Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Лабораторная работа №5
«Программирование клавиатуры»
Вариант 14

Выполнил:
Студент группы 150504
Желубовский С.В.

Проверил:
Преподаватель
Одинец Д.Н.

Минск, 2023

1. Постановка задачи

Программируя клавиатуру помигать 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 считывание 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*. Второй 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 set_mask(byte 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 <stdio.h>
#include <stdlib.h>

#include <dos.h>

#define SEC 1000
#define SUCCESS 0xFA
#define KEYBOARD_INTERRUPT 0x09
#define KEYBOARD_LIGHTS_CODE 0xED
```

```c
#define NONE 0x00
#define SCROLL_LOCK 0x01
#define NUM_LOCK 0x02
#define CAPS_LOCK 0x04

typedef unsigned char byte;

int command_succeeded = 0;

void interrupt (*old_handler)(void);

void interrupt new_handler(void) {
  byte scan_code = inp(0x60);
  printf("%X\n", scan_code);

  command_succeeded = (scan_code == SUCCESS);

  old_handler();
}

void set_mask(byte mask) {
  int i = 0;

  while (!command_succeeded) {
    // Wait until buffer is empty
    while ((inp(0x64) & 0x02));
    outp(0x60, KEYBOARD_LIGHTS_CODE);

    // Wait until buffer is empty
    while ((inp(0x64) & 0x02));
    outp(0x60, mask);

    if (++i == 3) {
      fputs("Failed to set mask 3 times in a row\n", stderr);
      setvect(KEYBOARD_INTERRUPT, old_handler);
      exit(EXIT_FAILURE);
    }
  }
  command_succeeded = 0;
}

int main(void) {
  old_handler = getvect(KEYBOARD_INTERRUPT);
  setvect(KEYBOARD_INTERRUPT, new_handler);

  set_mask(SCROLL_LOCK);
  delay(SEC);
  set_mask(NUM_LOCK);
  delay(SEC);
  set_mask(CAPS_LOCK);
  delay(SEC);
  set_mask(NONE);
  delay(SEC);
  set_mask(SCROLL_LOCK | NUM_LOCK | CAPS_LOCK);
  delay(SEC);
  set_mask(NONE);

  setvect(KEYBOARD_INTERRUPT, old_handler);
  return EXIT_SUCCESS;
}
```

4. Тестирование программы

Во время работы программы происходит мигание индикаторов клавиатуры.

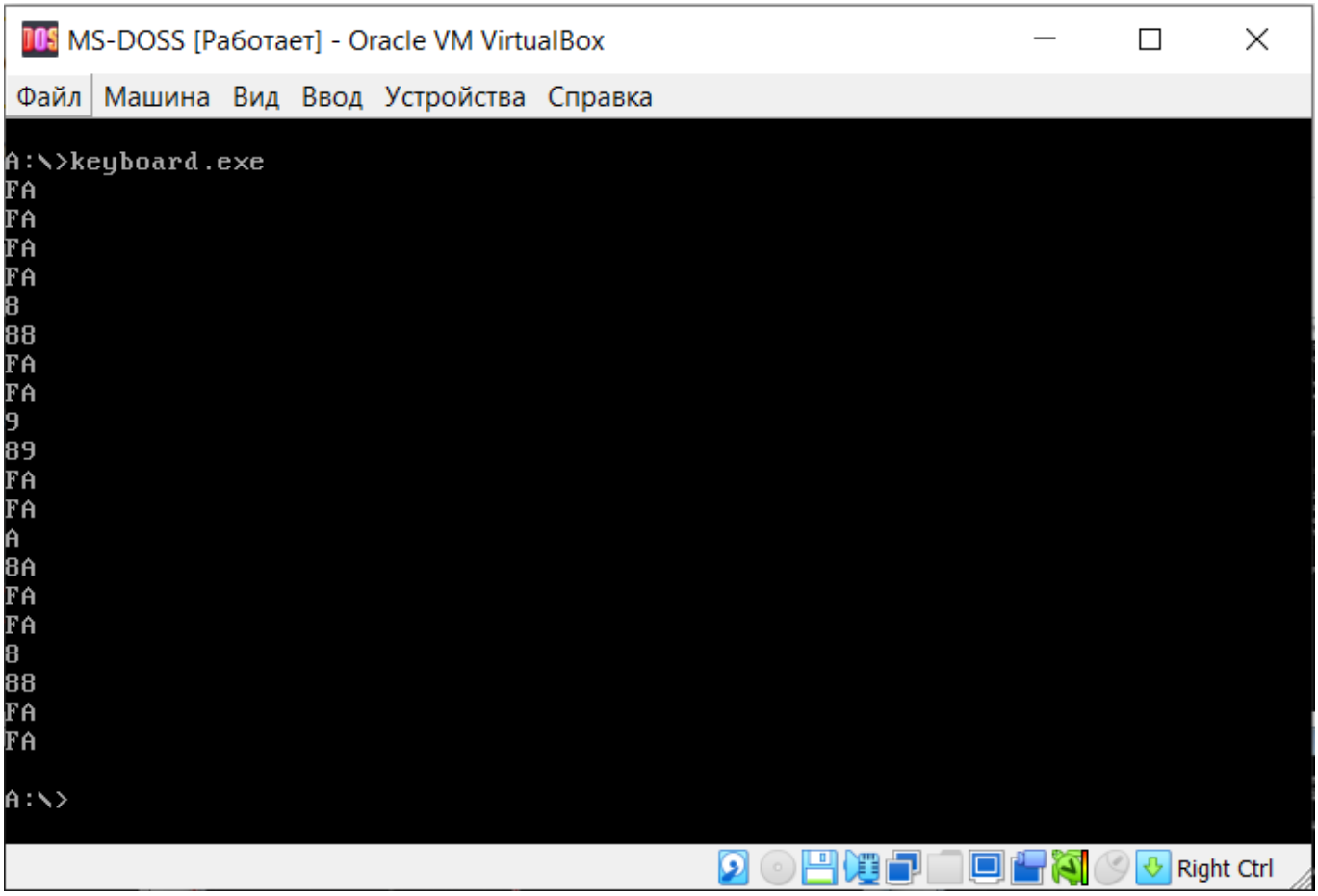

Рисунок 4.1. — Вывод скан-кода при нажатии на клавишу.

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.