unsigned char val); }
```

```
#define _CRT_SECURE_NO_WARNINGS
#include "keyfoo.h"

namespace keyfoo
{
    /*******создание файла в определенной папке*****/
    void filecreate()
    {
        const int MAX = 2000; //максимальная длина строки
        char str[MAX], f_name[MAX_PATH]; //строка , имя файла
        //changeornot();
        cout << setw(5) << "Имя файла: ";
        cin >> f_name;

        ofstream fout(f_name); // создаём объект класса ofstream для записи
        cin.get(str, MAX, '/'); //пока не будет введен конечный символ будет ввод текста
        продолжен
        fout << str; // запись строки в файл
        fout.close(); // закрываем файл
    }
    /*******функция удаления файла в определенной папке*****/
    void removefile() //added 02/09/2019 21:37
    {
        char ans[4], fname[MAX_PATH], dirname[MAX_PATH];
        //changeornot();
        do
        {
            cout << setw(5) << "Удаляется directory (d) или file (f)?";
            cin >> ans;
        } while ((strcmp(ans, "d") > 0) && (strcmp(ans, "f") > 0));
        cin.clear();
        if ((strcmp(ans, "f") == 0))
        {
            cout << "Файл для удаления: ";
            cin >> fname;
            if (remove(fname))
            {
                cout << "Ошибка удаления файла" << endl;
                cout << "error" << strerror(errno) << endl;
            }
            else
                cout << "Файл удален" << endl;
        }
        else
        {

```

```

        if ((strcmp(ans, "d") == 0))
        {
            cout << "Папка для удаления: ";
            cin >> dirname;
            if (_rmdir(dirname) == -1) //10/12/2019 15:38 не убирать эту строку,
иначе не удаляет файл
            {
                cout << "Ошибка удаления папки" << endl;
                cout << "error" << strerror(errno) << endl;
            }
            else
                cout << "Папка удалена" << endl;
        }
    }
}

/*****функция смены диска*****/
void chdir()
{
    TCHAR sPath[MAX_PATH], sPathTo2[MAX_PATH];
    cout << setw(5) << "Введите путь: ";
    cin >> sPathTo2;
    __try
    {
        ShellExecute(NULL, "open", sPathTo2, NULL, NULL, SW_SHOW); //работает
    }
    __except (filter(GetExceptionCode(), GetExceptionInformation()))
    {
        puts("in except");
    }
}

int charToBinary(unsigned char val)
{
    int binary = 0;

    for (int i = 7; i >= 0; --i)
    {
        binary *= 10;

