




ПРЕРЫВАНИЯ.



Обработка прерываний в реальном режиме работы микропроцессора



Лекция



кандидата технических наук
Ливак Е.Н.



Понятие прерывания

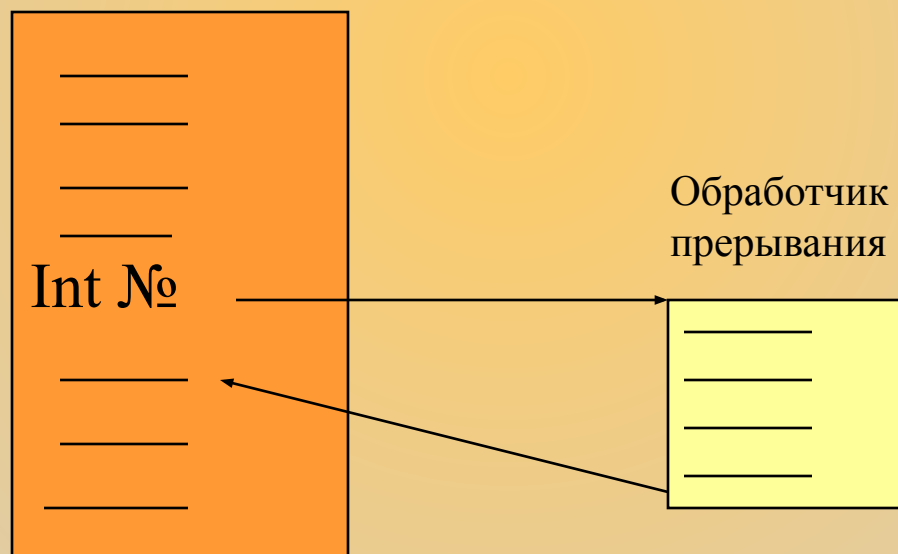


- ★ Прерывание означает временное прекращение основного процесса вычислений для выполнения некоторых запланированных или незапланированных действий, вызываемых работой аппаратуры или программы.
- ★ *Т.е. это процесс, временно переключающий микропроцессор на выполнение другой программы с последующим возвратом к прерванной программе.*



Обработчик прерывания

- ★ Программа обработки прерывания называется обработчиком прерывания.



прерванная программа продолжит выполнение с точки, где было приостановлено ее выполнение (со следующей команды)



Классификация прерываний

(в зависимости от источника прерывания)



аппаратные - возникают как реакция микропроцессора на физический сигнал от некоторого устройства (клавиатура, системные часы, жесткий диск, принтер и т.д.)

- при нажатии клавиши на клавиатуре вызывается аппаратное прерывание, при котором считывается код нажатой клавиши
- при перемещении мыши вызывается аппаратное прерывание, при котором считываются координаты положения мыши



программные - вызываются искусственно с помощью соответствующей команды из программы (int), предназначены для выполнения некоторых действий операционной системы



исключения - являются реакцией микропроцессора на нестандартную ситуацию, возникшую внутри микропроцессора во время выполнения некоторой команды программы (ошибки или исключительные ситуации, которые не могут быть обработаны самой программой, например, деление на 0)



Другая классификация

- ★ **Внешние** - вызываются внешними по отношению к микропроцессору событиями (по существу - это группа аппаратных прерываний)
- ★ **Внутренние** - возникают внутри микропроцессора во время вычислительного процесса (по существу - это исключительные ситуации и программные прерывания)





Виды прерываний



Маскируемое прерывание (IRQ)

аппаратное прерывание, которое можно игнорировать, (путем установки бита в регистре IMR



Немаскируемое прерывание (NMI)

