Лекция №6

Управление устройствами

(часть 1)

Прерывания

Прерывание

- Прерывание – сигнал, сообщающий процессору о наступлении какого-либо события.
- Система прерываний переводит процессор на выполнение потока команд, отличного от того, который выполнялся до сих пор, с последующим возвратом к исходному коду.

Типы прерываний

- Внешние (аппаратные) – события от периферийных устройств.
- Внутренние (исключения) – события в микропроцессоре.
- Программные – возникают при выполнении особой команды процессора, выполнение которой имитирует прерывание.

Обработчики прерываний

Обработчики прерываний – процедуры, вызываемые в результате прерывания.

- Внешние прерывания обрабатываются драйверами соответствующих устройств.
- Внутренние – специальными модулями ядра.
- Программные – процедурами ОС, обслуживающими системные вызовы.

Синхронные и асинхронные прерывания

- Синхронное появление прерывания подразумевает, что при каждом запуске программы прерывание произойдёт при выполнении одной и той же команды.
- Асинхронное появление прерывания подразумевает, что при каждом запуске программы прерывание может произойти при выполнении любой команды или не произойти вообще.

Внешние прерывания

- Прерывания, асинхронные по отношению к потоку инструкций прерываемой программы. Возникают в результате действий пользователя или поступления сигналов завершения операций ввода-вывода от аппаратуры.
- IRQ (Interrupt Request) – запрос на аппаратное прерывание.

Внутренние прерывания

Внутренние прерывания возникают синхронно выполнению программы при появлении аварийной ситуации в ходе исполнения какой-либо инструкции программы:

- при нарушении адресации;
- при делении на ноль;
- при переполнении или исчезновении порядка.

Программные прерывания

- Прерывание происходит в предсказуемой точке программы, заданной программистом.
- Использование программных прерываний приводит к более компактному коду программы, за счёт исключения кода обработчиков прерываний.
- Смена пользовательского режима на привилегированный, необходимый для работы с устройством, происходит одновременно с вызовом процедуры.

Вызов обработчика прерывания

Информация, необходимая для вызова обработчика прерывания:

- номер прерывания, сопоставленный с определённым устройством или процедурой ОС;
- начальный адрес обработчика прерывания в памяти.

Таблица векторов прерываний

- В таблицу векторов прерываний включаются записи, содержащие номер прерывания и начальный адрес обработчика.
- Для быстрого обращения к таблице векторов прерываний её помещают в начальных адресах оперативной памяти.

Примеры прерываний

- 00h – ошибка деления.
- 04h – ошибка переполнения.
- 08h – прерывание от таймера.
- 09h – прерывание от клавиатуры.
- 21h – программное прерывание.

Перехват прерывания

Перехват прерывания – замена обработчика прерывания на собственный. Алгоритм:

- чтение и сохранение начального адреса текущего обработчика прерывания, находящегося под нужным номером в таблице векторов прерываний;
- установка начального адреса своего обработчика в таблице векторов прерываний;
- перед завершением работы программы восстановление начального адреса старого обработчика прерывания.

Приоритезация прерываний

Приоритезация – все источники прерываний делятся на классы и каждому классу присваивается свой уровень приоритета запроса на прерывание.

Схемы обслуживания приоритетов прерываний:

- абсолютное обслуживание;
- относительное обслуживание.

Абсолютное и относительное обслуживание прерываний

- Относительное – при поступлении более приоритетного прерывания его обработка наступит после завершения текущей процедуры обработки прерывания.
- Абсолютное – при поступлении более приоритетного прерывания текущая процедура обработки прерывания вытесняется и процессор начинает выполнение обработчика поступившего прерывания.

Пример приоритезации

- IRQ0 – прерывание таймера (возникает 18,2 раза в секунду).
- IRQ1 - прерывание от клавиатуры (генерируется, когда пользователь нажимает и отжимает клавиши).
- IRQ2 - используется для каскадирования аппаратных прерываний.
- IRQ3 – прерывание порта COM2.
- IRQ4 – прерывание порта COM1.

Способы выполнения прерываний

- Опрашиваемый – процессору предоставляется информация об уровне приоритета прерывания.
- Векторный – процессору предоставляется информация об уровне приоритета прерывания и о начальном адресе обработчика прерывания.

