**Arch\_Win2000\_2007**

**Архитектура Windows 2000-2003**

*Архитектура Windows NT-2000*

*Основная характеристика ОС семейства MS Windows NT-2000-2003*

Эволюция Windows Server

Сервер файлов и печати уровня департамента



Сервера интранет и центры обработки данных

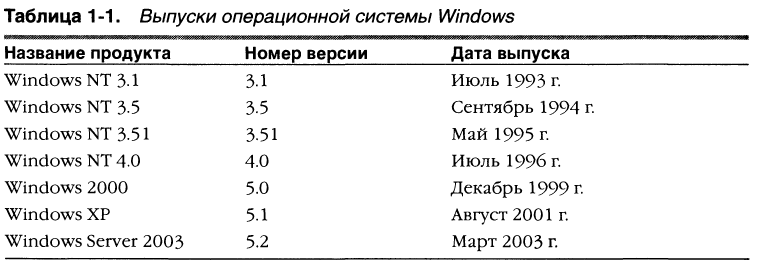


Простота объединения с помощью веб-сервисов





История развития Windows NT



Основная характеристика Windows NT-2000-2003

Система Windows NT-2003 не является дальнейшим развитием ранее существовавших продуктов. Ее архитектура создавалась с нуля с учетом предъявляемых к современной ОС требований:

* *совместимость (compatible);*
* *переносимость (portability);*
* *масштабируемость (scalability);*
* *безопасность (security);*
* *распределенная обработка (distributed processing);*
* *надежность и отказоустойчивость (reliability and robustness);*
* *локализации (localization);*
* *расширяемость (extensibility).*

Совместимость

* *совместимость (compatible)* – поддержка существующих файловых систем, прикладных сред и сетевых интерфейсов. Специальные сервисы для интеграции с UNIX – Windows Services for UNIX;
* *переносимость (portability);*
* *масштабируемость (scalability);*
* *безопасность (security);*
* *распределенная обработка (distributed processing);*
* *надежность и отказоустойчивость (reliability and robustness);*
* *локализации (localization);*
* *расширяемость (extensibility).*

Windows Services for UNIX

* упрощают интеграцию Windows 2000-2003 с существующими UNIX-сетями
* улучшают управляемость, упрощают администрирование сетей и учетных записей
* позволяют продолжить использование существующих UNIX-ресурсов и опыта, накопленного в работе с UNIX-системами

Переносимость

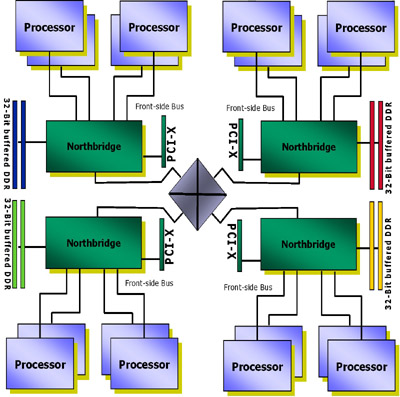
* *совместимость (compatible);*
* *переносимость (portability)* системы, которая работает как на CISC (x86), так и на RISC-процессорах (MIPS R4000 (только NT) и Digital Alpha AXP). ОС MS Windows 2003 поддерживает архитектуру x86 и IA64, AMD x86-64, EM64T.
* *масштабируемость (scalability);*
* *безопасность (security);*
* *распределенная обработка (distributed processing);*
* *надежность и отказоустойчивость (reliability and robustness);*
* *локализации (localization);*
* *расширяемость (extensibility).*



Масштабируемость

* *совместимость (compatible);*
* *переносимость (portability);*
* *масштабируемость (scalability)* означает, что Windows 2000-2003 поддерживает многопроцессорные системы с числом процессоров от 1 до 32.
* *безопасность (security);*
* *распределенная обработка (distributed processing);*
* *надежность и отказоустойчивость (reliability and robustness);*
* *локализации (localization);*
* *расширяемость (extensibility).*

Масштабируемость MS Windows XP-2003

* SMP-системы (оперативная память физически представляет последовательное адресное пространство, доступ к которому имеют одновременно все процессоры системы по единой шине).
* Логические процессоры (hyperthreading)
* NUMA (Non-Uniform Memory Architecture)

