

Операционные системы

Управление центральным процессором и объединение ресурсов

Управление центральным процессором...

Объекты Windows

Основные понятия

- Задание – набор процессов, управляемых как единое целое, с общими квотами и лимитами
- **Процесс** – контейнер для ресурсов
- **Поток** – сущность планируемая ядром
- Волокно – облегченный поток, управляемый полностью в пространстве пользователя

Процессы и потоки Windows

- Иерархия процессов (ссылки между процессами-родителями и процессами-потомками) в явном виде не поддерживается.
- Процессы имеют многопоточную организацию.
- Процессы и потоки реализованы в форме объектов, и доступ к ним осуществляется посредством службы объектов.
- Объекты-процессы и объекты-потоки имеют встроенные средства синхронизации.

Процессы

- Процесс – это совокупность системных ресурсов, задействованная для выполнения определенной работы.
- Понятие "процесс" включает следующее:
 - исполняемый код;
 - собственное адресное пространство, которое представляет собой совокупность виртуальных адресов, которые может использовать процесс;
 - ресурсы системы, такие как файлы, семафоры и т.п., которые назначены процессу операционной системой;
 - хотя бы один выполняемый поток.

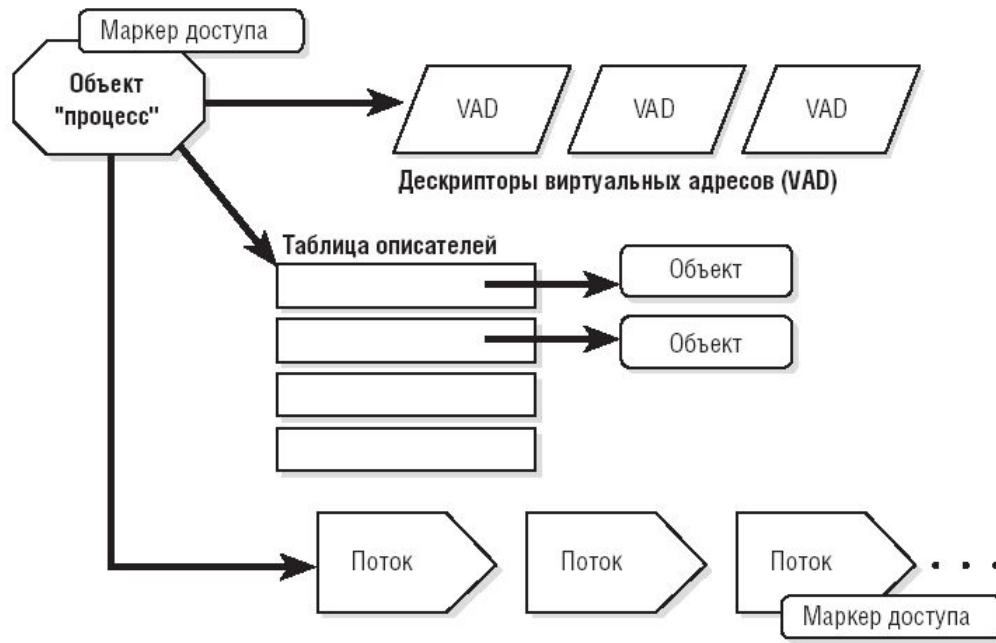
Атрибуты процесса

- Идентификатор процесса – уникальное значение, которое идентифицирует процесс в рамках ОС (только в текущий момент времени).
- Закрытое *виртуальное адресное пространство* – диапазон адресов виртуальной памяти, которым может пользоваться процесс.
- Исполняемую программу – начальный код и данные, проецируемые на виртуальное адресное пространство процесса.
- Список открытых дескрипторов различных системных ресурсов – семафоров, файлов и других объектов, доступных всем потокам в данном процессе.
- Маркер доступа – исполняемый объект, содержащий информацию о безопасности и идентифицирующий пользователя, группы безопасности и привилегии, сопоставленные с процессом.
- **Базовый приоритет – основа для исполнительного приоритета потоков процесса.**
- Процессорная совместимость – набор процессоров, на которых могут выполняться потоки процесса.
- Предельные значения квот ресурсов (см. Job).
- Время исполнения – общее количество времени, в течение которого выполняются все потоки процесса.
- Список потоков процесса (как минимум один поток).

ПОТОКИ

- Поток (нить) – это непрерывная последовательность инструкций, выполняющих определенную функцию.
- Потоки не имеют собственного адресного пространства и получают доступ к адресному пространству процесса-родителя. Это означает, что все потоки в процессе могут записывать и считывать содержимое памяти любого из потоков данного процесса.
- Потоки процесса могут сообща пользоваться ресурсами процесса-родителя.
- По умолчанию у потоков нет собственного маркера доступа и для целей безопасности используется *маркер доступа* процесса-родителя.
- При необходимости каждый поток может получить собственный маркер доступа.

Поток и ресурсы процесса



- Дескрипторы виртуальных адресов (virtual address descriptors, VAD) – это структуры данных, используемые VMM для учета виртуальных адресов, задействованных процессом.

Атрибуты потоков

- Идентификатор клиента – уникальное значение, которое идентифицирует поток при ее обращении к серверу.
- Контекст потока – информация, которая необходима ОС для того, чтобы продолжить выполнение прерванного потока. Контекст потока содержит текущее состояние регистров, стеков и индивидуальной области памяти.
- Два стека, один из которых используется потоком при выполнении в режиме ядра, а другой – в пользовательском режиме.
- Локальная память потока (thread local storage, TLS) – закрытая область памяти потока.
- **Динамический приоритет** – значение приоритета потока в данный момент.
- **Базовый приоритет** – нижний предел динамического приоритета потока.
- Процессорная совместимость потока – перечень типов процессоров, на которых может выполняться поток.
- Номер идеального процессора потока.
- Номер процессора, на котором поток выполнялся прошлый раз.
- Время выполнения потока – суммарное время выполнения потока в пользовательском режиме и в режиме ядра.
- Счетчик приостановок – текущее количество приостановок выполнения потока.

