Операционные среды, системы и оболочки

Лектор : кандидат физико-математических наук, доцент
Кишкан Владимир Владимирович

Тема 1. Введение. Назначение, функции и архитектура операционных систем. Основные определения и понятия

1.1. Определение операционной системы (ОС). Место ОС в программном обеспечении вычислительных систем
1.2. Эволюция операционных систем
1.3. Назначение, состав и функции ОС
1.4. Архитектуры операционных систем
1.5. Классификация операционных систем
1.6. Эффективность и требования, предъявляемые к ОС
1.7. Множественные прикладные среды. Совместимость
1.8. Инсталляция и конфигурирование операционных систем

1.1. Определение операционной системы (ОС). Место ОС в программном обеспечении вычислительных систем

1946 г. – ENIAC (Electronic Numerical Integrator and Computer) – полное отсутствие какого-либо ПО, программирование путем коммутации устройств.

Начало 50-х г. – появление алгоритмических языков и системного ПО.

Усложнение процесса выполнения программ:

1. Загрузка нужного транслятора.
2. Запуск транслятора и получение программы в машинных кодах.
3. Связывание программы с библиотечными подпрограммами.
4. Запуск программы на выполнение.
5. Вывод результатов работы на печатающее или другое устройство.

Для повышения эффективности использования ЭВМ вводятся операторы, затем разрабатываются управляющие программы – мониторы - прообразы операционных систем.

1952 г. – Первая ОС создана исследовательской лабораторией фирмы General Motors для IBM-701.

1955 г. – ОС для IBM-704. Конец 50-х годов: язык управления заданиями и пакетная обработка заданий.

1963 г. – ОС MCP (Главная управляющая программа) для компьютеров B5000 фирмы Burroughs: мультипрограммирование, мультипроцессорная обработка,виртуальная память, возможность отладки программ на языке исходного уровня, сама ОС написана на языке высокого уровня.

1963 г. – ОС CTSS (Compatible Time Sharing System – совместимая система разделения времени для компьютера IBM 7094 – Массачусетский технологический институт.

1963 г. – ОС MULTICS (Multiplexed Information and Computing Service) – Массачусетский технологический институт.

1974 г. – (UNICS) UNIX (Uniplexed Information and Computing Service) для компьютера PDP-7, публикация статьи Ритчи (C) и Томпсона.

1981 г. – PC (IBM), DOS (Seattle Computer Products) – MS DOS (Б. Гейтс).

1983г. – Apple, Lisa с Apple, Lisa с GUI (Даг Энгельбарт – Стэнфорд).

1985 г. – Windows, X Windows и Motif (для UNIX).

1987 г. – MINIX (Э. Таненбаум) – 11800 стр. С и 800 ассемблер (микроядро – 1600 С и 800 ассемблер)

1991 г. – Linux (Линус Торвальдс).

Расположение ОС в иерархической структуре программного и аппаратного обеспечения компьютера

ОПЕРАЦИОННАЯ СИСТЕМА

- это набор программ, контролирующих работу прикладных программ и системных приложений и исполняющих роль интерфейса между пользователями, программистами, приложениями и аппаратным обеспечением компьютера.

