Министерство образования Республики Беларусь

Учреждение образования Белорусский государственный университет

информатики и радиоэлектроники

# Кафедра ЭВМ

### Отчет по лабораторной работе № 6

"Программирование клавиатуры"

Выполнил:

студент группы 950502

Киреев Ю.В.

Проверил:

Одинец Д. Н.

#### Минск 2021

1. **Задание**

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.
5. **Теоретические сведения**

Для работы с клавиатурой используется 2 регистра: 60h – регистр данных, 64h – регистр состояния (статуса). Для управления индикаторами через 60h отправляется код EDh. Затем маска, в соответствии с которой должны загореться индикаторы.

Формат «маски»:

Биты 7-3 не используются Caps Lock Num Lock Scroll Lock

Так клавиатура работает медленно, запись байтов команды должна выполняться только после проверки незанятости входного регистра контроллера клавиатуры. Проверка

осуществляется считыванием и анализом регистра состояния контроллера клавиатуры (64h, бит 1). На обработку каждого байта клавиатура отвечает кодом возврата. Если в регистре 60h находится код

FA – байт обработан успешно,

FE – произошла ошибка.

В случае ошибки передачу байта нужно повторить. Пересылка выполняется до 3 раз, если ошибка не исчезла, нужно вывести сообщение и выйти из программы. При нажатии клавиши блок клавиатуры передает ее код сканирования центральному процессору. Когда клавиша отпускается, клавиатура снова передает ее код, но увеличенный на 128 (или шестнадцатеричное значение 80). То есть, коды для нажатия и отпускания клавиш различаются. Когда выполняется какое-либо действие с клавишей (нажатие или отпускание), процессор клавиатуры обнаруживает его и запоминает в буфере. Затем формируется прерывание с номером 9.

1. **Листинг программы**

#include <dos.h>

#include <conio.h>

#include <stdio.h>

const unsigned char QUIT\_CHAR = 0x81;

const int TARGET\_INTERRUPT = 9;

const int true = 1;

const int false = 0;

union REGS i, o;

int check\_alt();

void call\_int\_16(unsigned char, unsigned char );

void interrupt NewInterrupt(void);

void interrupt (\*oldInterrupt)(void);

void SaveOldInterrupt();

void SetNewInterrupt();

void RestoreOldInterrupt();

void WaitInputFree();

void keyboard\_input(void);

void EndInterrupt();

int commandIsExecuted;

int quitFlag;

int keyboard\_input\_flag;

void main()

{

unsigned far \*fp;

FP\_SEG(fp) = \_psp;

FP\_OFF(fp) = 0x2c;

\_dos\_freemem(\*fp);

commandIsExecuted = false;

quitFlag = 0;

keyboard\_input\_flag = false;

clrscr();

SaveOldInterrupt();

SetNewInterrupt();

while(!quitFlag)

{

if (keyboard\_input\_flag)

{

char buf[10];

//keyboard\_input();

delay(1000);

scanf("%s", buf);

printf("%s", buf);

if (buf[0] = 27)

break;

keyboard\_input\_flag = 0;

}

}

//RestoreOldInterrupt();

clrscr();

\_dos\_keep(0, (\_DS - \_CS) + (\_SP / 16) + 1);

return;

}

void SaveOldInterrupt()

{

oldInterrupt = getvect(TARGET\_INTERRUPT);

}

void SetNewInterrupt()

{

setvect(TARGET\_INTERRUPT, NewInterrupt);

}

void RestoreOldInterrupt()

{

setvect(TARGET\_INTERRUPT, oldInterrupt);

}

void interrupt NewInterrupt()

{

unsigned char value = 0;

oldInterrupt();

value = inp(0x60);

if (value == QUIT\_CHAR)

{

quitFlag = 1;

}

if (value == 0x02+0x80)

{

if (check\_alt())

{

keyboard\_input\_flag = 1;

keyboard\_input();

}

}

printf("%x ", value);

EndInterrupt();

}

int check\_alt() //проверка на нажатие alt

{

i.h.ah = 0x12;//для 102 клавиш

int86(0x16, &i, &o);

if (o.h.al == 0x8)

return 1;

else

return 0;

}

void call\_int\_16(unsigned char scan , unsigned char ASCII)

{

i.h.ah = 0x5;

i.h.ch = scan;

i.h.cl = ASCII;

int86(0x16, &i, &o);

}

void EndInterrupt()

{

outp(0x20, 0x20);

}

void WaitInputFree()

{

while((inp(0x64) & 0x02) != 0x00);

}

void keyboard\_input()

{

call\_int\_16(0x10, 'q');

delay(1000);

call\_int\_16(0x11, 'w');

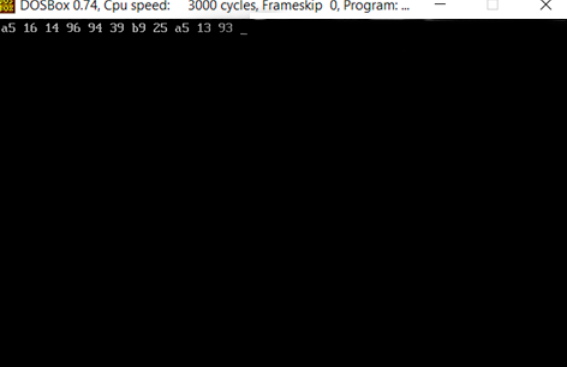
delay(1000);

call\_int\_16(0x12, 'e');

delay(1000);

call\_int\_16(0, ' ');

}



**Вывод:** в ходе лабораторной работы было выполнено программирование клавиатуры.