Векторный способ

- Устройству назначается вектор прерываний, который включает номер, присвоенный данному устройству.
- При получении сигнала запроса на прерывание процессор выполняет специальный цикл подтверждения прерывания.
- В течение цикла подтверждения процессор получает вектор прерывания устройства.
- Используя вектор прерывания, процессор находит обработчик данного прерывания.

Опрашиваемый способ

- Процессор получает только информацию об уровне приоритета прерывания (например, IRQ).
- При возникновении прерывания процессор определяет, какое устройство запросило прерывание из всех устройств, связанных с данным уровнем приоритета. Вызываются все обработчики прерываний, сопоставленные с данным уровнем приоритета, пока один из обработчиков не подтвердит, что именно он обслуживает данное устройство.

Комбинированный способ

- Процессор поддерживает векторную систему прерываний.
- Контроллеры периферийных устройств генерируют сигнал запроса с присвоенным уровнем IRQ.
- Если уровню IRQ соответствует больше одного устройства, то происходит опрос всех устройств данного уровня.
- Контроллер прерываний отображает поступающий от шины сигнал IRQ на определённый номер вектора (0..255).

Маскирование прерываний

Маскирование – при обслуживании некоторого запроса на прерывание все запросы с равным или более низким приоритетом маскируются, т.е. не обслуживаются.

Последовательность действий при обработке прерывания

- Возникновение сигнала (аппаратное прерывание) или условия (внутреннее прерывание).
- Аппаратное распознавание типа прерывания: если прерывания данного типа запрещены, то процессор продолжает работу без изменений; если разрешены, то определяется адрес обработчика прерывания и он автоматически вызывается.

Последовательность действий при обработке прерывания

- При смене процесса автоматически сохраняется его контекст, необходимый для последующего восстановления прерванного процесса.
- Если необходимо, то происходит смена режима работы процессора на привилегированный.

Последовательность действий при обработке прерывания

- Временно запрещаются прерывания данного типа или применяется маскирование прерываний.
- Происходит обработка прерывания.
- Контекст прерванного потока восстанавливается.
- Снимается блокировка запрещённых прерываний.

Диспетчеризация прерываний

- При каждом возникновении прерывания вызывается диспетчер прерываний.
- Если во время обработки одного прерывания возникают другие, то диспетчер прерываний упорядочивает работу обработчиков.
- Диспетчеризация осуществляется за счёт использования механизма приоритетных очередей, что приводит к исключению задержек критических задач, поступающих на выполнение.

Действия диспетчера прерываний

- При возникновении прерывания диспетчер вызывается первым.
- На некоторое время диспетчер запрещает все прерывания.
- Диспетчер выясняет источник прерывания.

Действия диспетчера прерываний

- Диспетчер сравнивает приоритет данного источника прерывания с приоритетом потока команд, выполняемого процессором.
- Если текущий приоритет ниже, то работа текущего обработчика приостанавливается и он помещается в очередь; если текущий приоритет выше, то в очередь помещается обработчик поступившего запроса.

Механизм прерываний в Windows NT

- Все источники прерываний делятся на классы, каждому классу присваивается уровень запроса прерывания (приоритет класса) – IRQL.
- ОС программно поддерживает внутреннюю переменную IRQL выполняемого процессором кода.
- Низший уровень IRQL соответствует обычным потокам, назначенным диспетчером потоков.

