Лабораторная работа №3

**"Подсистема прерываний"**

К теме:Организация подсистемы прерываний. Контроллер прерываний.

# Цель работы

Изучить организацию прерываний в IBM PC с использованием контроллера прерываний.

Продолжительность работы - 4 часа.

# ЗАДАНИЕ

Под MS DOS написать программу, которая:

1) выполняет инициализацию контроллера прерываний;

2) выводит на экран содержимое регистров запросов, обслуживаний и масок для ведущего и ведомого контроллеров (через видеобуфер).

\*При нажатии на клавиши меняется цвет или фон выводимой информации.

Программа должна быть резидентной. Все вектора прерываний переопределяются, новый базовый адрес выбирается в соответствии с вариантом.

# Теоретические сведения

Каждый момент времени центральный процессор может работать только с одним устройством. Циклический опрос каждого устройства с последующей обработкой запроса оказался неэффективным. Решение задачи оказался контроллер прерываний, который принимает запросы от устройств и в соответствии с приоритетом направляет их процессору, если прерывание от данного устройства не замаскировано (разрешено) в регистре масок. Если прерывание разрешено и устройство его запросило, то устанавливается соответствующий устройству бит в регистре запросов.

Контроллер прерываний состоит из двух микросхем, подключенных каскадно (ведущий и ведомый контроллеры), каждая из которых имеет по 8 линий прерываний (IRQ0-IRQ7, IRQ8-IRQ15). За каждой линией закреплено определенное устройство.

Когда процессор получает запрос на прерывание, он сохраняет свое текущее состояние и переключается на выполнение запрошенной операции. При этом устанавливается бит в регистре обслуживания (бит запроса сбрасывается). После обслуживания прерываний сбрасывается бит обслуживания, посылается сигнал EOI (endofinterrupt), процессор переключается на выполняемую ранее задачу.

Для доступа к контроллеру прерываний используются порты 20h и 21h (для ведущего), A0 иA1h (для ведомого).

Регистр масок доступен через порт 21h /A1h. Чтобы изменить определенный бит, нужно считать значение из этого регистра, изменить нужный бит, записать значение обратно.

Чтобы считать регистр запросов, его нужно выбрать (записать в 20h/A0h значение 0Ah), а затем считать содержимое из порта 20h/A0h.

Чтобы считать регистр обслуживания, его нужно выбрать (записать в 20h/A0h значение 0Bh), а затем считать содержимое из порта 20h/A0h.

Для резидентной программы понадобится следующий фрагмент кода:

unsigned far \*fp; //объявляем указатель

FP\_SEG (fp) = \_psp; // получаем сегмент

FP\_OFF (fp) = 0x2c; // и смещение сегмента данных с переменными среды

\_dos\_freemem(\*fp); //чтобы его освободить

\_dos\_keep(0,(\_DS -\_CS)+(\_SP/16)+1);//оставляем резидентной, указывая //первым параметром код завершения, а //вторым - объем памяти, который должен //быть зарезервирован для программы //после ее завершения

# ВАРИАНТЫ заданий

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Номер варианта** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| Базовый вектор | 60Н /  08H | 68Н /  08H | 70Н /  08H | 78Н /  08H | 80Н /  08H | 88Н /  08H | 90Н /  08H | 98Н /  08H | A0Н /  08H | A8Н /  08H |
| **Номер варианта** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| Базовый вектор | B0Н /  08H | B8Н /  08H | C0Н /  08H | C8Н /  08H | D0Н /  08H | D8Н /  08H | 08H / 60H | 08H / 68Н | 08H / 70Н | 08H / 78Н |
| **Номер варианта** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** |
| Базовый вектор | 08H / 80Н | 08H / 88Н | 08H / 90Н | 08H / 98Н | 08H / A0Н | 08H / A8Н | 08H / B0Н | 08H / B8Н | 08H / C0Н | 08H / C8Н |

# Вопросы к защите

1. Для чего в программе обработчика прерываний необходимо указывать команду EOI?
2. Для чего используются команды CLI и STI?
3. Объяснить понятие «вектор прерывания».
4. Что делает команда IRET?
5. Что такое «вложенное прерывание»?
6. Назначение таблицы векторов прерываний.

# Список рекомендуемой литературы

1. В. Несвижский "Программирование аппаратных средств в Windows", с. 495 (486).

\_102153768.pdf