Номер идеального процессора потока

- Идеальный процессор для потока выбирается случайным образом при его создании с использованием зародышевого значения (seed) в блоке процесса. Это значение увеличивается на 1 всякий раз, когда создается новый поток, поэтому создаваемые потоки равномерно распределяются по набору доступных процессоров. Например, первый поток в первом процессе в системе закрепляется за идеальным процессором 0, второй поток того же процесса – за идеальным процессором 1. Однако у следующего процесса в системе идеальный процессор для первого потока устанавливается в 1, для второго – в 2 и т. д. Благодаря этому потоки внутри каждого процесса равномерно распределяются между процессорами.
- Номер идеального процессора может быть изменен с помощью функции *SetThreadIdealProcessor*.

Задание (job)

- Процессы Windows не поддерживают в явном виде отношения родитель-потомок, поэтому начиная с Windows 2000 в модель процессов введено новое расширение – *задания (jobs)*.
- Задания предназначены в основном для того, чтобы группами процессов можно было оперировать и управлять как единым целым.
- Задание позволяет устанавливать определенные атрибуты и накладывать ограничения на процесс или процессы, сопоставленные с заданием.



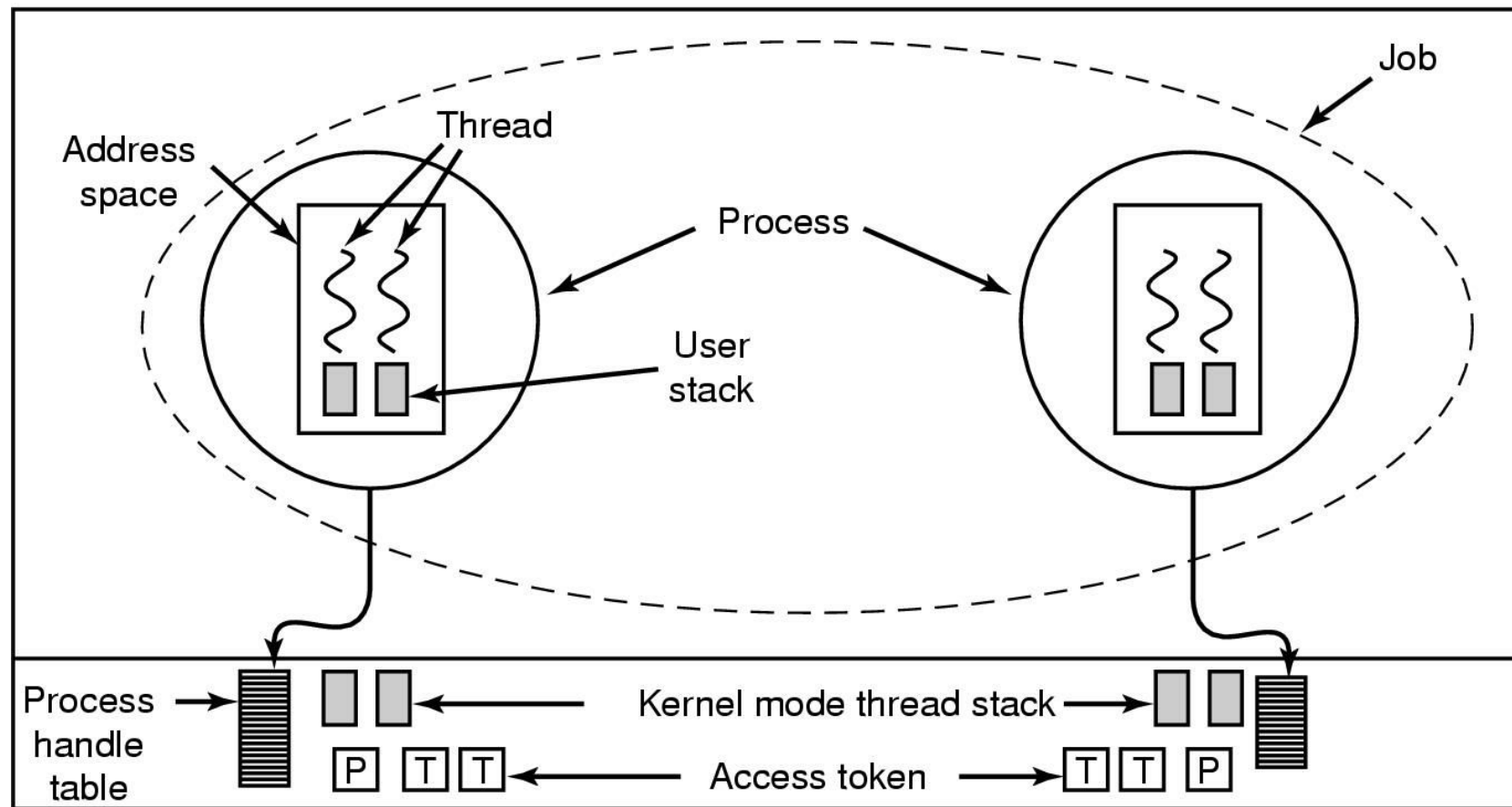
КВОТЫ И ЛИМИТЫ ДЛЯ ЗАДАНИЙ

- квоты (базовые и расширенные базовые ограничения):
 - максимальное количество процессов;
 - суммарное время центрального процессора (для каждого процесса и для задания в целом);
 - максимальное количество используемой памяти (для каждого процесса и для задания в целом);
- базовые ограничения по пользовательскому интерфейсу;
- ограничения, связанные с защитой.

Волокна (fibers)

- Введены в Windows 2000 для переноса существующих серверных приложений из UNIX.
- Реализованы на уровне кода пользовательского режима. В потоке может быть одно или несколько волокон. Для ядра поток – все то, что можно вытеснить и что выполняет код. Единовременно поток будет выполнять код лишь одного волокна.
- Процессорное время между волокнами распределяется по пользовательскому алгоритму.

