Министерство образования Республики Беларусь

Учреждение образования Белорусский государственный университет информатики и радиоэлектроники

Кафедра ЭВМ

Отчёт по лабораторной работе №3

“Программирования контроллера прерываний”

Вариант 6

Проверил: Выполнил:

к.т.н., доцент студент гр.150504

Одинец Дмитрий Николаевич Горбачевский К.В.

Минск 2020

**Задача**

Написать резидентную программу выполняющую перенос всех векторов аппаратных прерываний ведущего и ведомого контроллера на пользовательские прерывания. При этом необходимо написать обработчики аппаратных прерываний, которые будут установлены на используемые пользовательские прерывания и будут выполнять следующие функции:

1. Выводить на экран в двоичной форме следующие регистры контроллеров прерывания (как ведущего, так и ведомого):

* регистр запросов на прерывания;
* регистр обслуживаемых прерываний;
* регистр масок.

При этом значения регистров должны выводиться всегда в одно и то же место экрана.

1. Осуществлять переход на стандартные обработчики аппаратных прерываний, для обеспечения нормальной работы компьютера.

**Алгоритм**

* Все векторы аппаратных прерываний ведущего и ведомого контроллера переносятся на пользовательские прерывания с помощью функций getvect и setvect.
* Производится инициализация контроллеров, заключающаяся в последовательности команд: ICW1, ICW2, ICW3 и ICW4.
* С помощью функции \_dos\_keep осуществляется выход в DOS, при этом программа остаётся резидентной.
* В каждом обработчике выводятся в видеопамять в двоичной форме значения регистров запросов на прерывания, обслуживаемых прерываний, масок. Затем вызываются стандартные обработчики прерываний.

**Листинг программы**

data segment

startProgram db 'Start$'

enterSymbol db 'Enter symbol: $'

output db 'Output: $'

data ends

code segment

newLine:

mov ah, 02h

mov dl, 0ah

int 21h

mov ah, 02h

mov dl, 0dh

int 21h

ret

printFromCOM:

MOV DX, 01h ; dx = 1 (com2)

MOV AH, 02H ; read letter from channel

INT 14H ; al - symbol

xor dx, dx

MOV DL,AL ; al - data byte

MOV AH,02H ; print letter

INT 21H

ret

writeToCOM:

MOV DX, 0 ; port number com1

MOV AH,01H ; send letter to channel

INT 14H

ret

start:

mov ax, data

mov ds, ax

mov ah, 9h

mov dx, offset startProgram

int 21h

call newLine

call newLine

XOR AX,AX

MOV AL,11100011B ; port init

xor dx,dx ; dx = 0 (com1)

MOV AH,00H

INT 14H

mov ah, 9h

mov dx, offset enterSymbol

int 21h

xor ax, ax

MOV AH, 01H ; read a character from console

INT 21H ; in al - your symbol

call writeToCOM ; write al into com

MOV AL, 'e'

call writeToCOM ; write al into com

MOV AL, 'l'

call writeToCOM ; write al into com

MOV AL, 'l'

call writeToCOM ; write al into com

MOV AL, '0'

call writeToCOM ; write al into com

call newLine

call newLine

mov ah, 9h

mov dx, offset output

int 21h

xor ax, ax

MOV AL, 11100011B ; port init

MOV DX, 01h ; dx = 1 (com2)

MOV AH, 00H

INT 14H

call printFromCOM

call printFromCOM

call printFromCOM

call printFromCOM

call printFromCOM

MOV AX, 4C00H ; exit

INT 21H

code ends

end start

|  |  |
| --- | --- |
| **Chip - Purpose** | [**I/Oport**](https://wiki.osdev.org/I/O_ports) |
| Master PIC - Command | 0x0020 |
| Master PIC - Data | 0x0021 |
| Slave PIC - Command | 0x00A0 |
| Slave PIC - Data | 0x00A1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Chip** | **Interrupt numbers (IRQ)** | **Vector offset** | **Interrupt Numbers** |
| Master PIC | 0 to 7 | 0x08 | 0x08 to 0x0F |
| Slave PIC | 8 to 15 | 0x70 | 0x70 to 0x77 |

**Заключение**

В данной лабораторной работе разработана программа для работы с последовательным COM портом. Данные заносятся в COM1 и читаются из COM2.