**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. Использования прерываний

|  |  |  |
| --- | --- | --- |
| Студент гр. 2372 |  | Васильев Ю.А. |
| Преподаватель |  | Гречухин М. Н. |

Санкт-Петербург

2023

**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 пе¬ремещением 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 Fl - 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 не достаточно, то замените его по своему усмотрению. Сохраните отлаженную программу.

3. Отлаженные программы предъявить преподавателю.

**Теоретическая информация**

int getch (void)

Выполняет ввод с клавиатуры через функцию MS-DOS АН=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 небуферизуемый 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, a 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 АХ: младший байт - в регистре AL, старший - в АН. Если нажата ASCII-клавиша, в AL помещается ASCII-код символа, в АН - скэн-код. При нажатии специальных клавиш AL равен 0, а в АН возвращается расширенный скэн-код.

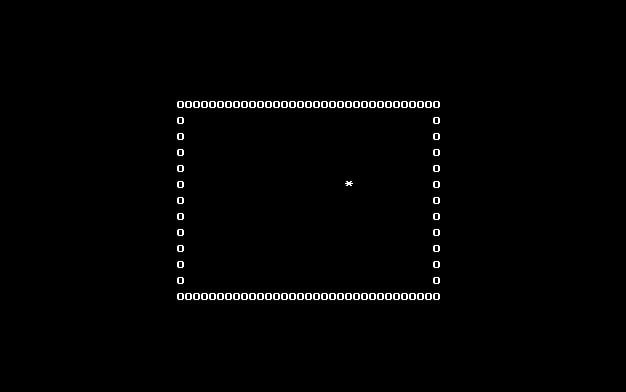
АН = 0lh - чтение без ожидания двухбайтового кода из буфера клавиатуры. Если буфер пуст, в 1 выставляется флаг нуля ZF. В противном случае в АХ возвращается двухбайтовый код из буфера клавиатуры, но продвижение указателя "головы" буфера не производится, т.е. код "остается" в буфере.

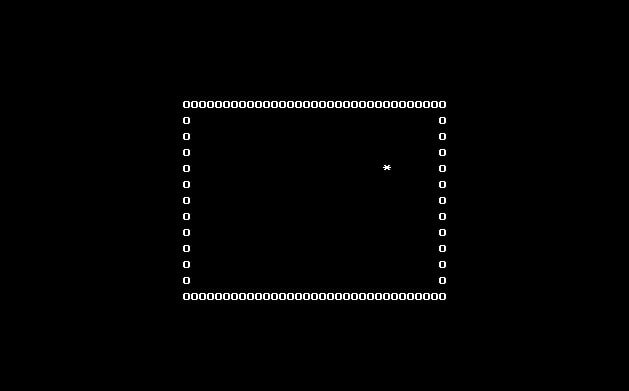
АН = 02h - определение состояния шифт- и триггерных клавиш. В регистре AL возвращается содержимое байта по адресу 40:17h (см. табл. 4.1).

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

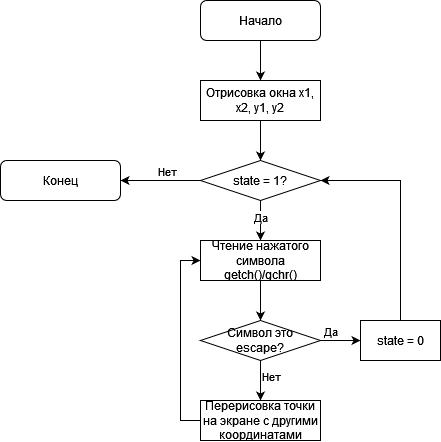
**Результат работы программы**

Программа выводит квадрат в котором можно двигаться с помощью стрелок на клавиатуре.

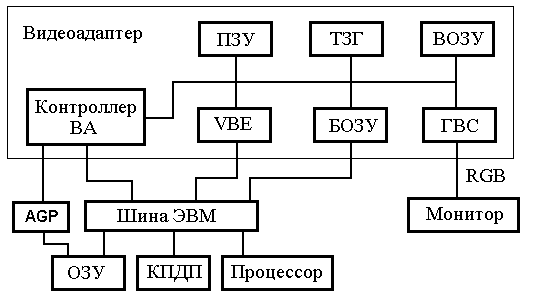




**Блок-схема**



**Структурная схема аппаратных средств**



**Текст программы**

#include <stdio.h>

#include <dos.h>

#include <conio.h>

const int x1 = 25;

const int y1 = 8;

const int x2 = 55;

const int y2 = 18;

void drawWindow() {

    window (x1-1, y1-1, x2+1, y2+2);

    textbackground(BLACK);

    clrscr();

    textcolor(WHITE);

    for (int i = 0; i < x2 - x1 + 3; i++)

        cprintf("o");

    for(int j = 0; j < (y2 - y1 + 1); j++)

    {

        cprintf("o");

        for(int k = 0; k < (x2 - x1 + 1); k++)

            cprintf(" ");

        cprintf("o");

    };

    for (int p = 0; p < x2 - x1 + 3; p++)

        cprintf("o");

    window(x1,y1,x2,y2);

    textbackground(BLACK);

    clrscr();

    textcolor(WHITE);

}

// My function for getch().

int gchr() {

    union REGS r;

    r.h.ah = 0x00;

    int86(0x16, &r, &r);

    return r.h.ah == 0x00 ? r.h.al : r.h.ah;

}

int move(int& x, int& y, int& state, int freplace = 0)

{

    if (kbhit()) {

        int k = freplace ? gchr() : getch();

        switch (k) {

            case 77:

                x++;

                break;

            case 75:

                x--;

                break;

            case 72:

                y--;

                break;

            case 80:

                y++;

                break;

            case 1:

            case 27:

                state = 0;

                break;

        }

        return 1;

    }

    return 0;

}

void main() {

    clrscr();

    textbackground(BLACK);

    int state = 1;

    int x = (x2 - x1) / 2;

    int y = (y2 - y1) / 2;

    drawWindow();

    \_setcursortype(\_NOCURSOR);

    gotoxy(x,y);

    cprintf("\*");

    while (state) {

        if (move(x, y, state, 1))

        {

            if (x == 0)

                x = x2 - x1 + 1;

            if (x == x2 - x1 + 2)

                x = 1;

            if (y == 0)

                y = y2 - y1 + 1;

            if (y == y2 - y1 + 2)

                y = 1;

            clrscr();

            gotoxy(x,y);

            cprintf("\*");

        }

    }

}

(При передаче 0 четвёртым аргументом в move используется стандартная функция чтения символа C++, иначе функция с прерываниями)