

Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Лабораторная работа №5  
«Программирование клавиатуры»

Выполнил:  
Студент группы 150501  
Кипятков В. И.

Проверил:  
Преподаватель  
Одинец Д.Н.

Минск, 2023

## 1. Постановка задачи

Программируя клавиатуру помогать ее индикаторами. Алгоритм мигания произвольный. Условия реализации программы, необходимые для выполнения лабораторной работы:

1. Запись байтов команды должна выполняться только после проверки незанятости входного регистра контроллера клавиатуры. Проверка осуществляется считыванием и анализом регистра состояния контроллера клавиатуры.

2. Для каждого байта команды необходимо считывать и анализировать код возврата. В случае считывания кода возврата, требующего повторить передачу байта, необходимо повторно, при необходимости – несколько раз, выполнить передачу байта. При этом повторная передача данных не исключает выполнения всех оставшихся условий.

3. Для определения момента получения кода возврата необходимо использовать аппаратное прерывания от клавиатуры.

Все коды возврата должны быть выведены на экран в шестнадцатеричной форме.

## 2. Алгоритм

Для вывода на экран скан-кодов или кодов возврата необходимо заменить обработчик прерывания 09h. При вызове данного обработчика выводится значение из порта 60h на экран. При управлении индикаторами значение из порта 60h (код возврата) необходимо анализировать на случай необходимости повторной передачи байтов команды.

Для управления индикаторами клавиатуры используется команда *EDh*. Вторым байт этой команды содержит битовую маску для настройки индикаторов (бит 0 – состояние Scroll Lock, бит 1 – состояние Num Lock, бит 2 – состояние Caps Lock). В данной программе управление индикаторами реализовано в функции `void indicator(int mask)`, где `mask` – битовая маска, определяющая состояние индикаторов.

Перед каждой командой записи происходит ожидание освобождения входного буфера клавиатуры: `while((inp(0x64) & 2) != 0);`

## 3. Листинг программы

Далее приведен листинг программы, реализующей все поставленные задачи.

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

const unsigned char QUIT_CHAR = 0x01;
const unsigned char HIGHLIGHT_CHAR = 0xa3;
const int TARGET_INTERRUPT = 9;
const int true = 1;
const int false = 0;

void interrupt NewInterrupt(void);
void interrupt (*oldInterrupt)(void);
```

```

void SaveOldInterrupt() { oldInterrupt = getvect(TARGET_INTERRUPT); }
void SetNewInterrupt() { setvect(TARGET_INTERRUPT, NewInterrupt); }
void RestoreOldInterrupt() { setvect(TARGET_INTERRUPT, oldInterrupt); }
void UpdateHighlightFlag(unsigned char);
void UpdateQuitFlag(unsigned char);
void WaitInputFree();
void SetMask(unsigned char mask);
void Highlight (void);
void EndInterrupt() { outp(0x20, 0x20); } // 0001 0100

int commandIsExecuted;
int quitFlag;
int needHighlight;

void main()
{
    commandIsExecuted = false;
    quitFlag = false;
    needHighlight = false;

    clrscr();

    SaveOldInterrupt();
    SetNewInterrupt();

    while(!quitFlag) {
        if (needHighlight) {
            Highlight();
            needHighlight = false;
        }

        RestoreOldInterrupt();
        clrscr();
        return;
    }

    void interrupt NewInterrupt()
    {
        unsigned char value = 0;
        oldInterrupt();
        value = inp(0x60);
        UpdateQuitFlag(value);
        UpdateHighlightFlag(value);
        commandIsExecuted = commandIsExecuted || (needHighlight == false) || (value == 0xFA);

        printf("\t%x", value);
        EndInterrupt();
    }

    void UpdateQuitFlag(unsigned char value)
    {
        if (value == QUIT_CHAR) {
            quitFlag = true;
        }
    }

    void UpdateHighlightFlag(unsigned char value)
    {
        if (value != HIGHLIGHT_CHAR) {
            return;
        }
        if (needHighlight) {
            needHighlight = false;
        } else {
            needHighlight = true;
        }
    }
}

```

