**Прерывания**

**Прерывание** – инициируемый определенным образом процесс, временно переключающий процессор на выполнение другой программы с последующим возобновлением выполнения прерванной программы

**Обработчик прерывания** (ISR, Interrupt Service Routine) – программа, которой передается управление при возникновении прерывания.

**Прерывания:**

* Аппаратные;
* Программные;

**Intel:**

* Предусмотрено 256 номеров прерываний, из них 0-31 (0 – 1F) зарезервированы, остальные доступны для программистов.

**IDT**

* IDT – Interrupt Descriptor Table
  + Находится по адресу 0000:000
  + Представляет собой массив полных адресов обработчиков прерываний
    - 2 байта – смещение;
    - 2 байта – сегмент
  + Индекс в массиве – номер перывания.
  + В real mode – Interrupt Vector Table

**Программные прерывания**

* Могут возникать:
  + При выполнении недопустимой команды:
    - Деление на 0;
    - Несуществующая инструкция;
    - И тд;
  + При использовании программной команды **int.**

**Команда int**

Int 99h

SS

SP

|  |
| --- |
|  |
|  |
| RetIP |
| RetCS |
| oldFlags |
|  |

**Возврат из ISR**

* Для возврата управления из обработчика прерывания используется команда **iret**.
  + В отличие от **ret/retf** всегда без операндов

**Аппаратные прерывания возникают при поступлении сигналов от внешних устройств**

* Номер прерывания выбирается в зависимости от источника сигнала
* Выполнение основной программы приостанавливается
* Выполняются те же операции, что и при использовании **int**.
* Процессор гарантирует, что текущая инструкция будет выполнена до передачи управления ISR.
* Если прерывание аппаратное, обработчик, как правила, должен сохранять значения регистров.
* Подмена обработчика позволяет включиться в процесс обработки прерывания.

Перехват прерываний

Что будет, если прерывание произойдет в этот момент?

Плохо

(Неправильный код):

**xor** **ax**, **ax**

**mov** **es**, **ax**

**mov** **cx**, **Segment**

**mov** **dx**, Offset

**mov** **[es**:IntNumber\*4**]**, **cx**

**mov** **[es**:IntNumber\*4 + 2**]**, **dx**

Уже лучше, но все равно неправильно:

**xor** **ax**, **ax**

**mov** **es**, **ax**

**mov** **cx**, **Segment**

**mov** **dx**, Offset

**cli** *; Прерывания не обрабатываются. Требование – держать отключенными прерывания как можно меньше*

А что, если прерывания уже были отключены?!

**mov** **[es**:IntNumber\*4**]**, **cx**

**mov** **[es**:IntNumber\*4 + 2**]**, **dx**

**sti** *; Теперь обрабатывается*

**Теперь норм:**

**xor** **ax**, **ax**

**mov** **es**, **ax**

**mov** **cx**, **Segment**

**mov** **dx**, Offset

**pushf**

**cli** *; Прерывания не обрабатываются. Требование – держать отключенными прерывания как можно меньше*

**mov** **[es**:IntNumber\*4**]**, **cx**

Что будет, если программа завершит выполнение, оставив прерывание перехваченным?

Плохо – поэтому после перехвата надо вернуть все

**mov** **[es**:IntNumber\*4 + 2**]**, **dx**

**popf**

*Флаг IF (Interrupt Flag) – определяет, будет ли процессор обрабатывать исключения*

Есть функции MS-DOS, позволяющие выполнить перехват прерываний

Плохая идея

Во время подмены обработчика А на С, А может подмениться на В

* 35h – возвращает адрес обработчика.
* 25h – заменяет адрес обработчика на заданный (выполняет перехват)

В зависимости от того, какое прерывание обрабатывается, обработчик может:

* Выполнять всю обработку самостоятельно
* Вызывать предыдущий обработчик в процессе обработки (в т.ч. после).