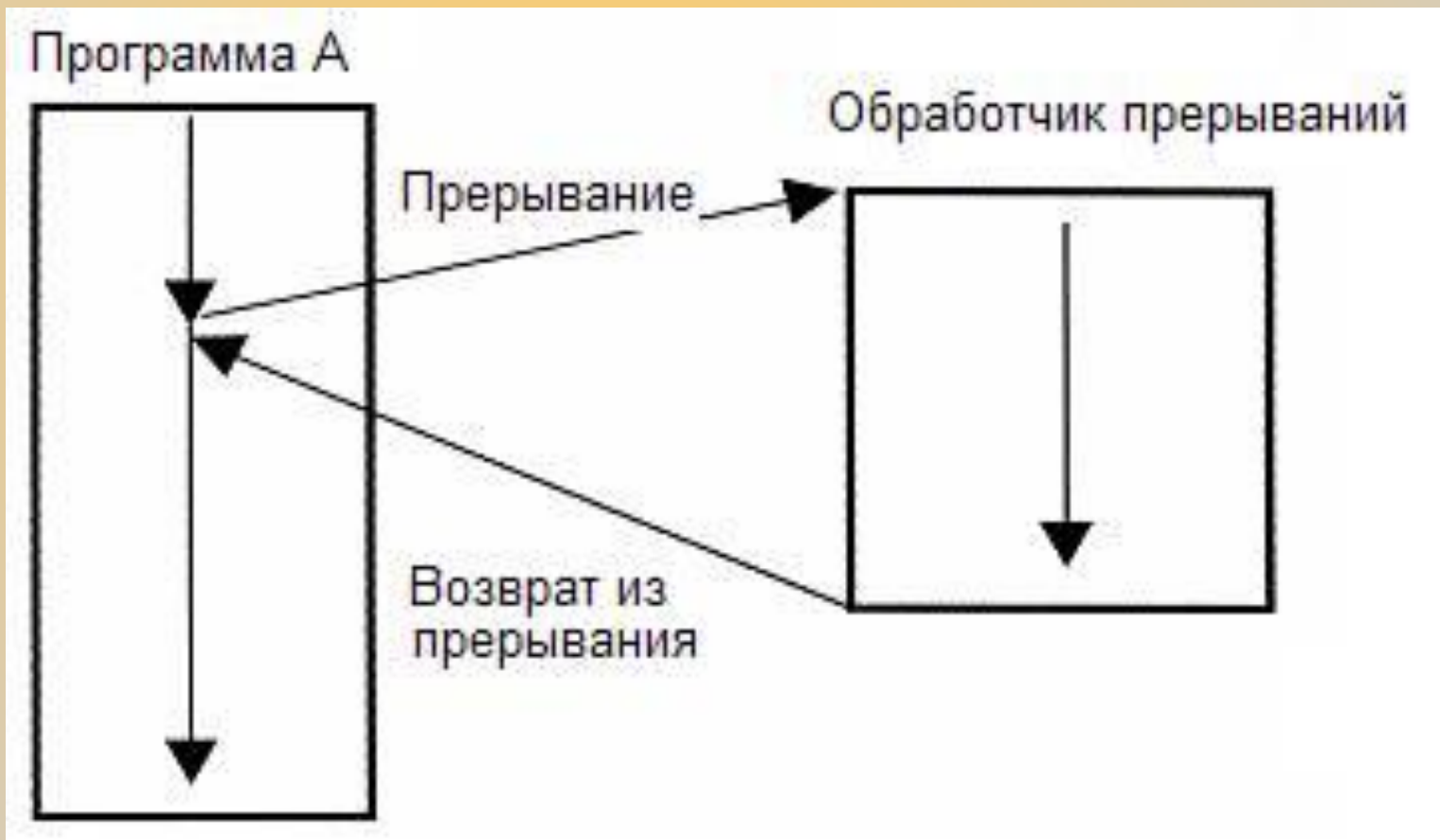
аппаратное прерывание, которое нельзя игнорировать. NMI используются для задач с высшим приоритетом, например таких как таймеры



Межпроцессорное прерывание (IPI): особый случай прерывания, которое генерируется одним процессором для прерывания другого процессора в многопроцессорной системе



Программное прерывание





Программные прерывания



Обработка машинной команды

int N

- это программное прерывание
- int (interrupt – англ. *прерывание*)
- N – номер (имя) прерывания



С помощью программных прерываний программист может обратиться в нужное для него время за обслуживанием своих запросов либо к ОС, либо к BIOS, либо к собственным обработчикам прерываний.





