5Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Лабораторная работа №05

«Программирование клавиатуры»

Выполнил: Проверил:

Студент группы 250503 Преподаватель

Патюпин М.С. Одинец Д.Н.

МИНСК 2024

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.

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

1. Алгоритм
2. Сохраняем указатель на старый обработчик
3. Меняем его на новый
4. Пока не установлен флаг выхода
5. Если установлен флаг мигания индикаторов

мигаем индикаторами

1. Восстанавливаем старый обработчик прерывания
2. Листинг программы

Далее приведен листинг программы, реализующей все поставленные задачи.

#include <dos.h>

#include <conio.h>

#include <stdio.h>

void interrupt newInt9(void); // Функция обработки прерывания

void interrupt(\*oldInt9)(void); // Указатель на обработчик прерывания

void indicator(unsigned char mask); // Функция управления индикаторами

void blinking(void); // Функция мигания индикаторами

int isResend = 1;// Флаг ошибки / необходимости повторной передачи данных

int quitFlag = 0; // Флаг выхода из программы

int blinkingON = 0; // Флаг мигания индикаторами

int count = 0; // Счетчик повторной передачи

void main() {

clrscr();

disable();

oldInt9 = getvect(9); // Сохраняем указатель на старый обработчик

setvect(9, newInt9); // Меняем его на новый

enable();

while (!quitFlag) { // Пока не установлен флаг выхода

if (blinkingON) // Если установлен флаг мигания индикаторов

blinking(); // мигаем индикаторами

}

disable();

setvect(9, oldInt9); // Восстанавливаем старый обработчик прерывания

enable();

return;

}

// Функция обработки прерывания

void interrupt newInt9() {

unsigned char value = 0;

value = inp(0x60); // Получаем значение из порта 60h

if (value == 0x01) quitFlag = 1; // Устанавливаем флаг выхода, если нажата Esc

if (value == 0x10 && blinkingON == 0) // Поставить или снять флаг мигания, если нажата клавиша Q

blinkingON = 1;

else if (value == 0x10 && blinkingON == 1)

blinkingON = 0;

if (value == 0xFE && blinkingON == 1) { // Если нет подтверждения успешного выполнения команды

isResend = 1; // Устанавливаем флаг повторной передачи байта

count++;

if (count > 3) {

printf("\n\nError!\n");

quitFlag = 1;

outp(0x20, 0x20); // Сброс контроллера прерываний

return;

}

}

else isResend = 0;

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

outp(0x20, 0x20); // Сброс контроллера прерываний

}

// Функция управления индикаторами

void indicator(unsigned char mask) {

isResend = 1;

while (isResend) { // Пока нет подтверждения успешного выполнения команды

while ((inp(0x64) & 0x02) != 0x00); // Ожидаем освобождения входного буфера клавиатуры

outp(0x60, 0xED); // Записываем в порт команду управления индикаторами

delay(50);

}

isResend = 1;

while (isResend) { // Пока нет подтверждения успешного выполнения команды

while ((inp(0x64) & 0x02) != 0x00); // Ожидаем освобождения входного буфера клавиатуры

outp(0x60, mask); // Записываем в порт битовую маску для настройки индикаторов

delay(50);

}

}

// Функция мигания индикаторами

void blinking() {

indicator(0x01); // вкл. индикатор Scroll Lock

delay(200);

indicator(0x04); // вкл. индикатор Caps Lock

delay(200);

indicator(0x02); // вкл. индикатор Num Lock

delay(200);

indicator(0x00); // выкл. все индикаторы

delay(200);

indicator(0x07); // вкл. индикаторы Scroll Lock, Num Lock и Caps Lock

delay(200);

indicator(0x00); // выкл. все индикаторы

delay(200);

indicator(0x06); // вкл. индикаторы Num Lock и Caps Lock

delay(200);

indicator(0x00); // выкл. все индикаторы

delay(200);

indicator(0x05); // вкл. индикаторы Scroll Lock и Caps Lock

delay(200);

indicator(0x00); // выкл. все индикаторы

delay(200);

indicator(0x03); // вкл. индикаторы Scroll Lock и Num Lock

delay(200);

indicator(0x00); // выкл. все индикаторы

delay(200);

indicator(0x01); // вкл. индикатор Scroll Lock

delay(200);

indicator(0x04); // вкл. индикатор Caps Lock

delay(200);

indicator(0x02); // вкл. индикатор Num Lock

delay(200);

indicator(0x00); // выкл. все индикаторы

delay(200);

indicator(0x07); // вкл. индикаторы Scroll Lock, Num Lock и Caps Lock

delay(200);

indicator(0x00); // выкл. все индикаторы

delay(200);

}

1. Тестирование программы

Во время работы программы поочередно загораются и тухнут световые индикаторы на клавиатуре.

1. Заключение

В ходе лабораторной работы удалось запрограммировать клавиатуру.

Программа компилировалась в Turbo C++ и запускалась в VirtualBox.