

Прерывания

Прерывания

¹Кафедра информационных технологий и систем
Национальная металлургическая академия Украины

15 сентября 2011 г.

Прерывание (англ. interrupt) — сигнал, сообщающий процессору о наступлении какого-либо события. При этом выполнение текущей последовательности команд приостанавливается и управление передаётся обработчику прерывания, который реагирует на событие и обслуживает его, после чего возвращает управление в прерванный код. В зависимости от источника возникновения сигнала прерывания делятся на:

- асинхронные или внешние (аппаратные) — события, которые исходят от внешних источников (например, периферийных устройств) и могут произойти в любой произвольный момент: сигнал от таймера, сетевой карты или дискового накопителя, нажатие клавиш клавиатуры, движение мыши. Факт возникновения в системе такого прерывания трактуется как запрос на прерывание (англ. Interrupt request, IRQ);

- синхронные или внутренние — события в самом процессоре как результат нарушения каких-то условий при исполнении машинного кода: деление на ноль или переполнение, обращение к недопустимым адресам или недопустимый код операции;

- программные (частный случай внутреннего прерывания) — инициируются исполнением специальной инструкции в коде программы. Программные прерывания как правило используются для обращения к функциям встроенного программного обеспечения (firmware), драйверов и операционной системы.

Контроллер прерываний (англ. Programmable Interrupt Controller, PIC) — микросхема или встроенный блок процессора, отвечающий за возможность последовательной обработки запросов на прерывание от разных устройств.

Как правило представляет собой электронное устройство, иногда выполненное как часть самого процессора или же сложных микросхем его обрамления, входы которого присоединены электрически к соответствующим выходам различных устройств. Номер входа контроллера прерываний обозначается "IRQ". Следует отличать этот номер от приоритета прерывания, а также от номера входа в таблицу векторов прерываний (INT). Так, например, в IBM PC в реальном режиме работы (в этом режиме работает MS-DOS) процессора прерывание от стандартной клавиатуры использует IRQ 1 и INT 9.

В процессоре Pentium был добавлен расширенный контроллер прерываний (APIC, Advanced PIC). Он состоит из модуля, встроенного в сам процессор (в случае многоядерной системы — в каждое ядро), называемого локальный контроллер прерываний (англ. local APIC), и центрального модуля, выполненного в одном экземпляре даже на многоядерном оборудовании, обычно как часть микросхем обрaмления процессора (англ. IO APIC).

Проводники IRQ от устройств подсоединены к ИО APIC. Для общения local APIC и ИО APIC, а также local APIC различных ядер друг с другом, используется передняя (frontside) шина многопроцессорной системы, также используемая для соединения процессоров и контроллера памяти. Варианты использования передней шины для общения APIC между собой — отдельные проводники, или же специальные типы транзакций — менялись от поколения к поколению процессоров Pentium и Core. APIC использовался в многоядерных/многопроцессорных системах, начиная с Intel Pentium (ядро P54). Начиная с этого процессора, каждый следующий снабжался интегрированным Local APIC-ом.

Преимущества расширенного контроллера прерываний:

- возможность реализации межпроцессорных прерываний — сигналов от одного процессора другому
- поддержка до 256 входов IRQ, в отличие от 16 на классической IBM PC
- крайне быстрый доступ к регистрам текущего приоритета прерывания и подтверждения прерывания. Контроллер прерываний, совместимый с IBM PC, исполнялся как устройство шины ISA с очень медленным доступом к его регистрам (порт 0x20).

В настоящий момент наблюдается тенденция к отказу от IO APIC, как и проводников IRQ, и переходу на Message Signaled Interrupts.

Message Signaled Interrupts (MSI, Прерывания, инициируемые сообщениями) в PCI версии 2.2 и более поздних, PCI-X, а также в PCI Express — альтернативная форма прерываний: вместо присваивания номера запроса на прерывание, устройству разрешается записывать сообщение по определённому адресу системной памяти, на деле отображенному на аппаратуру локального контроллера прерываний (local APIC) процессора. Для записи сообщения используется тот же механизм захвата шины (bus mastering), что и для DMA.

Для записи сообщений каждое устройство может иметь от одной до тридцати двух уникальных областей памяти. Все прерывания шины PCI Express всегда доставляются как MSI, даже при использовании эмуляции традиционных номеров проводников прерываний.