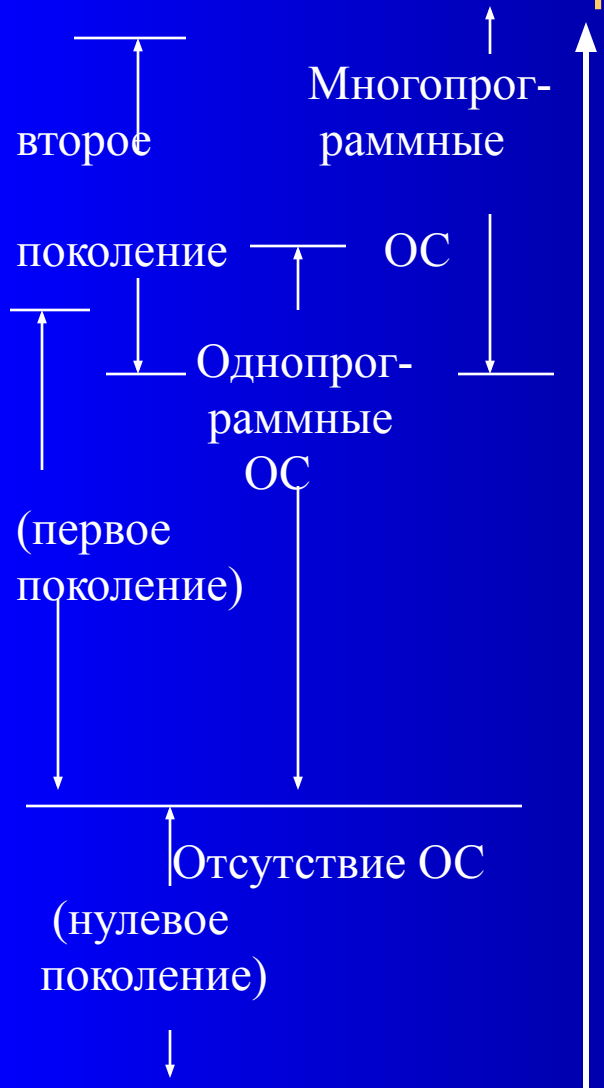
ОПЕРАЦИОННАЯ СРЕДА

- программная среда, образуемая операционной системой, определяющая интерфейс прикладного программирования (API) как множество системных функций и сервисов (системных вызовов), предоставляемых прикладным программам.

ОПЕРАЦИОННАЯ ОБОЛОЧКА

- часть операционной среды, определяющая интерфейс пользователя, его реализацию (текстовый, графический и т.п.), командные и сервисные возможности пользователя по управлению прикладными программами и компьютером

1.2. Эволюция операционных систем



1970 Динамическое распределение основной памяти
Разделение времени, многотерминальные системы
UNIX (PDP-7), Ken Thompson

1965 Управляемое мультипрограммирование
Классическое мультипрограммирование, OS/360
ОС CTSS (1963), MULTICS (начало работ)
Оверлейные структуры
Логическая система управления вводом-выводом

1960 Системы прерываний, контрольные точки
Управление файлами, таймеры
Спулинг (SPOOL)
Мониторы

1955 Методы доступа, полибуферизация
Загрузчики, редакторы связей

1950 Диагностические программы
Ассемблеры, макрокоманды
Библиотеки подпрограмм

1946 Первый компьютер

2007 Windows Vista, Windows 7

2005 Windows 2003, 64-разрядная

2003 Windows 2003
.NET Framework, MAC OS X

2000 Windows 2000
Windows 4.0 – 1996

1995 Windows 95
Корпоративные информационные системы
NetWare 4.0 – 93, Windows NT 3.1 – 93
Linux 0.01 - 1993

1990 MINIX – 87 (11800 стр. C + 800 стр. Asm.)
OS/2 - 87

1985 OS-Net (Novell) - 83, MS-Net - 84, Windows 1.0 – 85
Интернет (1983), Персональные компьютеры (1981)
MS DOS 1.0 – (1981)

1980 Сети ЭВМ, UNIX, TCP/IP
Локальные сети

1975 SNA (System Network Architecture), MULTICS
Протокол X.25, телеобработка, базы данных
Виртуальная ЭВМ, Виртуальная память

Операционные системы IBM

1. BPS/360 (Базовая программная поддержка)
2. BOS/360 (Базовая операционная система)
3. TOS/360 (Ленточная операционная система)
4. DOS/360 (Дисковая операционная система)
5. OS/360 – PCP (Первичная управляющая программа)
6. OS/360 – MFT (Мультипрограммирование с фиксированным числом задач)
7. OS/360 – MVT (Мультипрограммирование с переменным числом задач)
8. OS/360 – VMS (Система с переменной памятью)
9. CP-67/CMS (Управляющая программа 67/ диалоговая мониторная система)
10. DOS/VS (Дисковая виртуальная система)
11. OS/VS1 (Виртуальная система 1)
12. OS/VS2 (Виртуальная система 2)
13. VM/370 (Виртуальная машина)

1.3. Назначение, состав и функции ОС

Назначение

1. Обеспечение удобного интерфейса [приложения, пользователь] - компьютер за счет предоставляемых сервисов:

- 1.1. Инструменты для разработки программ
- 1.2. Автоматизация исполнения программ
- 1.3. Единообразный интерфейс доступа к устройствам ввода-вывода
- 1.4. Контролируемый доступ к файлам
- 1.5. Управление доступом к совместно используемой ЭВМ и ее ресурсам
- 1.6. Обнаружение ошибок и их обработка
- 1.7. Учет использования ресурсов

2. Организация эффективного использования ресурсов ЭВМ

- 2.1. Планирование использования ресурса
- 2.2. Удовлетворение запросов на ресурсы
- 2.3. Отслеживание состояния и учет использования ресурса
- 2.4. Разрешение конфликтов между процессами, претендующими на одни и те же ресурсы

3. Облегчение процессов эксплуатации аппаратных и программных средств вычислительной системы

3.1. Широкий набор служебных программ (утилит), обеспечивающих резервное копирование, архивацию данных, проверку, очистку, дефрагментацию дисковых устройств и др.