        if (val & (1 << i))
            binary += 1;
    }

    return binary;
}

/*****/
int filter(unsigned int code, struct _EXCEPTION_POINTERS *ep)
{
    puts("in filter.");
    if (code == EXCEPTION_ACCESS_VIOLATION)
    {
        puts("caught AV as expected.");
        return EXCEPTION_EXECUTE_HANDLER;
    }
    else
    {
        puts("didn't catch AV, unexpected.");
        return EXCEPTION_CONTINUE_SEARCH;
    }
}

```

4. Созданное приложение было протестировано через поочередное нажатие на указанные в варианте задания клавиши. Так при нажатии на клавишу s отображается её скан-код в виде последовательности 1 и 0, где первый бит означает, что клавиша с соответствующим символом была нажата.

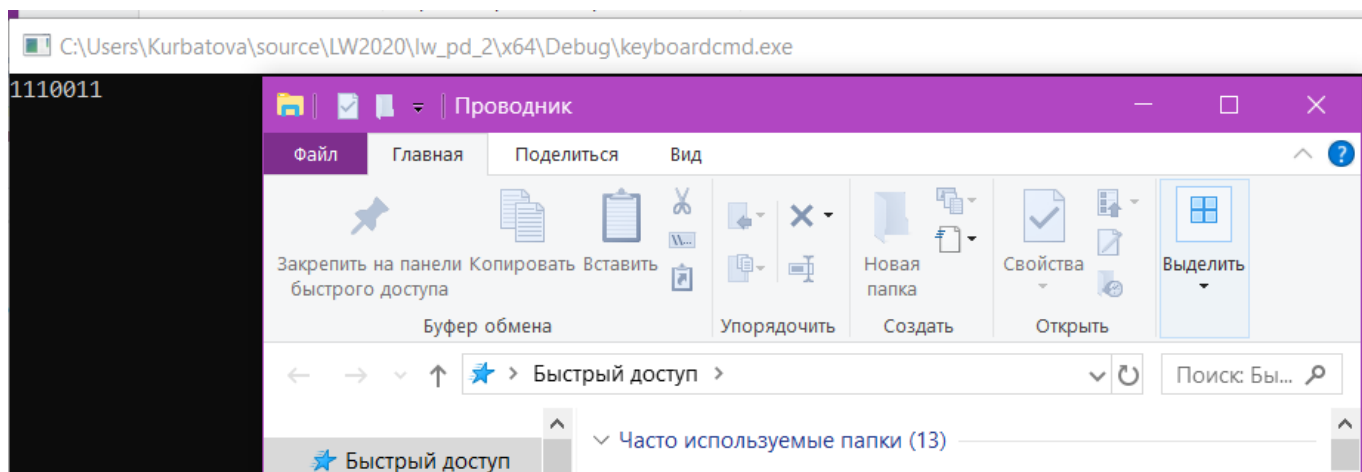


Рис. 2.9. Открытие проводника при нажатии клавиши: s

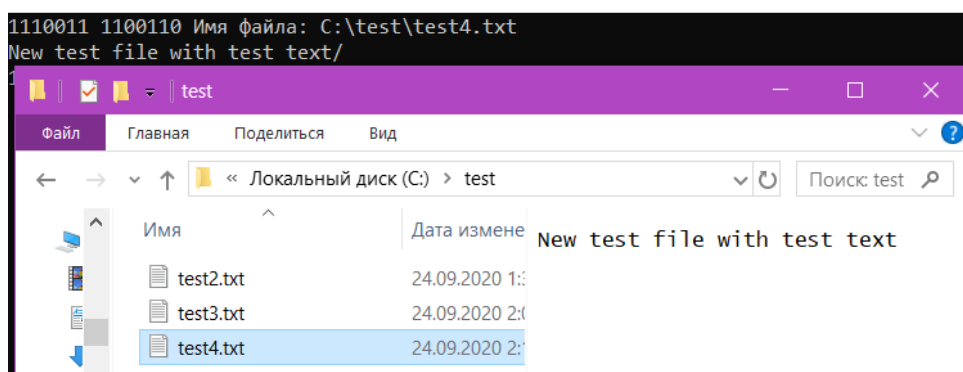


Рис. 2.10. Создание файла

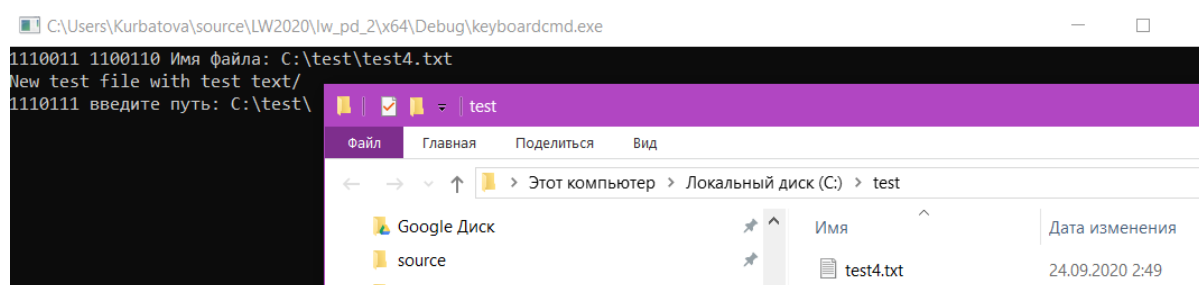


Рис. 2.11. Открытие папки по указанному пути

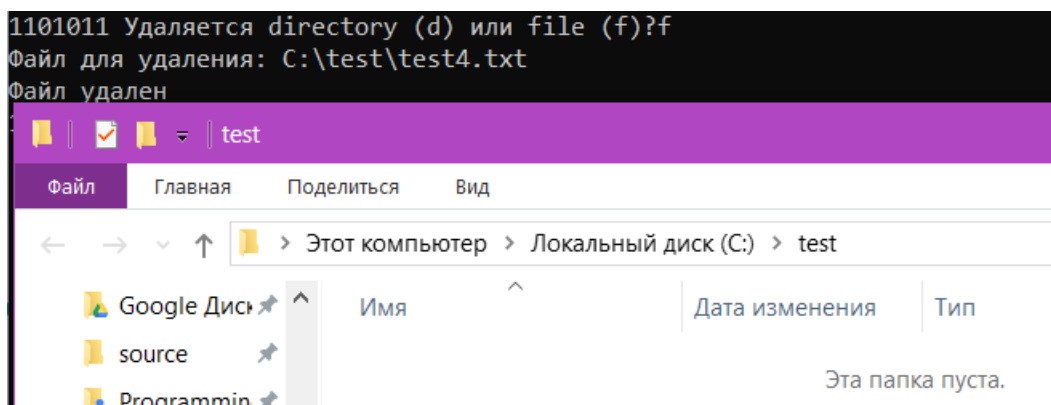


Рис. 2.12. Удаление файла

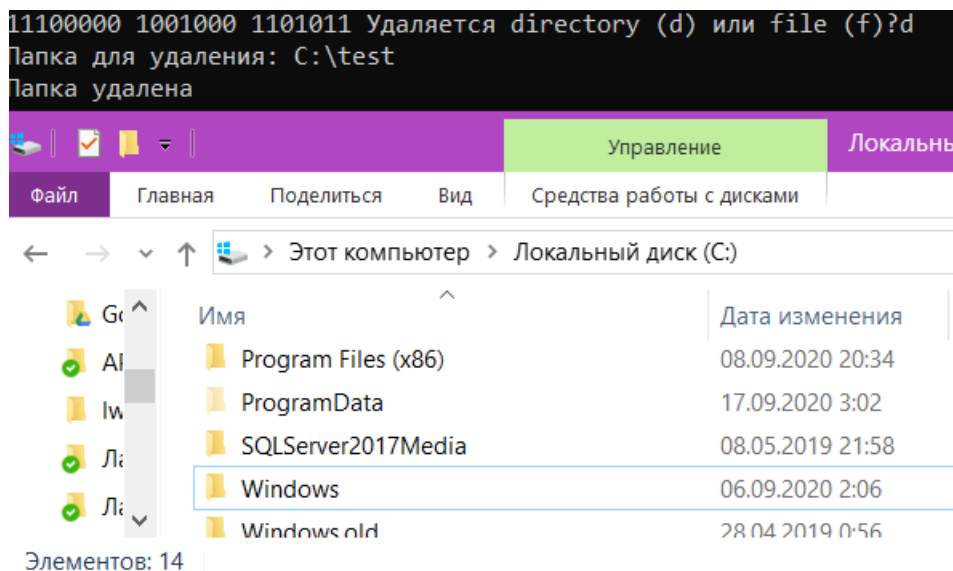


Рис. 2.13. Удаление папки

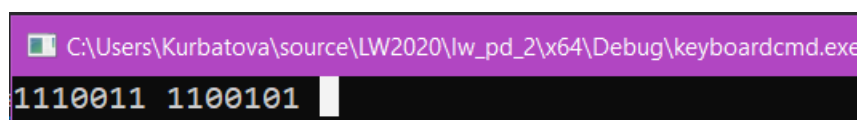


Рис. 2.14. Открытие и закрытие окна Проводника

Таким образом, можно наблюдать следующие двоичные коды для нажимаемых клавиш:

e: 1100101
f: 1100110
w: 1110111
k: 1101011
s: 1110011

Вывод: В лабораторной работе были освоены методы работы с интерактивными устройствами ввода информации. В ходе выполнения работы была создана программа с помощью которой можно открывать окно программы Проводник, создавать и удалять файлы и папки, завершать работу с Проводником. Управление программой происходит с устройства ввода – клавиатуры. При этом, если пользователь нажимает иные клавиши, кроме тех, что могут запустить выполнение функции, программа выводит на экран двоичные коды вводимых с клавиатуры символов. Каждой клавише соответствует определенный скан-код, являющийся условным обозначением клавиши. Клавиатура передает их, и уже далее в программе они ставятся в соответствие таблице расширенных символов ASCII.

Таким образом, было выяснено, что при создании программ можно определить выполнение функции при нажатии на определенную клавишу. Это во многом может облегчить работу с запущенным программным средством.