Достоинства MSI:

- возможность передачи некоторых данных вместе с информацией о наступлении события. Это зачастую избавляет обработчик прерывания от необходимости читать данную информацию из регистров состояния устройства.
- возможность полного отказа от проводников INT# от устройств и разъемов PCI до главного контроллера прерываний (IO APIC), а также от самого главного контроллера прерываний.
- в многопроцессорных системах устройства получают возможность самостоятельно выбирать процессор/ядро для обработки конкретного прерывания, причем делать это на уровне аппаратуры без исполнения кода. Это позволяет оптимизировать работу путем размещения большей части структур драйвера устройства и связанного с ним программного обеспечения в кэше конкретного процессора.

Поддерживается в операционной системе Microsoft Windows Vista и более поздних, в ОС FreeBSD с версии 6.3, а также в ядре Linux начиная с версии 2.6.8

В зависимости от возможности запрета внешние прерывания делятся на:

- маскируемые — прерывания, которые можно запрещать установкой соответствующих битов в регистре маскирования прерываний (в x86-процессорах — сбросом флага IF в регистре флагов);
- немаскируемые (англ. Non maskable interrupt, NMI) — обрабатываются всегда, независимо от запретов на другие прерывания. К примеру, такое прерывание может быть вызвано сбоем в микросхеме памяти.

Обработчики прерываний обычно пишутся таким образом, чтобы время их обработки было как можно меньшим, поскольку во время их работы могут не обрабатываться

другие прерывания, а если их будет много (особенно от одного источника), то они могут теряться.

Таблица векторов прерываний (англ. Interrupt Descriptor Table, IDT) используется в x86 архитектуре и служит для связи обработчика прерывания с вектором (номером) прерывания.

Для того чтобы связать адрес обработчика прерывания с номером прерывания, используется таблица векторов прерываний, занимающая первый килобайт оперативной памяти. Эта таблица находится в диапазоне адресов от 0000:0000 до 0000:03FFh и состоит из 256 элементов - дальних адресов обработчиков прерываний.

Элементы таблицы векторов прерываний называются векторами прерываний. В первом слове элемента таблицы записана компонента смещения, а во втором - сегментная компонента адреса обработчика прерывания.

Вектор прерывания с номером 0 находится по адресу 0000:0000, с номером 1 - по адресу 0000:0004 и т. д.

Для программиста, использующего язык C, таблицу векторов прерываний можно описать следующим образом:

```
void (far* interrupt_table[256])();
```

Инициализация таблицы выполняется частично системой базового ввода/вывода BIOS после тестирования аппаратуры и перед началом загрузки операционной системой.

Вашей программе может потребоваться организовать обработку некоторых прерываний. Для этого программа должна установить векторы нужных прерываний на свой обработчик. Это можно сделать, изменив содержимое соответствующего элемента таблицы векторов прерываний.

Очень важно не забыть перед завершением работы восстановить содержимое измененных векторов в таблице прерываний. Память, которая была распределена программе, после завершения работы программы освобождается. Она может быть использована, например, для загрузки другой программы. Если вы забыли восстановить вектор и пришло прерывание, то система может разрушиться - вектор теперь указывает на область, которая может содержать что угодно.

Последовательность действий для нерезидентных программ, желающих обрабатывать прерывания, должна быть такой:

- прочитать содержимое элемента таблицы векторов прерываний для вектора с нужным вам номером;
- запомнить это содержимое (адрес старого обработчика прерывания) в области данных программы;
- установить новый адрес в таблице векторов прерываний так, чтобы он указывал на начало вашей программы обработки прерывания;
- перед завершением работы программы прочитать из области данных адрес старого обработчика прерывания и записать его в таблицу векторов прерываний.

До окончания обработки прерывания обычно устанавливается запрет на обработку этого типа прерывания, чтобы процессор не входил в цикл обработки одного прерывания. Приоритизация означает, что все источники прерываний делятся на классы и каждому классу назначается свой уровень приоритета запроса на прерывание. Приоритеты могут обслуживаться как относительные и абсолютные.

- Относительное обслуживание прерываний означает, что если во время обработки прерывания поступает более приоритетное прерывание, то это прерывание будет обработано только после завершения текущей процедуры обработки прерывания.
- Абсолютное обслуживание прерываний означает, что если во время обработки прерывания поступает более приоритетное прерывание, то текущая процедура обработки прерывания вытесняется, и процессор начинает выполнять обработку вновь поступившего более приоритетного прерывания. После завершения этой процедуры процессор возвращается к выполнению вытесненной процедуры обработки прерывания.

Вектор прерывания — закреплённый за устройством номер, который идентифицирует соответствующий обработчик прерываний. Векторы прерываний объединяются в таблицу векторов прерываний, содержащую адреса обработчиков прерываний. Местоположение таблицы зависит от типа и режима работы процессора.