Взаимосвязь между заданиями, процессами и потоками



Управление центральным процессором...

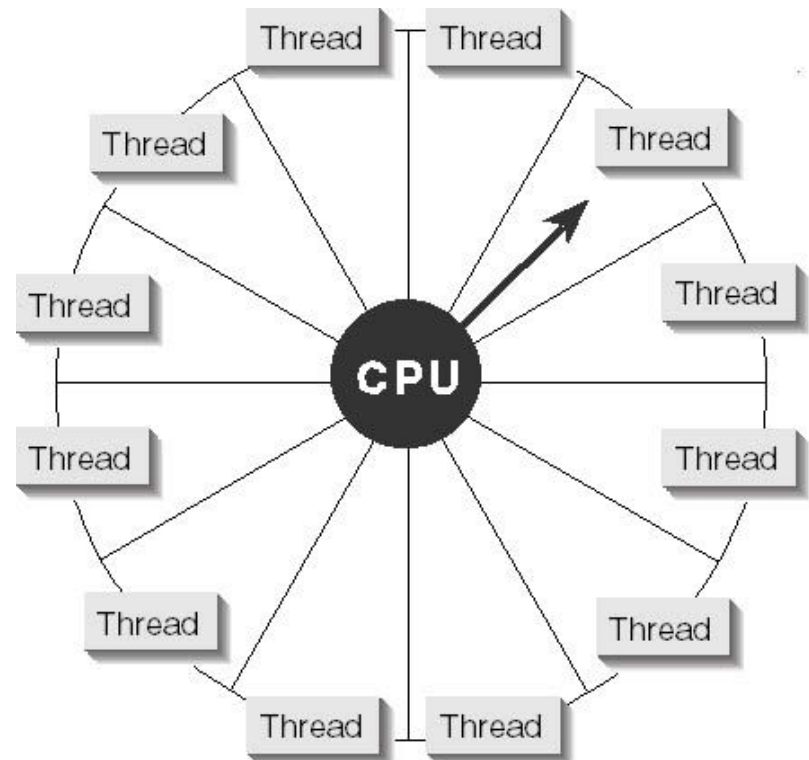
Планирование загрузки
однопроцессорной системы

Планирование загрузки процессорного времени

- В Windows реализована вытесняющая многозадачность, при которой ОС не ждет, когда поток сам захочет освободить процессор, а принудительно снимает его с выполнения после того, как тот израсходовал отведенное ему время (квант), или если в очереди готовых появился поток с более высоким приоритетом (дисциплина обслуживания с абсолютными приоритетами).
- Планировщик всегда выбирает из очереди поток с наивысшим приоритетом; если сразу несколько потоков имеют одинаковый приоритет, планировщик предоставляет квант процессорного времени каждому из них по очереди.
- Если поток по каким-то причинам преждевременно перестает быть активным (например, когда останавливается в ожидании завершения операции ввода/вывода), ядро Windows вызывает планировщик для поиска следующего потока, которому можно передать управление процессором.
- Если поток выбирает весь отведенный ему квант времени, ядро Windows вызывает планировщик с тем, чтобы другие потоки также могли выполнять свою работу. Переключение процессора с выполнения команд одного потока на выполнение команд другого называют переключением контекста.

Квантование времени

- ОС выделяет потокам кванты времени по принципу карусели.
- Чтобы все эти потоки работали, ОС отводит каждому из них определенное процессорное время. Выделяя потокам *кванты* времени по принципу карусели, она создает тем самым иллюзию одновременного выполнения потоков.
- Слайд иллюстрирует распределение процессорного времени между потоками на машине с одним процессором. Если в машине установлено более одного процессора, алгоритм работы ОС значительно усложняется (в этом случае система стремится сбалансировать нагрузку между процессорами).

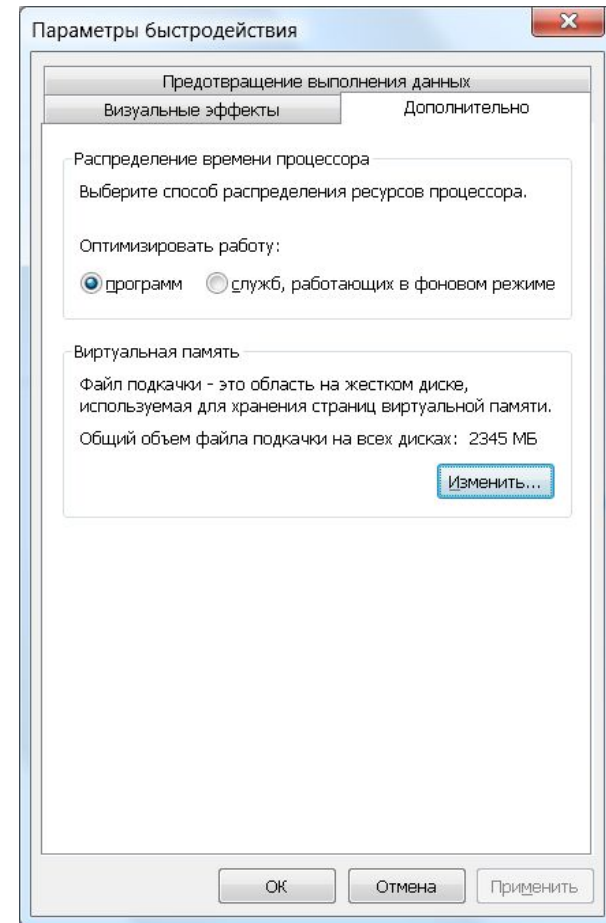


Квантование времени

- По умолчанию в Windows 2000 Professional и Windows XP потоки выполняются в течение 2 интервалов таймера (clock intervals), а в системах Windows Server 2000, 2003 – 12 интервалов таймера.
- В зависимости от конкретной аппаратной платформы, интервал таймера составляет примерно 10 или 15 мс.
- В серверных системах величина кванта увеличена для того, чтобы свести к минимуму переключение контекста. Получая больший квант, серверные приложения, которые пробуждаются при получении клиентского запроса, имеют больше шансов выполнить запрос и вернуться в состояние ожидания до истечения выделенного кванта.