Пример. Прерывание 21h (DOS) Как это работает?

- * 21 – номер (имя) прерывания, h – обозначение 16-ричной системы счисления
- * Прерывание 21h предназначено для предоставления программисту различных услуг со стороны ОС
- * Этими услугами является набор функций

Примеры функций

- * **00h** - завершение работы программы
- * **01h** - ввод символа с ожиданием и эхосопровождением.
 - На выходе из функции в регистре AL будет ASCII-код символа.
- * **02h** - вывод символа
 - На входе в регистр DL надо поместить ASCII-код символа
- * **05h** - вывод символа на принтер
 - На входе в регистр DL надо поместить ASCII-код символа
- * **09h** - вывод строки на экран
 - Пара регистров ds:dx должна содержать адрес строки, строка записана в сегменте данных
- * **0ah** - ввод строки с клавиатуры
- *

Внимание.
Функции запоминать не надо,
это будет изучаться
при программировании
на языке Ассемблер



Использование функций 21h Как это программируется?



- * Поместить номер функции в регистр **ah**
- * Поместить передаваемые функции параметры в определенные регистры (для каждой функции соответствующие, см. документацию)
- * Вызвать прерывание командой `int 21h`
- * Извлечь результаты работы функции из определенных регистров





Пример

Вывод строки на экран

```
message db 'Это строка','$'
```

```
mov ah,09h
```

```
mov dx,offset message
```

```
int 21h
```

09h - вывод строки на экран

На входе

в ds:dx адрес строки

с символом \$ на конце



Пример.

Вывод строк на экран

```
masm
model    small
.stack   100h
.data
message1 db  'Это первая строка', '$'
message2 db  'Это вторая строка $'
.code
start:
    mov  ax,@data
    mov  ds,ax
    mov  ah,09h
    mov  dx,offset message1
    int  21h
    mov  dx,offset message2
    int  21h
    mov  ax,4c00h ; стандартный выход - ah=00h
    int  21h
end start
```





Система прерываний



- ★ Система прерываний - это совокупность программных и аппаратных средств, реализующих механизм прерываний



Механизм прерываний поддерживается на аппаратном уровне




i



Аппаратные средства системы прерываний



1. выводы микропроцессора
(INTR, INTA, NMI)



2. программируемый контроллер прерываний (8259А, микросхема)

предназначен для фиксирования сигналов прерываний от восьми различных внешних устройств;

обычно используют две последовательно соединенные микросхемы, поэтому кол-во возможных источников внешних прерываний до 15 плюс одно немаскируемое прерывание;

именно он формирует номер вектора прерывания и выдает его на шину данных



1. внешние устройства



Программные средства системы прерываний



Таблица векторов прерываний

- Первый килобайт ОП (в реальном режиме 16-разрядных процессоров). Содержит адреса (векторы) обработчиков прерываний.



Два флага в регистре флагов flags/eflags:

- - IF (Interrupt Flag) - флаг прерывания.
 - Предназначен для маскирования (запрещения) аппаратных прерываний.
 - Если IF=1, микропроцессор обрабатывает внешние прерывания,
 - Если IF = 0, то игнорирует
- - TF(Trace Flag) - флаг трассировки.
 - Если TF=1, то микропроцессор переходит в режим покомандной работы.
 - В этом режиме в микропроцессоре генерируется внутреннее прерывание с номером 1



Машинные команды микропроцессора:

- int, into, iret, cli, sti



Таблица векторов прерываний (в реальном режиме)