NUMA

* Процессоры группируются в узлы (Nodes)
* В каждом узле несколько CPU и память (SMP-система, но за счет минимальной компоновки элементов достигается высокая пропускная способность между процессором и локальной памятью модуля )
* Узлы объединяются шиной

Безопасность

* *совместимость (compatible);*
* *переносимость (portability);*
* *масштабируемость (scalability);*
* Windows 2000-2003 имеет однородную *систему безопасности (security)*, удовлетворяющую стандарту С-2 "Оранжевая книга". В корпоративной среде критическим приложениям обеспечивается полностью изолированное окружение;
* *распределенная обработка (distributed processing);*
* *надежность и отказоустойчивость (reliability and robustness);*
* *локализации (localization);*
* *расширяемость (extensibility).*

Распределенная обработка

* *совместимость (compatible);*
* *переносимость (portability);*
* *масштабируемость (scalability);*
* *безопасность (security);*
* *распределенная обработка (distributed processing)* означает, что Windows 2000-2003 имеет встроенные в систему сетевые возможности (TCP/IP, Netbios, IPX/SPX);
* *надежность и отказоустойчивость (reliability and robustness);*
* *локализации (localization);*
* *расширяемость (extensibility).*

Надежность и отказоустойчивость

* *совместимость (compatible);*
* *переносимость (portability);*
* *масштабируемость (scalability);*
* *безопасность (security);*
* *распределенная обработка (distributed processing);*
* *надежность и отказоустойчивость (reliability and robustness)* обеспечиваются архитектурными особенностями, которые защищают прикладные программы от повреждения друг другом и ОС. Windows 2000-2003 использует отказоустойчивую структурированную обработку особых ситуаций на всех архитектурных уровнях, которая включает восстанавливаемую файловую систему NTFS и обеспечивает защиту с помощью встроенной системы безопасности и усовершенствованных методов управления памятью;
* *локализации (localization);*
* *расширяемость (extensibility).*

Локализация

* *совместимость (compatible);*
* *переносимость (portability);*
* *масштабируемость (scalability);*
* *безопасность (security);*
* *распределенная обработка (distributed processing);*
* *надежность и отказоустойчивость (reliability and robustness);*
* возможности *локализации (localization)* предоставляют средства для работы на национальных языках, что достигается применением стандарта ISO Unicode (разработан Международной организацией по стандартизации);
* *расширяемость (extensibility).*

Расширяемость

* *совместимость (compatible);*
* *переносимость (portability);*
* *масштабируемость (scalability);*
* *безопасность (security);*
* *распределенная обработка (distributed processing);*
* *надежность и отказоустойчивость (reliability and robustness);*
* *локализации (localization);*
* благодаря модульному построению системы обеспечивается *расширяемость (extensibility)* Windows 2000-2003 – гибкое добавление новых модулей на различные уровни ОС.

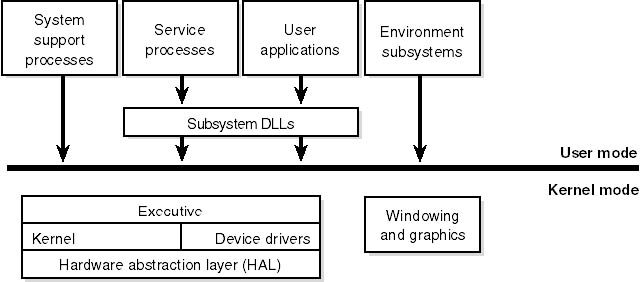
Семейство Windows 2000

* Windows 2000 Professional
  + - До 2-х ЦПУ, 4 ГБ ОЗУ
* Windows 2000 Server
  + - До 4-х ЦПУ, 4 ГБ ОЗУ
* Windows 2000 Advanced Server
  + - До 8-х ЦПУ, 8 ГБ ОЗУ
    - 2-узловая кластеризация
* Windows 2000 Datacenter Server
  + - До 32-х ЦПУ, 64 ГБ ОЗУ
    - 4-узловая кластеризация
* Windows Server 2003 Web Edition
  + - До 2-х ЦПУ, 2 ГБ ОЗУ
* Windows Server 2003 Standard Edition
  + - До 4-х ЦПУ, 4 ГБ ОЗУ
* Windows Server 2000 Enterprise Edition
  + - До 8-х ЦПУ, 32 ГБ ОЗУ
    - 2-узловая кластеризация
* Windows Server 2003 Datacenter Edition
  + - До 32-х ЦПУ, 128 (64) ГБ ОЗУ
    - 4-узловая кластеризация