Настройка кванта

- Вы можете изменить квант для потоков всех процессов, но выбор ограничен всего двумя значениями:
 - «оптимизация работы программ» -короткий квант (2 такта таймера), используется по умолчанию для клиентских компьютеров);
 - «оптимизация работы служб, работающих в фоновом режиме» - длинный (12 тактов таймера), используется по умолчанию для серверных систем).



Приоритетный режим обслуживания

- Разработчик ПО может использовать приоритеты от 1 до 31.
- Нулевой приоритет зарезервирован для потока обнуления страниц.
- Поток наследует приоритет процесса, породившего его.
- ОС Windows NT 4.0 предоставляет 4 класса приоритетов: Realtime, High, Normal и Idle.
- ОС Windows 2000: еще 2 дополнительных класса приоритетов – Below Normal и Above Normal.
- Относительный приоритет потока: idle, lowest, below normal, normal (обычный), above normal, highest и time-critical.

Классы приоритета процессов

Класс приоритета	Описание
Real-time (24)	Потоки в этом процессе обязаны немедленно реагировать на события, обеспечивая выполнение критических по времени задач. Такие потоки вытесняют даже компоненты ОС. Процессы этого класса нельзя запустить, если пользователь не имеет привилегии Increase Scheduling Priority (администратор и пользователь с расширенными полномочиями).
High (13)	Потоки в этом процессе тоже должны немедленно реагировать на события, обеспечивая выполнение критических по времени задач. Этот класс присвоен, например, Task Manager.
<i>Above normal (10)</i>	<i>Класс приоритета, промежуточный между normal и high.</i>
Normal (8)	Потоки в этом процессе не предъявляют особых требований к выделению им процессорного времени (99% приложений).
<i>Below normal (6)</i>	<i>Класс приоритета, промежуточный между normal и idle.</i>
Idle (4)	Потоки в этом процессе выполняются, когда система не занята другой работой. Этот класс приоритета обычно используется для утилит, работающих в фоновом режиме, экранных заставок и пр.

Относительные приоритеты потоков

Относительный приоритет потока	Описание
Time-critical	Поток выполняется с приоритетом 31 в классе real-time и с приоритетом 15 в других классах
Highest	Поток выполняется с приоритетом на два уровня выше обычную для данного класса
Above normal	Поток выполняется с приоритетом на один уровень выше обычного для данного класса
Normal	Поток выполняется с обычным приоритетом процесса для данного класса
Below normal	Поток выполняется с приоритетом на один уровень ниже обычного для данного класса
Lowest	Поток выполняется с приоритетом на два уровня ниже обычного для данного класса
Idle	Поток выполняется с приоритетом 16 в классе real-time и с приоритетом 1 в других классах

Иллюстрация по приоритетам для Windows 2000

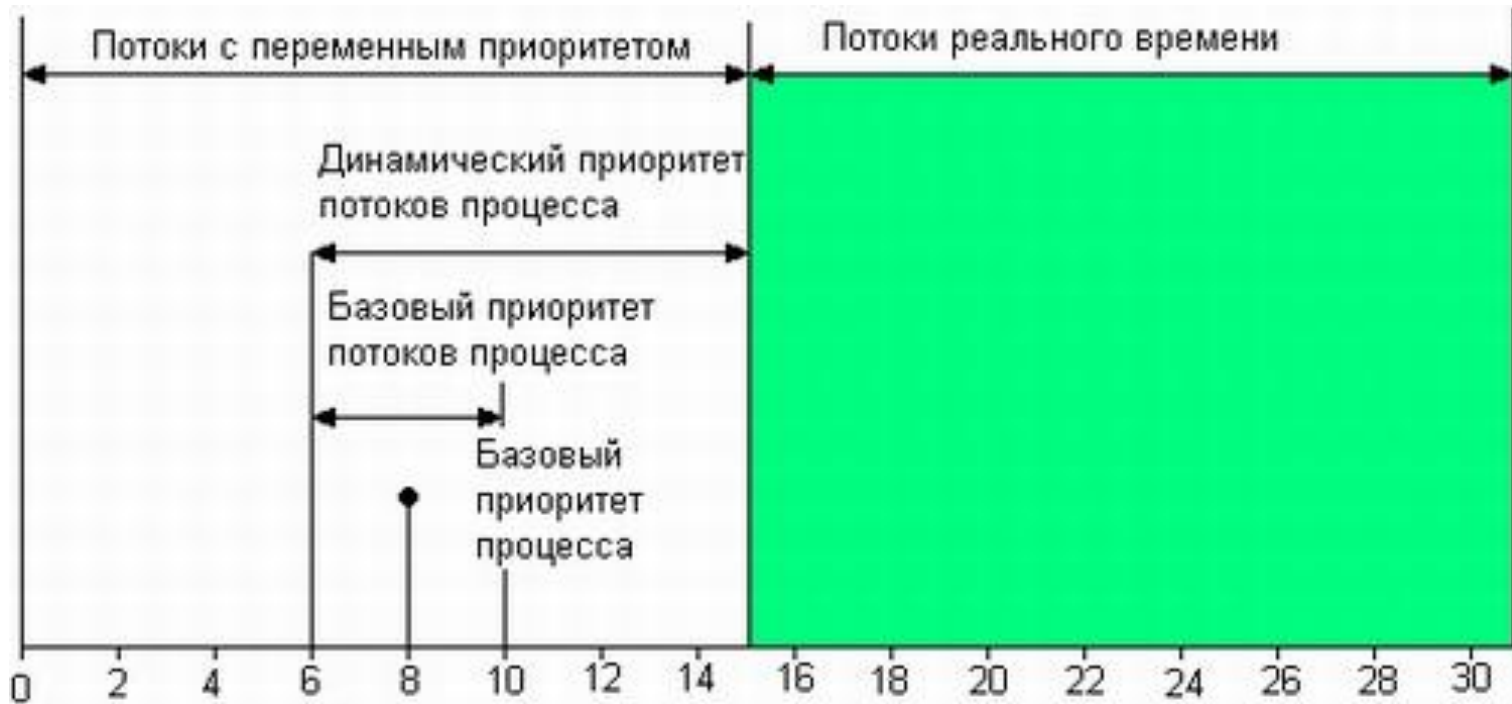


- ✓ Нулевой приоритет зарезервирован для потока обнуления страниц
- ✓ Приоритеты 17-21 и 27-30 могут использоваться только при написании драйвера, работающего в режиме ядра.