```

void SetMask(unsigned char mask)
{
    commandIsExecuted = false;

    while (!commandIsExecuted)
    {
        WaitInputFree();
        outp(0x60, 0xED);
        outp(0x60, mask);
        printf("\tset mask %d", mask);
        //delay(500);
    }
}

void WaitInputFree()
{
    while((inp(0x64) & 0x02) != 0x00);
}

void Highlight ()
{
    printf("\tBegin highlight");
    SetMask(0x01); // 00000001
    delay(1000);
    SetMask(0x00); // 00000000
    delay(1000);
    SetMask(0x02); // 00000010
    delay(1000);
    SetMask(0x00); // 00000000
    delay(1000);
    SetMask(0x04); // 00000100
    delay(1000);
    SetMask(0x00); // 00000000
    delay(1000);
    printf("\tEnd highlight");
}

```

## 4. Тестирование программы

Во время работы программы происходит мигание индикаторов клавиатуры.

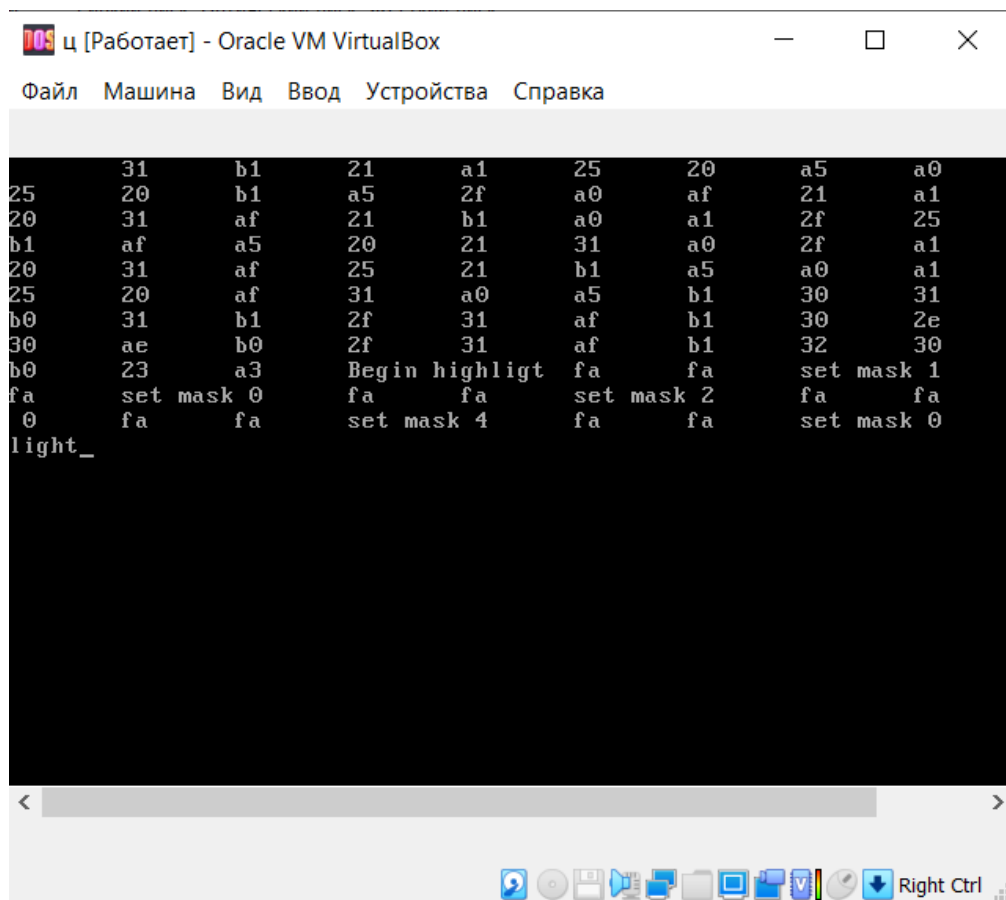


Рисунок 4.1. — Вывод скан-кода при нажатии на клавишу.

## 5. Заключение

В данной лабораторной работе были выполнены все поставленные задачи: программа выводит на экран скан-коды клавиш при их нажатии и отпускании, также реализовано мигание индикаторов пока не будет нажата клавиша Esc.

Программа компилировалась в Turbo C++ и запускалась в DOS, который эмулировался с помощью VirtualBox.