**Архитектура MS Windows 2000-2003**

*Архитектура ОС Windows 2000-2003*

Упрощенная архитектура Windows 2000



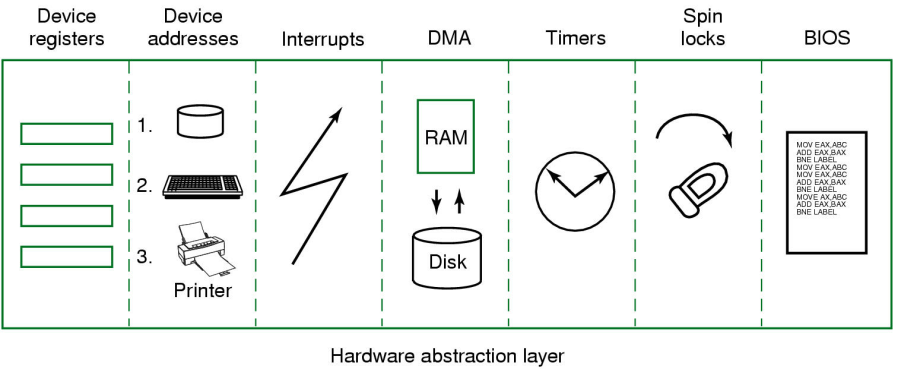
Краткая характеристика структуры

* Многоуровневая ОС
* Ядро работает в защищенном режиме
* Присутствует микроядро, но оно дополнительно не защищено от остального ядра
* Присутствуют элементы объектно-ориентированного подхода
* Важные для производительности ОС компоненты выполняются в режиме ядра, где они взаимодействуют с оборудованием и друг с другом без использования переключателей контекста и смены режимов. Например, менеджер памяти, менеджер кэш-памяти, менеджер объектов, менеджер системы безопасности, сетевые протоколы, файловые системы, управление потоками и процессами работают в режиме ядра. Все эти компоненты полностью защищены от выполняемых приложений, которые не имеют прямого доступа к коду и данным из привилегированной части операционной системы.
* Компоненты режима ядра Windows 2000-2003 спроектированы на основе принципов построения объектно-ориентированных систем. Например, они не работают напрямую со структурами данных, поддерживаемых индивидуальными компонентами. Вместо этого для передачи параметров, доступа и/или модификации структур данных они используют формальный интерфейс. Вместе с тем, несмотря на повсеместное использование объектов для представления разделяемых системных ресурсов, Windows 2000-2003 не является объектно-ориентированной системой в точном смысле этого понятия, поскольку основная часть кода системы написана на Си из соображений обеспечения высокой скорости выполнения и переносимости.

Режим ядра

* **исполняющая система NT,** которая включает управление памятью, процессами, потоками, безопасностью, вводом/выводом, межпроцессорными обменами;
* **ядро (микроядро) Windows NT** выполняет низкоуровневые функции ОС: диспетчеризация потоков, прерываний и исключений, синхронизация процессоров. Ядро также включает набор процедур и базовых объектов, используемый исполняемой частью для создания высокоуровневых конструкций;
* **уровень абстракции от оборудования (HAL – Hardware Abstraction Layer),** изолирует ядро, драйверы устройств и исполняемую часть NT от аппаратных платформ, на которых должна работать ОС. Подобный подход позволяет обеспечить переносимость Windows NT.
* **драйверы устройств** включают как файловую систему, так и аппаратные драйверы, которые транслируют пользовательские вызовы функций ввода/вывода в запросы физических устройств ввода/вывода;
* **функции графического интерфейса пользователя** работают с окнами, элементами управления и рисунками.

Hardware Abstraction Layer (Слой Абстракции Аппаратных средств)

Some of the hardware functions the HAL manages (пер. Некоторые из функций аппаратных средств HAL справляются)

Ядро (микроядро) Windows 2000

*Ядро* (Microkernel) является основой модульного строения ОС и координирует выполнение большинства базовых операций Windows NT. Программное обеспечение ядра выполняется полностью в привилегированном режиме, является неперемещаемым и невыгружаемым.