Таблица векторов прерываний (англ. Interrupt Descriptor Table, IDT) используется в x86 архитектуре и служит для связи обработчика прерывания с вектором (номером) прерывания.

В микропроцессорах 8086/80186 таблица векторов прерываний расположена в первом килобайте памяти начиная с адреса 0000:0000 и содержит 256 векторов прерываний в формате сегмент:смещение. Начиная с процессора 80286, адрес в физической памяти и размер таблицы прерываний определяется 48-битным регистром IDTR.

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, ошибка трассировки, переполнение).

- В реальном режиме элементом IDT является 32-битный FAR адрес обработчика прерывания.
- В защищённом режиме элементом IDT является шлюз прерывания длиной 8 байт, содержащий сегментный (логический) адрес обработчика прерывания, права доступа и др.
- В длинном режиме размер дескриптора прерывания увеличен до 16 байт.
- В режиме V86 при использовании расширения VME таблица векторов располагается по виртуальному адресу 0000:0000. Без использования этого расширения (и в том случае, если переадресация прерываний запрещена) при возникновении прерывания процессор покидает режим V86 и выполняет обычный обработчик защищённого режима.

Программное прерывание — синхронное прерывание, которое может осуществить программа с помощью специальной инструкции.

В процессорах архитектуры x86 для явного вызова синхронного прерывания имеется инструкция `Int`, аргументом которой является номер прерывания (от 0 до 255). В IBM PC-совместимых компьютерах обработку некоторых прерываний осуществляют подпрограммы BIOS, хранящиеся в ПЗУ, и это служит интерфейсом для доступа к сервису, предоставляемому BIOS. Также, обслуживание прерываний могут взять на себя BIOS карт расширений (например, сетевых или видеокарт), операционная система и даже обычные (прикладные) программы, которые постоянно находятся в памяти во время работы других программ (т. н. резидентные программы). В отличие от реального режима, в защищённом режиме x86-процессоров обычные программы не могут обслуживать прерывания, эта функция доступна

MS-DOS использует для взаимодействия со своими модулями и прикладными программами прерывания с номерами от 20h до 3Fh (числа даны в шестнадцатеричной системе счисления, как это принято при программировании на языке ассемблера x86). Например, доступ к основному множеству функций MS-DOS осуществляется исполнением инструкции Int 21h (при этом номер функции и её аргументы передаются в регистрах). Это распределение номеров прерываний не закреплено аппаратно и другие программы могут устанавливать свои обработчики прерываний вместо или поверх уже имеющихся обработчиков, установленных MS-DOS или другими программами, что, как правило, используется для изменения функциональности или расширения списка системных

Аппаратные прерывания вырабатываются устройствами компьютера, как правило, при завершении ими операций обмена данными или при изменении состояния. В зависимости от типа устройства обработчик прерывания может выполнять те или иные функции. Например, по прерыванию таймера соответствующий обработчик увеличивает содержимое счетчика, расположенного в оперативной памяти. По содержимому этого счетчика программы могут определить текущее время.

В отличие от программных прерываний, вызываемых запланировано программой или драйвером, аппаратные прерывания всегда происходят асинхронно по отношению к выполняющимся программам. Кроме того, может возникнуть одновременно сразу несколько прерываний!

Для того чтобы система "не растерялась" решая какое прерывание обслуживать в первую очередь, существует специальная схема приоритетов. Каждому прерыванию назначается свой приоритет. Если происходит одновременно несколько прерываний, система отдает предпочтение самому высокоприоритетному, откладывая на время обработку остальных прерываний.

Самый высокий приоритет у прерываний от интервального таймера, затем идет прерывание от клавиатуры.

Наименьший приоритет имеет прерывание принтера.

Если вам надо добавить какие-либо собственные действия к тем, что выполняет стандартный обработчик прерывания, то можно организовать цепочку прерываний.

Для организации цепочки прерываний нужно записать в векторную таблицу адрес своего обработчика, не забыв сохранить прежнее содержимое таблицы. Ваш обработчик получает управление по прерыванию, выполняет какие-либо действия, затем передает управление старому обработчику.

Можно сделать и по-другому: ваш обработчик вызывает старый обработчик как подпрограмму, а затем после возврата из старого обработчика выполняет дополнительные действия. Иными словами, вы можете вставить дополнительную обработку как до вызова старого обработчика, так и после его вызова.

В библиотеке транслятора C имеется функция для организации цепочки прерываний с именем `_chain_intr`. Для описания функции, выполняющей роль обработчика прерывания, следует использовать ключевое слово `interrupt`.

```
void interrupt far int_func(...) // Тело обработчика  
прерывания
```