Базовый и динамический приоритеты потока

- Уровень приоритета, получаемый комбинацией относительного приоритета потока и класса приоритета процесса, которому принадлежит данный поток, называют *базовым уровнем приоритета потока*. Иногда система изменяет уровень приоритета потока.
- Использование *динамических приоритетов*, изменяющихся во времени, позволяет реализовать адаптивное планирование, при котором не дискриминируются интерактивные задачи, часто выполняющие операции ввода-вывода и недоиспользующие выделенные им кванты.
- Система повышает приоритет только тех потоков, базовый уровень которых находится в пределах 1-15. Именно поэтому данный диапазон называется "областью динамического приоритета" (dynamic priority range). Система не допускает динамического повышения приоритета потока до уровней реального времени (более 15). Поскольку потоки с такими уровнями обслуживают системные функции, это ограничение не дает приложению нарушить работу ОС. И, кстати, система никогда не меняет приоритет потоков с уровнями реального времени (от 16 до 31).

Динамические приоритеты



В ходе выполнения нити ее приоритет (1-15) может меняться – механизм *адаптивного планирования*.

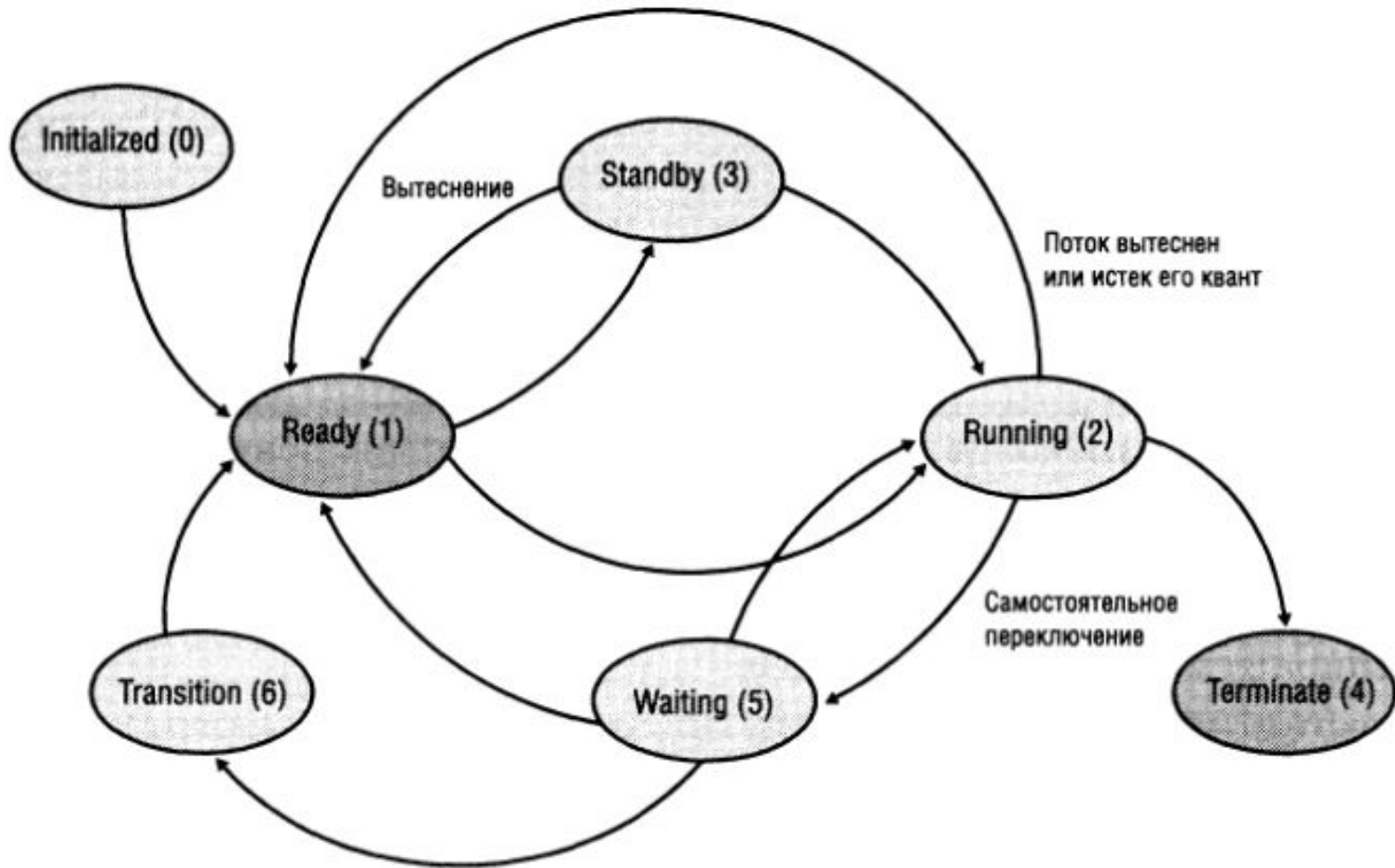
Принципы адаптивного планирования

- Если поток полностью исчерпал свой квант, то его приоритет понижается на некоторую величину.
- Приоритет потоков, которые перешли в состояние ожидания, не использовав полностью выделенный им квант, повышается.
- Приоритет не изменяется, если поток вытеснен более приоритетным потоком.
- Повышение приоритета для “голодающих” потоков.