В отличие от остальной исполняемой части операционной системы, ядро никогда не выгружается из оперативной памяти, его выполнение никогда не прерывается другими потоками.

Код ядра написан в основном на Си, а части, дающие наибольшую нагрузку на процессор, на языке Ассемблере.

Ядро, в первую очередь, занимается планированием действий процессора. В случае если компьютер содержит несколько процессоров, ядро может выполнятся на всех процессорах (SMP) и синхронизирует их работу с целью достижения максимальной производительности системы. Ядро осуществляет диспетчеризацию потоков (threads), таким образом, чтобы максимально загрузить процессоры системы и обеспечить первоочередную обработку потоков с более высоким приоритетом.

Ядро также обеспечивает работу других базовых объектов ядра, которые используются исполняющей системой (и в некоторых случаях экспортируются в режим пользователя).

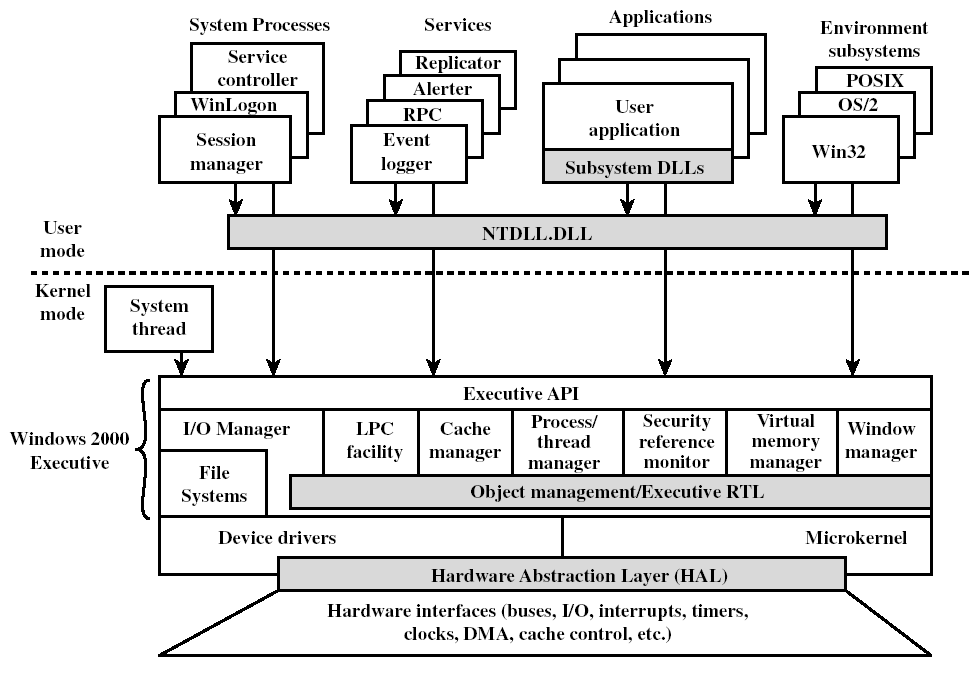
Основные принципы планирования потоков в Windows 2000

* Квантование времени
* Абсолютные приоритеты
* Динамические приоритеты

Пользовательский режим

* **Специальные процессы поддержки системы,** например, процесс регистрации пользователя и менеджер сессий, которые не являются службами NT.
* **Процессы сервера, которые являются службами NT** (аналог демонов в ОС Unix). Примером может быть регистратор событий (Event Logger). Многие дополнительно устанавливаемые приложения, такие как Microsoft SQL Server и Exchange Server, также включают компоненты, работающие как службы NT.
* **Подсистемы среды** представляют собой защищенные серверы пользовательского режима (user-mode), которые обеспечивают выполнение и поддержку приложений, разработанных для различного операционного окружения (различных ОС). Примером подсистем среды могут служить подсистемы Win32, Posix и OS/2 2.1.
* **Пользовательские приложения** одного из пяти типов: Win32, Windows 3.1, MS-DOS, Posix или OS/2 1.2.