3.2. Средства диагностики и восстановления работоспособности вычислительной системы и операционной системы:

- диагностические программы для выявления ошибок в конфигурации ОС;
- средства восстановления последней работоспособной конфигурации;
- средства восстановления поврежденных и пропавших системных файлов и др.

4. Возможность развития

4.1. Обновление и возникновение новых видов аппаратного обеспечения

4.2. Новые сервисы

4.3. Исправления (обнаружение программных ошибок)

4.4. Новые версии и редакции ОС

Состав компонентов и функции операционной системы:

1. Управление процессами
2. Управление памятью
3. Управление файлами
4. Управление внешними устройствами
5. Защита данных
6. Администрирование
7. Интерфейс прикладного программирования
8. Пользовательский интерфейс


1.4. Архитектуры операционных систем

ОСНОВНЫЕ ПРИНЦИПЫ РАЗРАБОТКИ АРХИТЕКТУРЫ ОПЕРАЦИОННЫХ СИСТЕМ:

1. Концепция многоуровневой иерархической вычислительной системы (виртуальной машины) с ОС многослойной структуры.
2. Разделение модулей ОС по функциям на две группы: ядро – модули, выполняющие основные функции ОС, и модули, выполняющие остальные (вспомогательные) функции.
3. Разделение модулей ОС по размещению в памяти вычислительной системы: резидентные, постоянно находящиеся в оперативной памяти, и транзитные, загружаемые в оперативную память только на время выполнения своих функций.
4. Реализация двух режимов работы вычислительной системы: привилегированного режима (режима ядра – kernel mode) или режима супервизора (supervisor) и пользовательского режима (user mode) или режима задача (task mode).
5. Ограничение функций ядра (а, следовательно и числа его модулей) до минимально необходимых функций.

6. Модульное строение (однократно используемые – при загрузке ОС) и повторно используемые (привилегированные – не допускают прерываний, реентерабельные – допускают прерывания и повторный запуск, повторновходимые – допускают прерывания после завершения секций).

7. Параметрическая универсальность. Возможность генерации ОС и создания нескольких рабочих конфигураций.

8. Функциональная избыточность.

9. Функциональная избирательность.

10. Открытость, модифицируемость, расширяемость (возможность получения текстов исходных модулей).

11. Мобильность – возможность переноса на различные аппаратные платформы.

12. Совместимость – возможность выполнения приложений, рассчитанных на другие ОС.

13. Безопасность – защита от несанкционированного доступа, защита легальных пользователей друг от друга, аудит, возможность восстановления ОС после сбоев и отказов.

Модульно – интерфейсный подход (структурный подход)

1. Декомпозиция системы на на модули по структурному или функциональному признаку.
2. Модули и их взаимные связи образуют абстракцию системы высокого уровня.
3. Описывается каждый модуль и определяется его интерфейс.
4. Проводится декомпозиция каждого модуля и т. д.

Спецификации модулей и их интерфейсов дают структурную основу для проектирования каждого модуля и всей системы в целом.

Правильное определение и выделение модулей представляет собой сложную задачу. Тесно связанные между собой части системы должны входить в один и тот же модуль.

Разработчики программного обеспечения начинают работу с очень грубого и неполного наброска схемы системы и преждевременно обращают внимание на детали отдельных модулей. Поэтому решения, влияющие на систему глобальным образом, принимаются не из тех предпосылок, из которых нужно и без ясного понимания их последствий.

Преждевременная реализация приводит к неустойчивости программного обеспечения, которая часто требует огромных усилий по поддержанию системы.

