

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра вычислительной техники**

**ОТЧЕТ**  
**по лабораторной работе №4**  
**по дисциплине «Организация ЭВМ и систем»**  
**Тема: Клавиатура IBM PC. Использование прерываний**

|                    |       |                 |
|--------------------|-------|-----------------|
| Студентка гр. 1361 | _____ | Горбунова Д. А. |
| Студент гр.1361    | _____ | Голубев Д. В.   |
| Преподаватель      | _____ | Гречухин М. Н.  |

Санкт-Петербург  
2022

## **Цель работы**

Изучение возможностей работы с клавиатурой, ознакомление со стандартными средствами библиотеки C++ и средствами системы прерываний DOS и BIOS, обслуживающими клавиатуру.

## **Формулировка задания**

1. Разработать, написать и отладить программу управления перемещением символа (например, "\*") в пределах заданного на экране окна. Для управления использовать клавиши из набора: «стрелка вверх» (СтВВ), «стрелка вниз» (СтВН), «стрелка вправо» (СтВП), «стрелка влево» (СтВЛ) или функциональные клавиши F1 - F12 (варианты см. в таблице 4.2). Для ввода использовать стандартные функции языка C++. Сохранить отлаженную программу.

2. Изменить программу, заменив стандартные функции библиотеки C++ своими. Для написания функций используйте заданное прерывание (см. таблицу), если его возможностей достаточно. Если его возможностей недостаточно, то замените его по своему усмотрению. Сохраните отлаженную программу.

## **Теоретическая информация**

Краткие сведения о подсистеме ввода информации с клавиатуры, используемых прерываниях, буфере клавиатуры и функциях обслуживания ввода с клавиатуры.

Подавляющее большинство программ выполняют ввод информации с клавиатуры. Ввод информации в компьютер может быть выполнен на трех уровнях: обращением к функциям MS-DOS; обращением к функциям BIOS; физическим доступом к аппаратным средствам.

Ввод информации на уровне MS-DOS позволяет "пропустить" клавиатурный ввод через устанавливаемые драйверы, обеспечивает отслеживание нажатия комбинации клавиш Ctrl-C (Ctrl-Break), стандартную для MS-DOS обработку ошибок.

Доступ к клавиатуре на уровне BIOS позволяет программе отслеживать нажатие всех, а не только символьных клавиш, выполнять управление аппаратурой клавиатуры и пр. Интерфейсом Turbo C с BIOS является функция bioskey().

Непосредственный доступ к буферу клавиатуры резко повышает производительность программы. В некоторых случаях необходима имитация нажатий клавиш клавиатуры с записью кодов непосредственно в буфер. При этом физически нажатия клавиш не происходят. Так строятся многие демонстрационные программы, которые открывают или закрывают окна меню, выполняют необходимый выбор, показывают работу программы в "автоматическом" режиме и т.п. На том же самом принципе имитации нажатий клавиш построены программы, способные переносить одним нажатием клавиши целые куски текста из одной программы в любой текстовый редактор. Примером такой программы является входящая в Turbo C резидентная Help-система THELP.COM.

```
int getch (void)
```

Выполняет ввод с клавиатуры через функцию MS-DOS AH=07h. Она не выполняет "эхо" вывода на экран. В этой связи полезна для организации интерфейса с пользователем, при котором нажатие той или иной клавиши вызывает немедленную реакцию программы без отображения введенного символа на экране.

```
int getche (void)
```

Выполняет не буферизуемый ввод с клавиатуры через функцию MS-DOS AH=07h, но в отличие от предыдущей функции обеспечивает вывод введенного символа на экран. Перевод строки происходит при достижении правой вертикальной границы текущего активного окна.

```
char *getpass(char * prompt)
```

Выводит на экран ASCII-строку, на начало которой указывает prompt, а затем принимает с клавиатуры без "эха" строку символов. Вводимые символы (не более 7) помещаются во внутреннюю статическую память. Функция

возвращает указатель на внутреннюю статическую строку, переопределяемую каждым новым обращением к функции. Основное назначение данной функции - ввод паролей в программе без отображения их на экран.

`int kbhit (void)`

Проверяет, пуст ли буфер клавиатуры. Если в буфере есть символы, функция возвращает ненулевое значение, в противном случае она возвращает 0. Использует функцию 0Bh MS-DOS. Является удобным средством предотвращения "зацикливания" при ожидании невозможного в данный момент события. Кроме того, при выполнении функции 0Bh осуществляется проверка нажатия комбинации клавиш Ctrl-Break, что позволяет выполнить аварийное завершение программы.

Интерфейсом программ в персональном компьютере с клавиатурой является прерывание 16h BIOS. Далее приводится описание его функций.

АН = 00h - чтение с ожиданием двухбайтового кода из буфера клавиатуры. Прочитанный код возвращается в регистре AX: младший байт - в регистре AL, старший - в AH. Если нажата ASCII-клавиша, в AL помещается ASCII-код символа, в AH - скэн-код. При нажатии специальных клавиш AL равен 0, а в AH возвращается расширенный скэн-код.

