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?

Как уже указывалось, при передаче прерывания на обработку контроллер прерывания сбрасывает соответствующий бит регистра IRR и устанавливает бит в ISR. Последний блокирует все прерывания с таким же и более низким приоритетом, поэтому при окончании обработки прерывания он должен быть сброшен. Это выполняется с помощью команды окончания обработки прерывания (End of Interrupt, EOI).

1. Для чего используются команды CLI и STI? В системе команд [x86](https://ru.wikipedia.org/wiki/X86) совместимых [процессоров](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80) инструкция **CLI** — сокращение от «Clear Interrupt-Enable Flag». Она сбрасывает interrupt flag (IF) в регистре [EFLAGS](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D0%B8%D1%81%D1%82%D1%80_%D1%84%D0%BB%D0%B0%D0%B3%D0%BE%D0%B2). Когда этот флаг сброшен процессор игнорирует все прерывания (кроме NMI) от внешних устройств.

В системе команд [x86](https://ru.wikipedia.org/wiki/X86) совместимых процессоров, инструкция **STI** — сокращение от «Set Interrupt-Enable Flag». Она устанавливает флаг interrupt flag (IF) в регистре [EFLAGS](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D0%B8%D1%81%D1%82%D1%80_%D1%84%D0%BB%D0%B0%D0%B3%D0%BE%D0%B2), что разрешает процессору обрабатывать асинхронные прерывания от внешних устройств.

1. Объяснить понятие «вектор прерывания».//адрес по которому будет сделан переход в случаи прерывания
2. Что делает команда IRET?//возвращает из прерывания (16-битном размере операнда) в прерванную программу
3. Что такое «вложенное прерывание»? Ситуация, когда в процессор поступает высокоприоритетное прерывание, которое должно быть обслужено до завершения обработки текущего
4. Назначение таблицы векторов прерываний. служит для определения корректного ответа на [прерывания](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D1%80%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) и [исключения](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B8%D1%81%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%BD%D0%B8%D0%B9). В **IDT** используются следующие типы прерываний: аппаратные прерывания, программные прерывания и прерывания, зарезервированные процессором, называемые исключениями (первые 32) на случай возникновения некоторых событий ([деление на ноль](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BD%D0%B0_%D0%BD%D0%BE%D0%BB%D1%8C), ошибка трассировки, переполнение). Первый этап инициализации выполняется [BIOS](https://ru.wikipedia.org/wiki/BIOS), перед загрузкой [ОС](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0). Второй непосредственно самой операционной системой. Операционной системе доступно изменение некоторых адресов прерываний.

# Список рекомендуемой литературы

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

\_102153768.pdf