- * Содержит адреса (векторы) обработчиков прерываний
- * Адреса строго упорядочены в соответствии с номерами прерываний!
- * Состоит из 256 элементов по 4 байта каждый
- * Структура элемента:
 - 2 байта - смещение начала программы-обработчика прерывания от начала кодового сегмента (IP)
 - 2 байта - базовый адрес сегмента, в котором находится программа-обработчик (CS)

**Как определить адрес,
по которому находится вектор прерывания с номером N ?**
Смещение элемента таблицы векторов прерываний = $N * 4$

Полный размер таблицы?
 $4 * 256 = 1024$
Первый килобайт ОП



Обработка прерываний в реальном режиме работы микропроцессора



производится в 3 этапа:

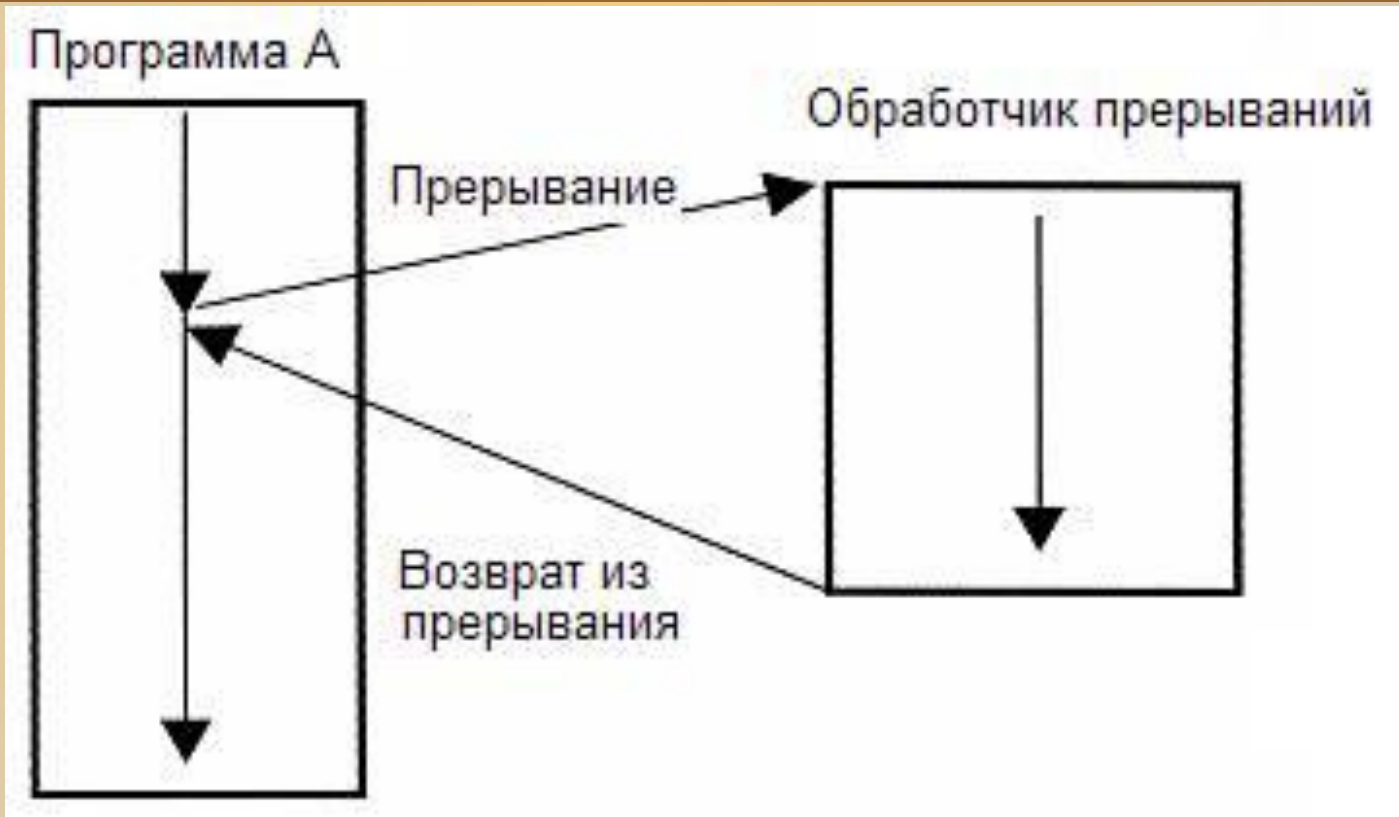


- 1) прекращение выполнения текущей программы;
- 2) переход к выполнению и выполнение программы обработки прерывания;
- 3) возврат управления прерванной программе





Программное прерывание



Адрес следующей
выполнимой команды –
в паре регистров CS:IP



ЭТАП 1. Прекращение выполнения текущей программы

- ★ Должно произойти так, чтобы потом можно было вернуться и продолжить работу
 - ⇒ необходимо сохранить содержимое регистров, так как они являются ресурсами, разделяемыми между программами
- ★ Обязательными для сохранения являются регистры `cs`, `ip`, `flags`
 - пара `CS:IP` содержит адрес команды, с которой необходимо начать выполнение после возврата, `flags` - состояние флагов после выполнения последней команды прерванной программы
