

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ЭВМ

Отчет по лабораторной работе №5
«Программирование клавиатуры»

Проверил:
к.т.н., доцент
Одинец Д.Н.

Выполнил:
студент гр. 150501
Почебут А.С.

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 прерывание 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 байтом 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.

3. Листинг программы

Далее приведен листинг программы, реализующей все поставленные задачи.

```
#include <stdio.h>
#include <stdlib.h>

#include <dos.h>

#define DELAY_TIME          1000
#define SUCCESS              0xFA
#define KEYBOARD_INTERRUPT  0x09
#define KEYBOARD_LIGHTS_CODE 0xED
#define EMPTY_BUFFER        0x02 // 0000 0010

#define KEYBOARD_STATE_REGISTER 0x64
#define KEYBOARD_DATA_REGISTER  0x60
```

```

#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(KEYBOARD_DATA_REGISTER);
    printf("Scan code: %X\n", scan_code);
    // FA - succeed
    // FE - error

    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(KEYBOARD_STATE_REGISTER) & EMPTY_BUFFER)); // 0000 0010
        outp(KEYBOARD_DATA_REGISTER, KEYBOARD_LIGHTS_CODE);

        // Wait until buffer is empty
        while ((inp(KEYBOARD_STATE_REGISTER) & EMPTY_BUFFER));
        outp(KEYBOARD_DATA_REGISTER, mask);
        // 0000 caps num scroll

        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;
}

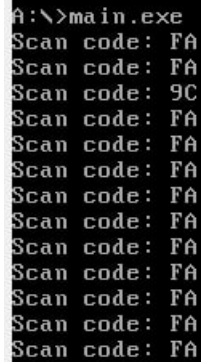
int main(void) {
    old_handler = getvect(KEYBOARD_INTERRUPT);
    setvect(KEYBOARD_INTERRUPT, new_handler);

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

    setvect(KEYBOARD_INTERRUPT, old_handler);
    return EXIT_SUCCESS;
}

```

4. Тестирование программ



```
A:\>main.exe  
Scan code: FA  
Scan code: FA  
Scan code: 9C  
Scan code: FA  
Scan code: FA  
Scan code: FA  
Scan code: FA  
Scan code: FA  
Scan code: FA  
Scan code: FA  
Scan code: FA  
Scan code: FA  
Scan code: FA  
Scan code: FA
```

Рисунок 4.1 – Результат работы программы при запуске.

5. Заключение

В данной лабораторной работе были выполнены все поставленные задачи: написана программа, которая мигает индикаторами.

Программа компилировалась в Borland C и запускалась в DOS, который эмулировался с помощью MS-DOS в VirtualBox.