Пример “голодающего” потока

- Представьте, что поток с приоритетом 4 готов к выполнению, но не может получить доступ к процессору из-за того, что его постоянно занимают потоки с приоритетом 8.
- Это типичный случай "голодания" потока с более низким приоритетом. Обнаружив такой поток, не выполняемый на протяжении уже трех или четырех секунд, система поднимает его приоритет до 15 и выделяет ему двойную порцию времени. По его истечении потоку немедленно возвращается его базовый приоритет.

Граф состояний потоков для Windows 2000



Состояния потоков

- **Готовность.** При поиске потока на выполнение диспетчер просматривает только потоки, находящиеся в состоянии готовности, у которых есть все для выполнения, но не хватает только процессора.
- **Первоочередная готовность (standby).** Для каждого процессора системы выбирается один поток, который будет выполняться следующим. Когда условия позволяют, происходит переключение на контекст этого потока.
- **Выполнение.** Как только происходит переключение контекстов, поток переходит в состояние выполнения и находится в нем до тех пор, пока не выполнится одно из условий освобождения процессора (см. далее).
- **Ожидание.** Поток может входить в состояние ожидания несколькими способами: поток по своей инициативе ожидает некоторый объект для того, чтобы синхронизировать свое выполнение, например, ожидает завершения операции ввода-вывода.
- **Переходное состояние.** Поток входит в переходное состояние, если он готов к выполнению, но ресурсы, которые ей нужны, заняты. Например, страница, содержащая стек потока, может быть выгружена из на диск. При освобождении ресурсов поток переходит в состояние готовности.
- **Завершение.** Когда выполнение потока закончилось, он входит в состояние завершения.

Условия освобождения потоком процессора

- исчерпан квант;
- появляется более приоритетный готовый поток;
- поток сам добровольно переходит в очередь готовых (Sleep (0), SwitchToThread);
- поток добровольно блокируется, уходя в состояние ожидания (Sleep (n), WaitForSingleObject, синхронный ввод/вывод, ...);
- поток снижает свой приоритет;
- поток добровольно исключает текущий процессор из маски совместимости (привязки) (для МПС);
- поток завершает свое выполнение.

Поток простоя

- Если нет ни одного потока, готового к выполнению на процессоре. Windows подключает к данному процессору поток простоя.
- Поток простоя не имеет приоритета, поскольку он выполняется лишь в отсутствие других потоков.
- Для каждого процессора (в МПС) создается свой поток простоя.