- ★ Эти регистры сохраняются микропроцессором автоматически в стеке
- ★ Сохранение остальных регистров - должно обеспечиваться программистом !!!





Сегмент
данных

С
Е
Г
М
Е
Н
Т

К
О
Д
А

Int №

...

CS

IP

FLAGS

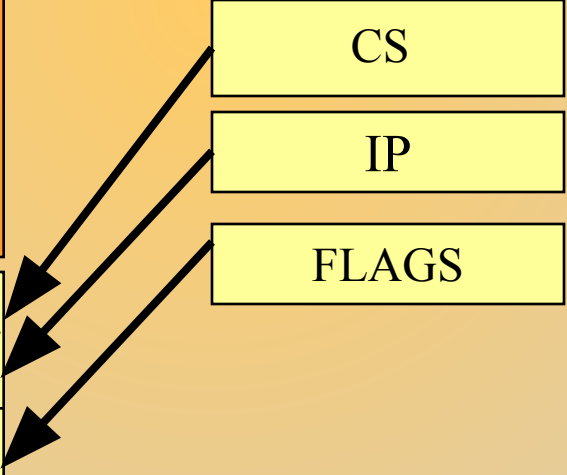
...

Содержимое регистров CS, IP, FLAGS копируется в стек программы

CS

IP

FLAGS



Сегмент стека программы

Вопрос.
А что будет,
если стек полон?
(переполнение стека)



Этап 2. Переход к выполнению и выполнение программы обработки прерывания (обработчика прерывания)



Микропроцессор

- 1) По номеру источника прерывания определяет смещение в таблице векторов прерываний
- 2) Помещает первые два байта в регистр IP
- 3) Помещает вторые два байта в регистр CS
- 4) Передает управление по адресу CS:IP



Далее выполняется сама программа обработки прерывания.





Этап 3.

Возврат управления прерванной программе

★ Необходимо привести стек в состояние, в котором он был сразу после передачи управления обработчику прерывания

- Для этого программист должен указать необходимые действия по восстановлению регистров и очистке стека.

★ Последние команды в обработчике прерывания –
sti
iret

sti - разрешить аппаратные прерывания (устанавливает флаг IF=1, не имеет операндов)

i

iret - извлечь последовательно три слова (3*2 байта) из стека и поместить их соответственно в регистры ip, cs, flags.