АН = 01h - чтение без ожидания двухбайтового кода из буфера клавиатуры. Если буфер пуст, в 1 выставляется флаг нуля ZF. В противном случае в AX возвращается двухбайтовый код из буфера клавиатуры, но продвижение указателя "головы" буфера не производится, т.е. код "остается" в буфере.

АН = 02h - определение состояния шифт- и триггерных клавиш. В регистре AL возвращается содержимое байта по адресу 40:17h (см. табл. 4.1).

Функция АН = 05h не имеет аналогов в библиотеке Turbo C и может использоваться для имитации нажатия клавиш в демонстрационных программах, программах переноса текста и т.д.

Функции АН = 10 - 12h являются аналогами функций 00 - 02h, но предназначены для использования в компьютерах с клавиатурой 101 /102 клавиши.

Функции АН = 00 - 02h прерывания 16h BIOS положены в основу функции bioskey() библиотеки Turbo C. Далее следует описание этой функции.

`int bioskey(int cmd)`

Обращается в зависимости от значения в cmd к функциям АН = 00 - 02h прерывания 16h. Возвращаемое функцией значение повторяет значение регистра АХ при выходе из прерывания.

### **Результат работы программы**

Программа выводит на экран следующее. Окно с координатами 15, 5, 65, 25 и элемент, который стоит перемещать (рисунок 1). Так же программа выводит реализацию перемещения, данного элемента вправо и влево пошагово.