Подробная архитектура Windows 2000

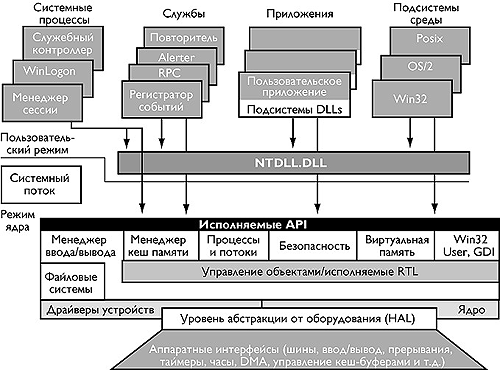
* На слайде приведена общая архитектура Windows 2000 и ее компонентов.
* Элементы над разделительной линией представляют собой процессы пользовательского режима, а под ней располагаются процессы ОС, выполняемые ядром.
* Потоки пользовательского режима выполняются в защищенном адресном пространстве. Однако, во время их выполнения в режиме ядра, они получают доступ к системному пространству. Таким образом, системные процессы, процессы сервера (службы), подсистема среды или пользовательское приложение имеют свое собственное адресное пространство.
* Hardware abstraction layer (HAL) *(Слой абстракции аппаратных средств)*
* Microkernel *(Микроядро)*
* Device drivers *(Драйверы устройства)*

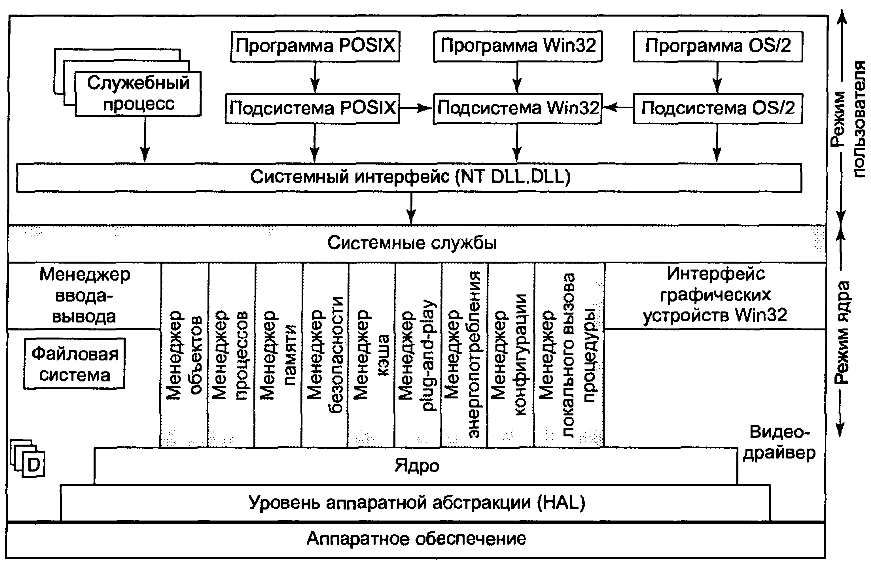
W2K Executive *(Руководитель)*

* I/O manager *(Менеджер ввода / вывода)*
* Object manager *(Возразите менеджеру)*
* Security reference monitor *(Ссылка безопасности контролирует)*
* Process/thread manager *(Обработайте/пронизывайте менеджера)*
* Local procedure call (LPC) Facility *(Местный вызов процедуры (LPC) Средство)*
* Virtual memory manager *(Менеджер виртуальной памяти)*
* Cache manager *(Менеджер тайника)*
* Windows/graphics modules *(Пользовательские процессы модулей Windows/графики)*

User Processes

* Special system support processes (процессы поддержки процессов)
* Server processes (процессы сервисов)
* Environment subsystems (подсистемы окружения)
* User applications (пользовательские приложения)





