

Прерывания.

Существуют аппаратные и программные прерывания. Программные прерывания вызываются командой **INT**.

Аппаратные прерывания делятся на внутренние и внешние. внутренние прерывания инициализируются процессором (деление на ноль, пошаговый режим). Внешние прерывания приходят от периферийных устройств (сигналы от таймера, принтера, клавиатуры).

Действия процессора по обслуживанию прерывания не зависят от типа прерывания.

Начальные адреса ОЗУ от **0** до **3Fh** отведены под векторы прерываний. Вектором прерывания называется 4-х байтовая область, в которой хранится адрес программы обработки прерывания. Два старших байта содержат адрес сегмента, два младших – эффективный адрес обработчика прерывания. Векторы и соответствующие им прерывания имеют номера. Вектор с номером **0** располагается, начиная с адреса **0**, вектор **1** – с адреса **4** и т.д. Всего существует **256** векторов прерываний.

Большая часть векторов прерываний предназначена для выполнения определенных действий и автоматически заполняется адресами системных программ при загрузке системы. Прерывания с номерами от **0** до **1Fh** используются как прерывания внутренней аппаратуры (BIOS), прерывания с номерами от **20h** до **3Fh** используются ОС, остальные прерывания используются внешней аппаратурой, системными драйверами. Прерывания с номерами от **60h** до **66h** свободны и могут использоваться прикладными программами.

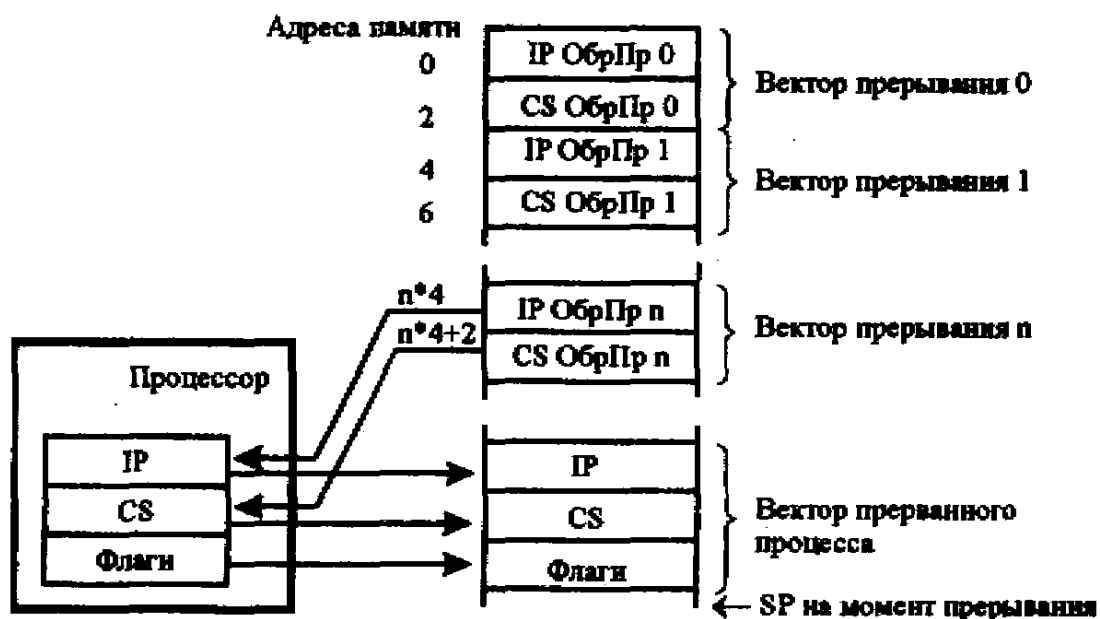
Для того, чтобы прикладной обработчик получил управление, его адрес нужно поместить в соответствующий вектор прерывания.

Получив сигнал прерывания с определенным номером, процессор сохраняет в стеке выполняемой программы текущее содержимое регистра флагов, **CS** и **IP**. Регистры **CS** и **IP** содержат полный адрес возврата в прерванную программу. Затем процессор загружает **CS** и **IP** из соответствующего вектора прерываний и таким образом происходит переход на программу обработчик прерывания.

Обработчик должен заканчиваться командой **IRET**. Эта команда извлекает из стека адрес возврата, загружая его в регистры **CS** и **IP** и регистр флагов.

Для того, чтобы прикладной обработчик получил управление, его адрес нужно поместить в соответствующий вектор прерывания.

Схема обслуживания прерывания.



Команды для работы с прерываниями.

1. Вызов процедуры обработки прерывания

Call to Interrupt Procedure

INT номер_прерывания
И8

Алгоритм работы:

Push flags

if=0 (*запретить внешние прерывания*)

tf=0 (* запретить пошаговый режим*)

Push cs

Push ip

cs:= 0:[номер_прерывания * 4 +2]

ip:= 0:[номер_прерывания * 4]

Описание:

Вызывает процедуру обработки прерывания. Непосредственный операнд в диапазоне от 0 до 255 задает индекс элемента в таблице векторов прерывания.

Пример

вызов обработчика аппаратного прерывания 08h:

int 08h

2. Вызов процедуры обработки прерывания, если установлен флаг переполнения

Call to Interrupt Procedure if Overflow

INTO

Алгоритм работы:

Push flags

if=0 (*запретить внешние прерывания*)

tf=0 (* запретить пошаговый режим*)

Push cs

Push ip

If of=1

cs:= 0:[4 * 4 +2]

ip:= 0:[4 * 4]

Описание:

Вызывает обработчик прерывания с номером 4, если установлен флаг переполнения. Используется после арифметических команд.

3. Возврат из прерывания

Interrupt RETurn

IRET

Алгоритм работы:

ip:=Pop

cs:=Pop

flags:=Pop

Описание:

Считывает три верхние записи стека в регистры: **ip, cs, flags**.