Рисунок 1 — Результат работы программы

## Приложение 1. Блок-схема реализуемого кода

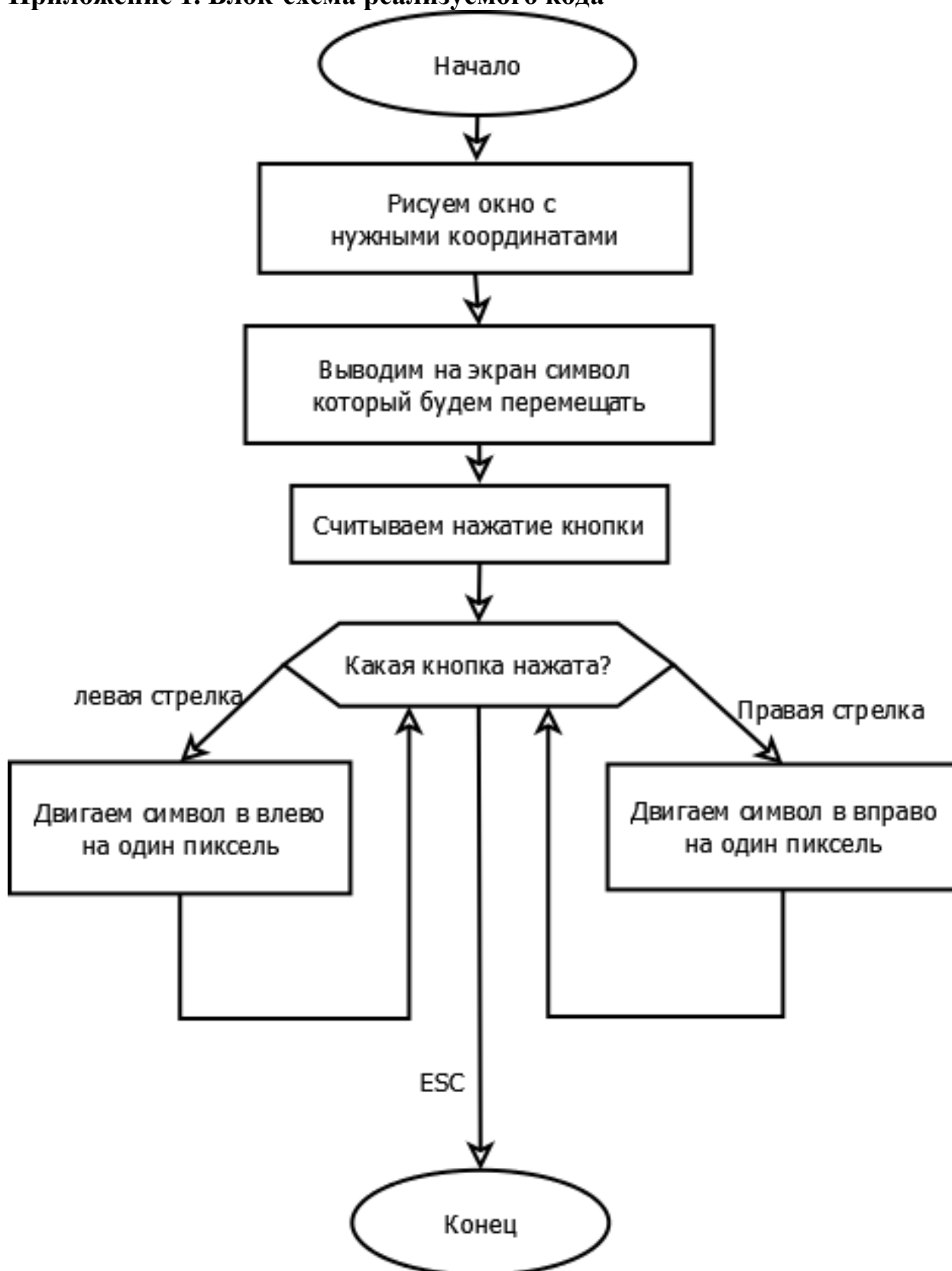


Рисунок 2 – Блок-схема кода.

## Приложение 2. Структурная схема аппаратных средств

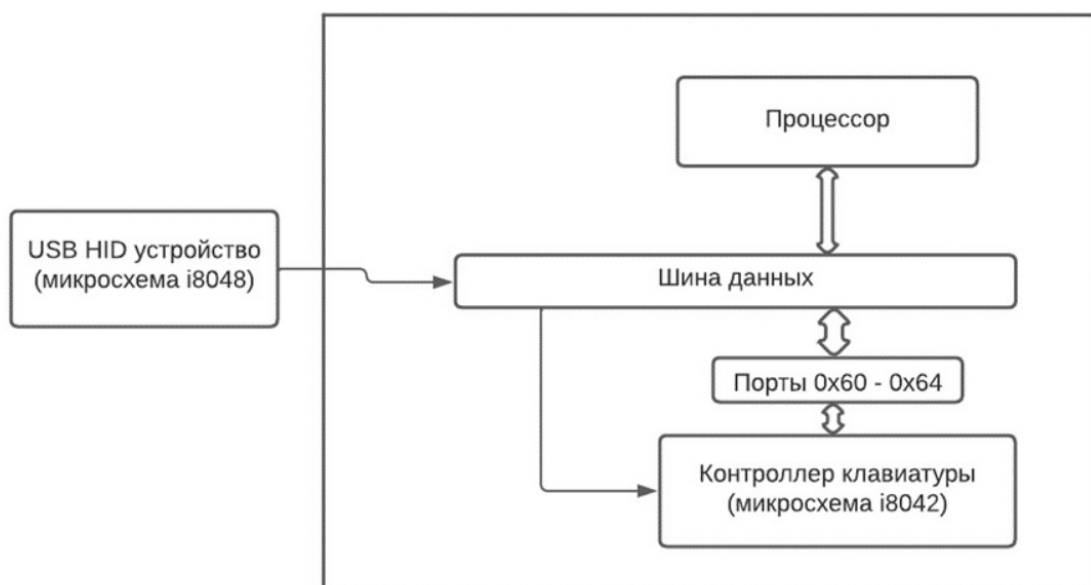


Рисунок 4 – Структурная схема аппаратных средств



### Приложение 3. Исходный код программы

#### Код 1. (С использованием различных функций)

```
#include <conio.h>
#include <dos.h>
void main()
{
    textbackground(10);
    clrscr();
    window(15, 5, 65, 15);
    textbackground(2);
    clrscr();
    textcolor(0);
    char c;
    int x = 1, y = 1;
    gotoxy(x, y);
    cprintf("*");
    _setcursortype (_NOCURSOR);
    do
    {
        c = getch();
        if ( ((c == 75) || (c == 77)) )
        {
            switch (c)
            {
                case 75: if(x > 1) {x--;} break;
                case 77: if(x <= 50) {x++;} break;
            }
            clrscr();
            gotoxy(x, y);
            cprintf("*");
        }
    } while (c != 27);
}
```

#### Код 2. (Через прерывания)

```
#include <conio.h>
#include <dos.h>

int getkey();

int main() {
    int ch, x = 1, y = 1;
```

```

textbackground(8);
clrscr();
window(15, 5, 65, 15);
textbackground(2);
textcolor(0);
_setcursortype (_NOCURSOR);
do {
    clrscr();
    gotoxy(x, y);
    cprintf("*");
    ch = getkey();
    switch (ch)
    {
    case 77: //->
        if (x < 50)
            x++;
        else if (x == 50 && y <= 10)
        {
            y++;
            x = 1;
        }
        break;
    case 75: //<-
        if (x > 1)
            x--;
        else if (x == 1 && y > 1)
        {
            y--;
            x = 50;
        }
        break;
    default: break;
    }
} while (ch != 27);
return 0;
}

int getkey()
{
    union REGS regs;
    int c = 0, e = 0, res;
    regs.h.ah = 0x7;
    int86(0x21, &regs, &regs);
    c = regs.h.al;
    if (!c) {

```

```
        int86(0x21, &regs, &regs);
        e = regs.h.al;
    }
    if (c)
        res = c;
    else
        res = e;
    return res;
}
```