Исполняющая система

* *Исполняющая система* включает в свой состав набор программных конструкций привилегированного режима (kernel-mode), предоставляющих базовый сервис ОС подсистемам среды.
* Исполняющая система состоит из нескольких компонентов; каждая из них предназначена для поддержки определенного системного сервиса.
* **Менеджер процессов и потоков управляет процессами и потоками.** Фактически потоки и процессы поддерживаются в NT нижележащим слоем. Исполняемая часть добавляет дополнительную семантику и функции к этим объектам нижнего уровня.
* **Менеджер виртуальной памяти** использует схему управления, при которой каждый процесс получает собственное достаточно большое адресное пространство, защищенное от воздействия других процессов. Менеджер памяти также обеспечивает низкоуровневую поддержку для менеджера кэш-памяти.
* **Монитор безопасности** проводит политику обеспечения мер безопасности на локальном компьютере, охраняя системные ресурсы и выполняя процедуры аудита и защиты объектов.
* **Диспетчер ввода/вывода** использует независимый от устройств ввод/вывод и отвечает за пересылку данных соответствующим драйверам для дальнейшей обработки.
* **Менеджер кэш-памяти** улучшает производительность системы ввода/вывода файлов, размещая читаемые с диска данные в основной памяти для ускорения доступа к ним, а также откладывая на короткое время запись измененных данных на диск.
* **Менеджер объектов,** который создает, удаляет объекты и абстрактные типы данных, а также управляет ими. Объекты используются в Windows NT для представления таких ресурсов операционной системы, как процессы, потоки и объекты синхронизации.
* **Диспетчер Plug&Play –** определяет какие драйвера нужны для конкретного устройства и загружает их.
* **Диспетчер конфигурации** отвечает за организацию и управление системным реестром.
* **Диспетчер электропитания –** координирует события, связанные с электропитанием, генерирует уведомления о событиях
* **LPC** (Local Procedure Call) передает сообщения между клиентским процессом и процессом сервера на том же самом компьютере. По сути, LPC – это оптимизированная версия известной процедуры удаленного вызова RPC (Remote Procedure Call).
* Широкий **набор библиотечных функций общего типа:** обработка строк, арифметические операции, преобразование типов данных, обработка структур.
* **Процедуры распределения памяти,** взаимообмен между процессами через память, два специальных типа объектов синхронизации – ресурсы и объекты fast mutex.

*Объекты Windows 2000-2003*

Объекты MS Windows

В ОС Windows 2000-2003 объект – это отдельный экземпляр периода выполнения (runtime instance) статически определенного типа объекта.

***Тип объектов*** (object type), иногда называемый *классом объектов* (object class) состоит из общесистемного типа данных, функций, оперирующих экземплярами этого типа данных, и набора атрибутов.

***Атрибут объекта*** *(object attribute) –* это поле данных внутри объекта частично определяющее его состояние.

***Методы объекта (средства для манипулирования объектами)*** обычно считывают атрибуты объекта.

* Не все структуры данных в Windows 2000-2003 являются объектами. В объекты помещаются лишь те данные, которые нужно разделять, защищать, именовать или сделать доступными программам пользовательского режима (через системные сервисы).
* Структуры, используемые только одним из компонентов операционной системы для поддержки каких-то внутренних функций, к объектам не относятся.

***MS Windows 2000 – 27 объектов***

***MS Windows XP-2003 – 29 объектов***

Назначение объектов

Объекты очень удобны для поддержки четырех важных функций ОС:

* присвоения понятных имен системным ресурсам;
* разделения ресурсов и данных между процессами;
* защиты ресурсов от несанкционированного доступа;
* учета ссылок (благодаря этому система узнает, когда объект больше не используется, и автоматически уничтожает его).

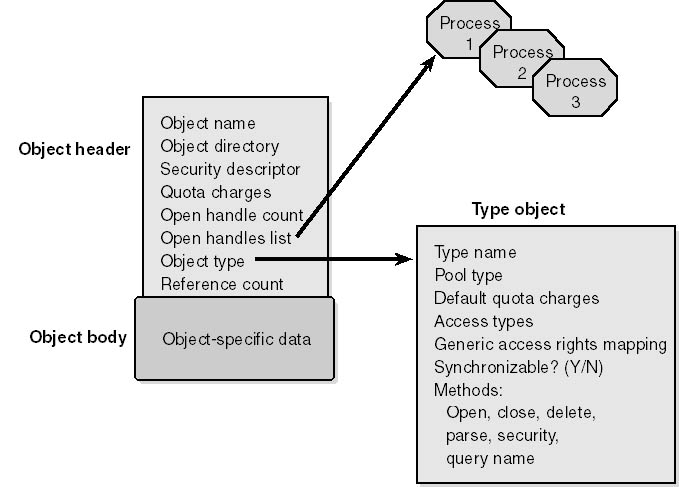
Типы объектов Windows 2000-2003

* **Объектыисполнительной системы** (executive object) представляются различными компонентами исполнительной системы. Они доступны программам пользовательского режима (защищенным подсистемам)посредством базовых сервисов и могут создаваться и использоваться как подсистемами, так и исполнительной системой.
* **Объекты ядра** (kernel object) – это более примитивный набор объектов, реализованный ядром. Большинство этих объектов создаются и используются только внутри исполнительной системы.
  + Управляющие объекты (объекты прерываний, … )
  + Объекты диспетчеризации (семафоры, события, мьютексы, таймеры,…)