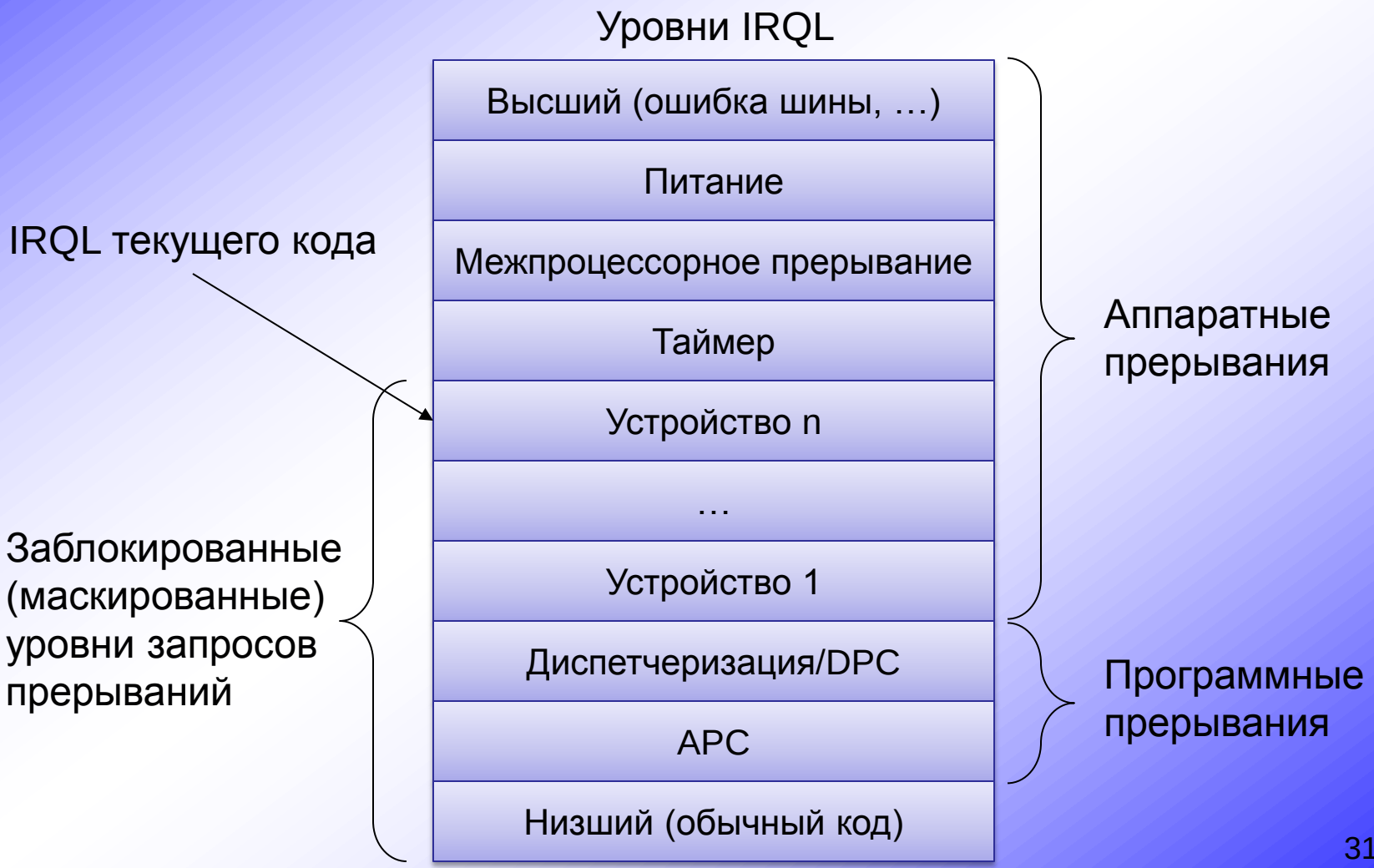
Диспетчер прерываний Windows NT

- При поступлении в процессор сигнала запроса на прерывание диспетчер запоминает информацию об источнике прерывания и анализирует его приоритет.
- Если приоритет запроса ниже или равен IRQL прерванного кода, то обслуживание запроса откладывается, а информация помещается в очередь; если текущий IRQL ниже, то текущий обработчик вытесняется и помещается в очередь, а управление передаётся обработчику поступившего запроса, при этом IRQL становится равным уровню принятого на выполнение запроса.

Диспетчер прерываний Windows NT

- По окончании работы обработчика управление возвращается диспетчеру, который из очереди прерываний выбирает наиболее приоритетное.
- IRQL снижается до уровня выбранного прерывания.

Иерархия IRQL в Windows NT



Приоритезация прерываний от одного устройства

- Приоритезация прерываний, генерируемых одним устройством, нужна для увеличения скорости выполнения критичных задач.
- Длительные по времени задачи (копирование больших объёмов данных) выполняются с более низким приоритетом, чем задачи, критичные ко времени исполнения.

Прерывание диспетчеризация/DPC

- В очередь DPC (отложенный вызов процедуры) помещаются запросы, вызывающие диспетчер потоков, и отложенные вызовы.
- Во время выполнения высокоприоритетных прерываний диспетчер потоков вызывается при помощи программного прерывания.
- В качестве отложенных вызовов могут вызываться части драйверов, выполняющие не критичные к срокам действия с большим объёмом работы.

Прерывание APC

- APC – процедуры, всегда выполняющиеся в контексте определённого процесса.
- У каждого потока есть своя очередь APC.
- Основное назначение – перемещение данных, полученных драйвером от устройства ввода-вывода, из системной области памяти в индивидуальную часть адресного пространства процесса, запросившего операцию ввода-вывода.