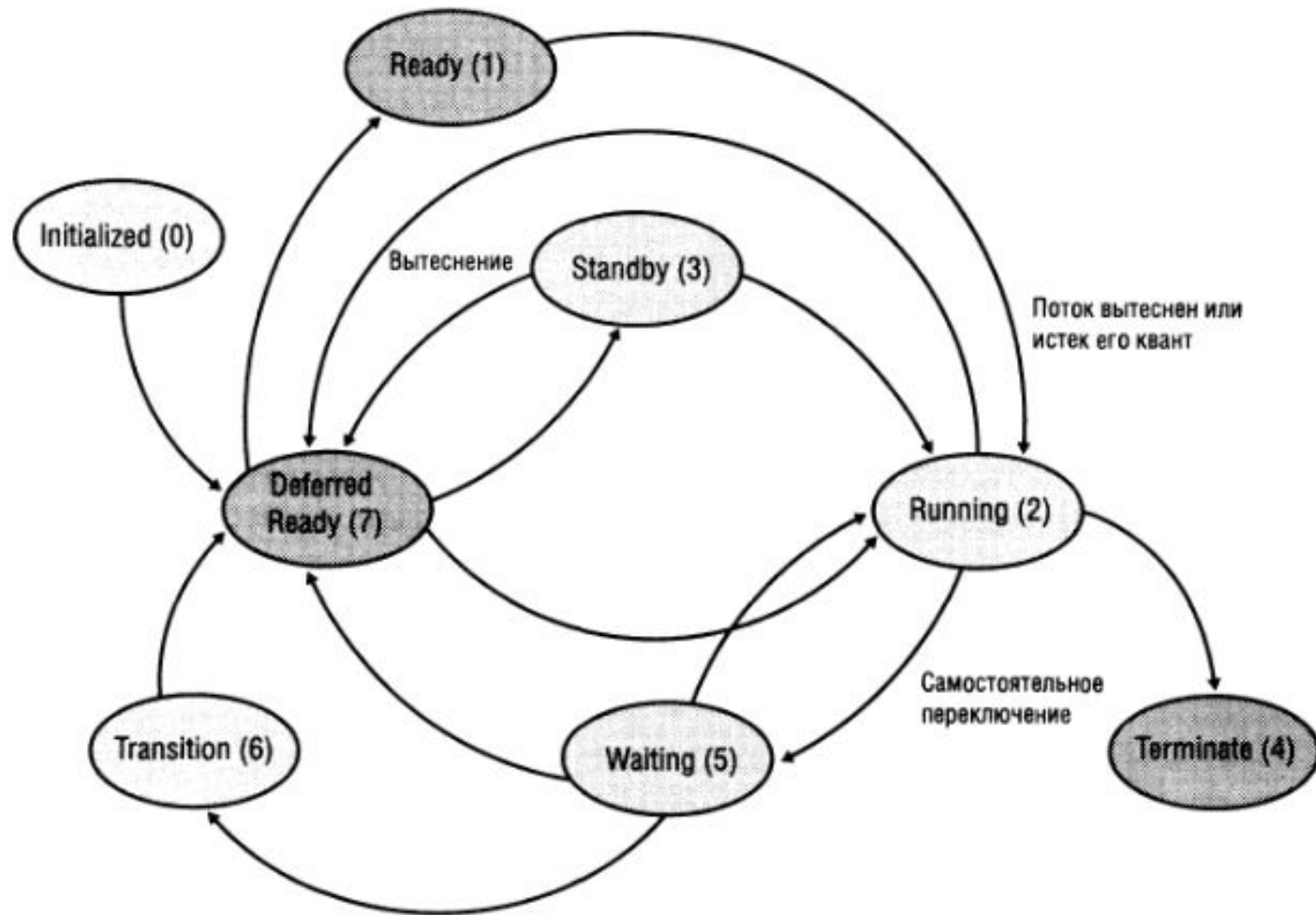
Поток простоя

- Поток простоя выполняет следующие действия.
 - Включает и отключает прерывания (тем самым давая возможность доставить отложенные прерывания).
 - Проверяет, нет ли у процессора незавершенных отложенных программных прерываний (DPC – Deferred procedure call). Если таковые есть, сбрасывает отложенное программное прерывание и доставляет эти DPC.
 - Проверяет, выбран ли какой-нибудь поток для выполнения на данном процессоре, и, если да, организует его диспетчеризацию.
 - Вызывает из HAL процедуру обработки процессора в простое (если нужно выполнить какие-либо функции управления электропитанием).
- В Windows Server 2003 поток простоя также проверяет наличие потоков, ожидающих выполнения на других процессорах, но об этом пойдет речь в разделе по планированию потоков в многопроцессорных системах.

Граф состояний потоков для Windows 2003

- Схема состояний потоков в Windows Server 2003 показана на следующем слайде.
- Обратите внимание на новое состояние Deferred Ready (готов, отложен). Это состояние используется для потоков, выбранных для выполнения на конкретном процессоре, но пока не запланированных к выполнению.
- Это новое состояние предназначено для того, чтобы ядро могло свести к минимуму срок применения общесистемной блокировки к базе данных планирования (scheduling database).

Граф состояний потоков для Windows 2003



Управление центральным процессором...

Управление потоками в многопроцессорных системах

Управление потоками в МПС

- Операционные системы Windows 2000-2003 руководствуется следующими базовыми стратегиями для решения того, на каких процессорах какие потоки выполняются:
 - выбор процессора для готового потока, при наличии простаивающих процессоров;
 - выбор процессора для готового потока, при отсутствии простаивающих процессоров;
 - выбор потока для конкретного процессора, который освободился.

Выбор процессора для потока при наличии простаивающих

- Как только поток готов к выполнению, Windows сначала пытается подключить его к простаивающему процессору.
- Если таких процессоров несколько, то выбор процессора производится в зависимости от версии операционной системы Windows (см. далее).
- Как только процессор выбран, соответствующий поток переводится в состояние Standby.
- При выполнении на этом процессоре поток просто обнаруживает, что поток выбран и подключает его к процессору.

Выбор процессора для потока

- Выбор процессора для потока в Windows 2000:
 - Если простаивающих процессоров несколько, то предпочтение отдается сначала идеальному процессору для данного потока, затем предыдущему, а потом текущему (т. е. процессору, на котором работает поток простоя, отвечающий за планирование).
 - Если все эти процессоры заняты, операционная система выбирает первый простаивающий процессор, на котором может работать данный поток, для чего сканируется маска свободных процессоров в направлении убывания их номеров.
- Выбор процессора для потока в Windows XP и 2003:
 - Выделяются простаивающие процессоры из числа тех, на которых маска привязки разрешает выполнение данного потока. Если маска привязки разрешает выполнение потока на текущем процессоре, то поток планируется к выполнению именно на этом процессоре.
 - Если маска привязки не разрешает выполнение потока на текущем процессоре, то выполняется следующий алгоритм.

Алгоритм выбор процессора для потока в Windows XP и 2003

- Если система имеет архитектуру NUMA и в узле, где находится идеальный процессор для потока, есть простаивающие процессоры, то список всех простаивающих процессоров уменьшается до этого набора. Если в результате такой операции в списке не останется простаивающих процессоров, список не сокращается.
- Если в системе работают процессоры с технологией HT и имеется физический процессор, все логические процессоры которого свободны, список простаивающих процессоров уменьшается до этого набора. Если в результате такой операции в списке не останется простаивающих процессоров, список не сокращается.
- Если текущий процессор относится к набору оставшихся простаивающих процессоров, поток планируется к выполнению именно на этом процессоре.
- Если текущий процессор не входит в список оставшихся простаивающих процессоров, если это система с технологией HT и если есть простаивающий логический процессор на физическом, который содержит идеальный процессор для данного потока, то список простаивающих процессоров ограничивается этим набором. В ином идет поиск простаивающих логических процессоров на предыдущем физическом процессоре. Если такой набор не пуст, список простаивающих процессоров уменьшается до этого набора.
- Из оставшегося набора простаивающих процессоров исключаются все процессоры, находящиеся в состоянии сна. (Эта операция не выполняется, если в ее результате такой список опустел бы.)
- Поток подключается к процессору с наименьшим номером в оставшемся списке.

Выбор процессора для потока при отсутствии простаивающих

- При отсутствии простаивающего процессора Windows выполняет постановку нового потока на идеальный процессор:
 - Если этот процессор не входит в маску привязки потока, Windows выбирает для потока процессор с наибольшим номером (в Windows 2000 маска привязки может исключить идеальный процессор).
 - Если к идеальному процессору уже подключен какой-то поток, Windows сравнивает приоритеты текущего и нового потока. Если приоритет выполняемого потока меньше, чем нового, то текущий поток вытесняется в пользу нового.
 - Если для идеального процессора уже выбран поток, ожидающий в состоянии Standby выделения процессорного времени, и его приоритет ниже, чем потока, готовящегося к выполнению, последний вытесняет первый и становится следующим выполняемым на данном процессоре.

Выбор потока для конкретного процессора (Windows 2000 и XP)

- В многопроцессорной системе Windows 2000 или Windows XP из очереди готовых выбирается поток с наивысшим приоритетом, для которого выполняется одно из дополнительных условий:
 - поток уже выполнялся в прошлый раз на данном процессоре;
 - данный процессор должен быть идеальным для этого потока;
 - поток провел в состоянии Ready более трех тактов системного таймера;
 - поток имеет приоритет не менее 24.