Примеры объектов

* Файл
* Регион памяти
* Поток
* Процесс
* Семафор
* Таймер

Структура объектов Windows 2000-2003



|  |  |
| --- | --- |
| Имя объекта | Делает объект видимым другим процессам для совместного использования |
| Каталог объектов | Обеспечивает иерархическую структуру, в которой хранятся имена объектов |
| Дескриптор  безопасности | Определяет, кто и каким образом может использовать данный объект |
| Расход квоты | Задает квоту на использование ресурсов, которая списывается с процесса при открытии описателя данного объекта |
| Счетчик открытых дескрипторов | Подсчитывает количество открытых дескрипторов данного объекта |
| Список открытых дескрипторов | Содержит список процессов, открывших дескрипторы данного объекта |
| Временный/ постоянный статус | Указывает, можно ли уничтожить имя и освободить память объекта, если он более не используется |
| Режим: пользовательский/ ядра | Определяет доступность объекта в пользовательском режиме |
| Указатель на типовой объект | Ссылается на типовой объект, который содержит атрибуты, общие для набора однотипных объектов |

Удержание объектов

Удержание объектов включает две фазы. Первая фаза называется **удержанием имени** (name retention) и управляется количеством открытых дескрипторов данного объекта. Всякий раз, когда процесс открывает описатель объекта, диспетчер объектов увеличивает счетчик открытых дескрипторов в заголовке объекта.

После того, как процесс закончил работу с объектом и закрыл имеющиеся у него дескрипторы данного объекта, диспетчер объектов уменьшает счетчик.

Вторая фаза удержания объектов – это **прекращение удержания** (т.е. удаление объектов), когда они более не используются.

Учет использования ресурсов

Каждому пользователю назначаются **предельные размеры квот**, ограничивающие суммарный объем системной памяти, который может быть использован его процессами. Соответственно, заголовок каждого объекта содержит атрибут, называемый "расход квоты" и содержащий значение, которое диспетчер объектов вычитает из выделенной процессу квоты, когда поток этого процесса открывает дескриптор данного объекта.

Защита объектов

ОС Windows 2000 поддерживает два вида контроля доступа к объектам:

* **управление избирательным доступом**(discretionary access control) – основной механизм контроля доступа, при котором владельцы объектов разрешают или запрещают доступ к ним для других пользователей (процессов). Когда процесс пытается использовать какой-либо объект, система проводит сравнение атрибутов безопасности, находящихся в идентификаторе безопасности маркера доступа с каждым элементом контроля доступа (access control element, ACE) в списке контроля доступа (access control list, ACL) самого объекта.
* **управление привилегированным доступом**(privileged access control) – необходим в тех случаях, когда управления избирательным доступом недостаточно. Данный метод гарантирует, что пользователь сможет обратиться к защищенным объектам, даже если их владелец недоступен.
* Когда процесс пытается использовать какой-либо объект, система проводит сравнение атрибутов безопасности, находящихся в идентификаторе безопасности маркера доступа с каждым элементом контроля доступа (ACE) в списке контроля доступа (ACL) самого объекта.
* Проверяется каждый элемент ACE до тех пор, пока не будет найдено элемента, гарантирующего требуемый доступ для данного маркера доступа, или не будет найден элемент, запрещающий такой доступ или не будет достигнут конец списка ACL.
* Как правило, дескриптор безопасности может содержать несколько элементов ACE, — в этом случае права, гарантируемые каждой записью, суммируются. Например, если один элемент ACE гарантирует доступ по чтению группе, а другой доступ по записи пользователю-владельцу маркера, который также принадлежит к этой группе, то данный пользователь будет иметь доступ к объекту и по чтению, и по записи.

Отношения между маркером доступа и атрибутами безопасности объекта