Функции подсистемы ввода-вывода

Функции подсистемы управления устройствами ввода-вывода

- Организация параллельной работы устройств ввода-вывода и процессора.
- Согласование скоростей обмена и кэширование данных.
- Разделение устройств между процессами.
- Обеспечение удобного логического интерфейса между устройствами и остальной частью ОС.

Функции подсистемы управления устройствами ввода-вывода

- Поддержка широкого спектра драйверов с возможностью простого включения в систему нового драйвера.
- Динамическая загрузка и выгрузка драйверов.
- Поддержка нескольких файловых систем.
- Поддержка синхронных и асинхронных операций ввода-вывода.

Взаимодействие устройства ввода-вывода и ОС

- Контроллер – специализированный блок управления устройством ввода-вывода.
- Драйвер – системный программный модуль, управляющий устройством.
- Контроллер получает от драйвера выводимые на устройство данные, а также команды управления. По окончании выполнения задачи контроллер генерирует прерывание.

Организация параллельной работы устройств ввода-вывода и процессора

- Контроллер берёт на себя управление устройством, работа контроллера проходит независимо от ОС в периоды между выдачами команд.
- Подсистема ввода-вывода в реальном масштабе времени планирует запуск и приостановку различных драйверов, обеспечивая приемлемое время реакции каждого драйвера на независимые события контроллера.

Согласование скоростей обмена и кэширование данных

- Согласование необходимо из-за различия скорости обмена данными устройства с оперативной памятью и скоростью работы устройства. При этом сокращается реальное количество операций ввода-вывода.
- Для согласования скоростей используется буферизация данных и реализация синхронного доступа считывающего и пишущего процессов (потоков) к буферу.
- Примеры: печать документов, набор символов на клавиатуре.

Согласование скоростей обмена и кэширование данных

- Если разница в скорости небольшая, то в качестве буфера используется часть оперативной памяти.
- Если разница в скорости большая, то в качестве буфера используются дисковые файлы.
- В случае обмена большим количеством данных может использоваться большая буферная память в контроллере устройства.

Разделение устройств и данных между процессами

- При совместном использовании устройств процессами необходим контроль доступа процессов к устройству.
- Если возможен доступ к отдельным порциям данных, хранимых или отображаемых устройством, то необходим контроль доступа процессов и к устройству и к порциям данных.
- При совместном использовании устройств необходимо разграничивать порции данных различных процессов друг от друга.

Обеспечение удобного логического интерфейса между устройствами и ОС

- Основа логического интерфейса – файловая модель периферийных устройств (виртуальные устройства).
- Файловая модель – любое устройство представляется последовательным набором байт, с которым можно работать при помощи системных вызовов (например, записывать данные, направляемые этому устройству).

Поддержка широкого спектра драйверов

Типы интерфейсов для драйверов:

- DKI – интерфейс «драйвер-ядро» (обязательно должен быть стандартизирован);
- DDI – интерфейс «драйвер-устройство» (должен быть стандартизирован при запрете в системе непосредственного взаимодействия драйвера с устройством).
- Запрет на прямое взаимодействие приводит к независимости драйвера от аппаратуры ПК.
- Для устройств разных классов интерфейсы DKI/DDI могут быть различны.

Поддержка синхронных и асинхронных операций ввода-вывода

- Синхронный режим – процесс приостанавливает свою работу, пока не будет завершена операция ввода-вывода.
- Асинхронный режим – процесс продолжает выполняться одновременно с операцией ввода-вывода.

Обработка ошибок

- При критических ошибках – вывод сообщения об ошибке и завершение работы системы.
- При ошибках программирования (обращение к несуществующему устройству, попытки чтения данных с устройства вывода и наоборот) – код ошибки возвращается вызывающему процессу.

Обработка ошибок

- При ошибках ввода-вывода (обращение к выключенному устройству, обращение к повреждённому блоку) – попытка устранения ошибки драйвером устройства, в случае неудачи ОС может повторить операцию, игнорировать ошибку или уничтожить процесс.

Рассмотренные вопросы

- Типы и обработка прерываний.
- Приоритезация и маскирование прерываний.
- Диспетчеризация прерываний.
- Иерархия уровней прерываний.
- Функции подсистемы ввода-вывода.

Всем спасибо – все свободны,

если нет вопросов