Таблица IDT (Interrupt Description Table)

Таблица дескрипторов прерываний

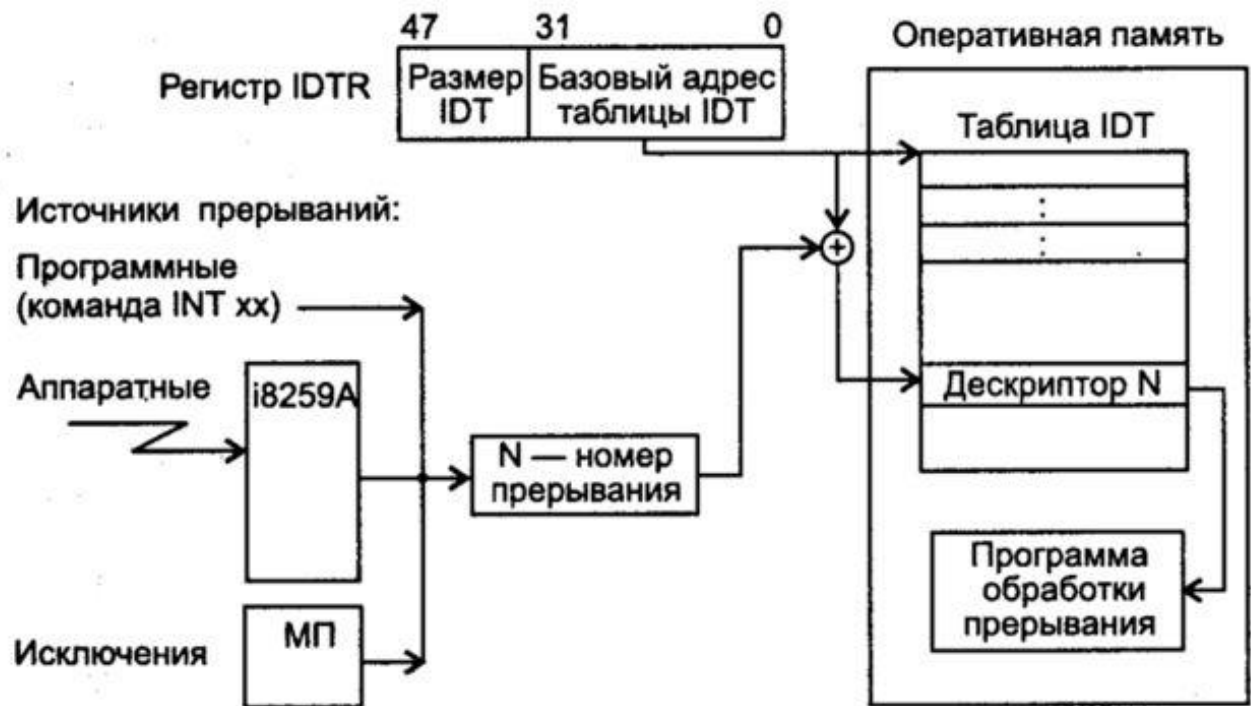
- ★ 256 прерываний (как и в реальном режиме)
- ★ дескрипторы строго упорядочены в соответствии с номерами прерываний
- ★ Состоит из 256 элементов (строк) по 8 байтов каждый
 - Таблица векторов прерываний – по 4 байта
- ★ Каждый дескриптор (строка) содержит адрес обработчика прерывания

**Как определить адрес,
по которому находится дескриптор прерывания с номером N
?**

Адрес таблицы дескрипторов прерываний - в регистре idtr
Смещение элемента в таблице IDT = $N * 8$

Схема поиска обработчика прерывания в защищенном режиме работы микропроцессора

Схема обработки прерывания в защищенном режиме



IDT – (Interrupt Descriptions Table) – таблица дескрипторов прерываний





Обработка прерывания в защищенном режиме работы микропроцессора

Отличия от реального режима

1. Поиск в ОП таблицы IDT (ее адрес в системном регистре `idtr`)
2. Поиск адреса обработчика прерывания
 - ★ Если возникло прерывание с номером N , то по смещению $[idtr] + (8 * N)$ находится 8-байтовый дескриптор прерывания N , содержащий адрес обработчика прерывания

Остальное аналогично