Выбор потока для конкретного процессора (Windows 2003)

- Поскольку в Windows Server 2003 у каждого процессора собственный список потоков, ждущих выполнения на этом процессоре, то по окончании выполнения текущего потока процессор просто проверяет свою очередь готовых потоков.
- Если его очереди пусты, к процессору подключается поток простоя. Затем этот поток начинает сканировать очереди готовых потоков при других процессорах и ищет потоки, которые можно было бы выполнять на данном процессоре.
- Обратите внимание, что в NUMA-системах поток простоя проверяет процессоры сначала в своем узле, а потом в других узлах.

Управление центральным процессором...

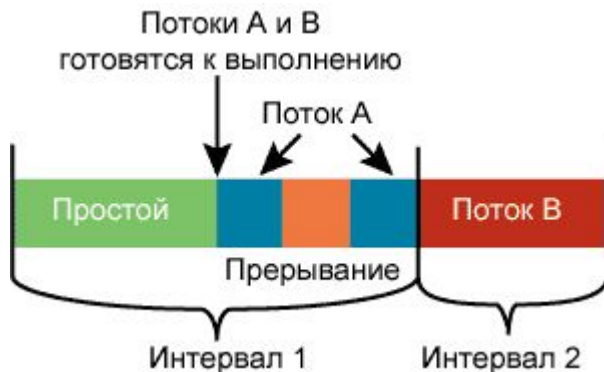
Особенности планирования потоков в Windows Vista и Windows Server 2008

Проблема неравномерного распределения ресурсов процессора

- Все версии ОС Windows собирают статистику загрузки процессора, для этого каждые 10 или 15 мс (в зависимости от аппаратной платформы), выполняется подпрограмма прерывания интервального таймера.
- Данная подпрограмма обновляет статистику загрузки процессора потоком, который был ею прерван. При этом в версиях ОС Windows младше Vista и Server 2008 считается, что этот поток выполнялся весь интервал, хотя в действительности выполнение потока могло начаться перед самым завершением данного интервала.
- Далее возможна ситуация, когда поток может формально получить ресурсы процессора, но не получить возможности выполняться, поскольку вместо этого могли быть выполнены подпрограммы аппаратного или программного прерывания.
- Таким образом, использование подобного механизма сбора статистики загрузки процессора может приводить к нерациональному распределению ресурсов самого процессора.

Иллюстрация неравномерного распределения ресурсов процессора

- Рассмотрим случай, если два потока с одинаковым приоритетом одновременно готовы к выполнению:
 - Поток А выполняется до завершения следующего временного интервала, когда планировщик предполагает, что он выполнялся в течение всего интервала, и поэтому решает, что очередь потока А закончилась. Более того, потоку А засчитывается еще и прерывание, произошедшее во время его выполнения.
 - На следующий интервал планировщик отдает предпочтение потоку В, который выполняется в течение полного интервала.

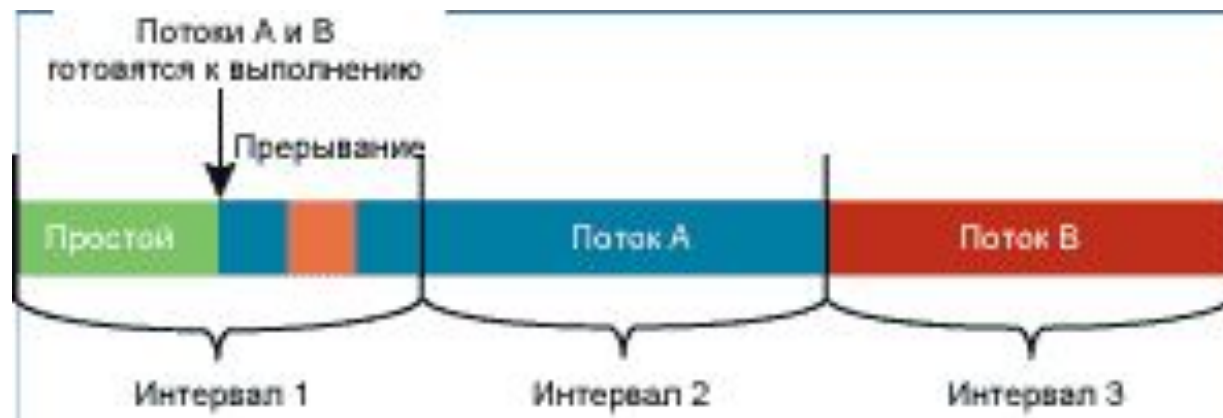


Усовершенствования ядра ОС Windows Vista – Server 2008

- В ОС Windows Vista и Server 2008 для более равномерного выделения ресурсов вводится использование счетчика циклов процессора.
- Планировщик отслеживает точное количество циклов процессора, в течение которых выполняется поток, с помощью регистра счетчика циклов современных процессоров. Определив, сколько циклов может выполнить процессор на протяжении интервала времени, планировщик может точнее раздавать ресурсы процессора. К тому же, планировщик не засчитывает выполнение прерывания во время выполнения потока.
- Это означает, что поток в ОС Windows Vista и Server 2008 всегда получит, по крайней мере, свою очередь выполнения, не превышающую один дополнительный временной интервал, что обеспечивает более справедливое выделение ресурсов и предсказуемое поведение приложений.

Решение неравномерного распределения ресурсов ЦП

- На рисунке показан пример решения описанной ситуации неравномерного распределения ресурсов ЦП - ядро Windows Vista выделяет обоим потокам как минимум, по одному временному интервалу выполнения.



Управление центральным процессором...

Дополнительный материал

Программа Process Monitor версии 1.1



- Программа *Process Monitor* является усовершенствованным инструментом отслеживания для Windows, который в режиме реального времени отображает активность файловой системы, реестра, а также процессов и потоков (<http://technet.microsoft.com/ru-ru/sysinternals/bb896645.aspx>).