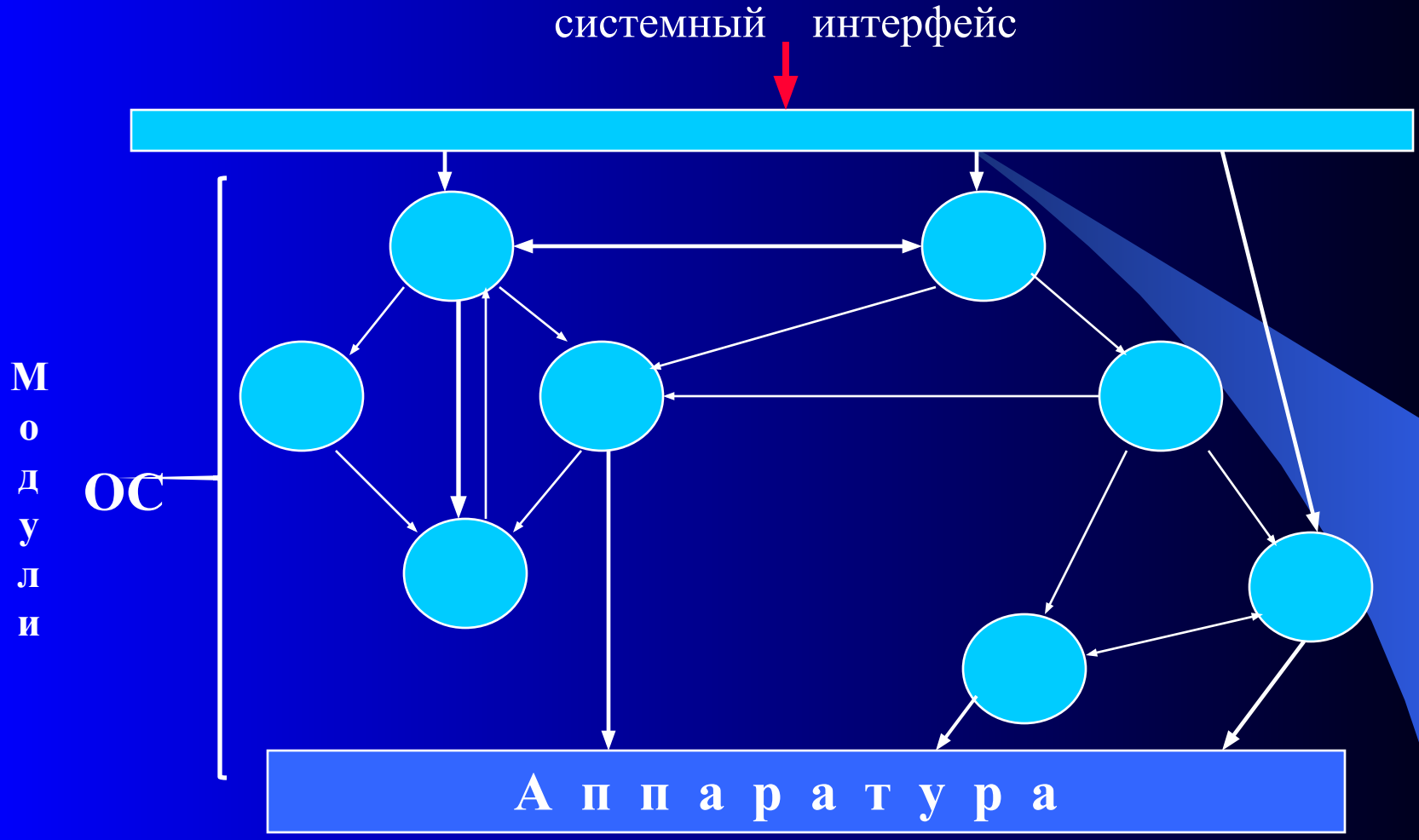
Многослойная (иерархическая) структура операционной системы и метод проектирования «сверху вниз» и «снизу вверх»

1. Операционная система представляется в виде иерархии слоев.
2. Верхний слой определяет виртуальную машину с желаемыми свойствами.
3. Каждый следующий слой детализирует вышележащий, выполняя для него некоторый набор функций.
4. Межслойные интерфейсы подчиняются строгим правилам. Связи внутри слоя могут быть произвольными.
5. Отдельный модуль слоя L(i) может выполнить работу самостоятельно или последующим вариантам: обратиться только к слою L(i –1); обратиться к некоторой команде определенного слоя L(q), который выполняет требуемую функцию (i – 2 <= q <= 0); обратиться к любому последующему слою L(s), (i – 2 <= s <= 0).

Достоинства:

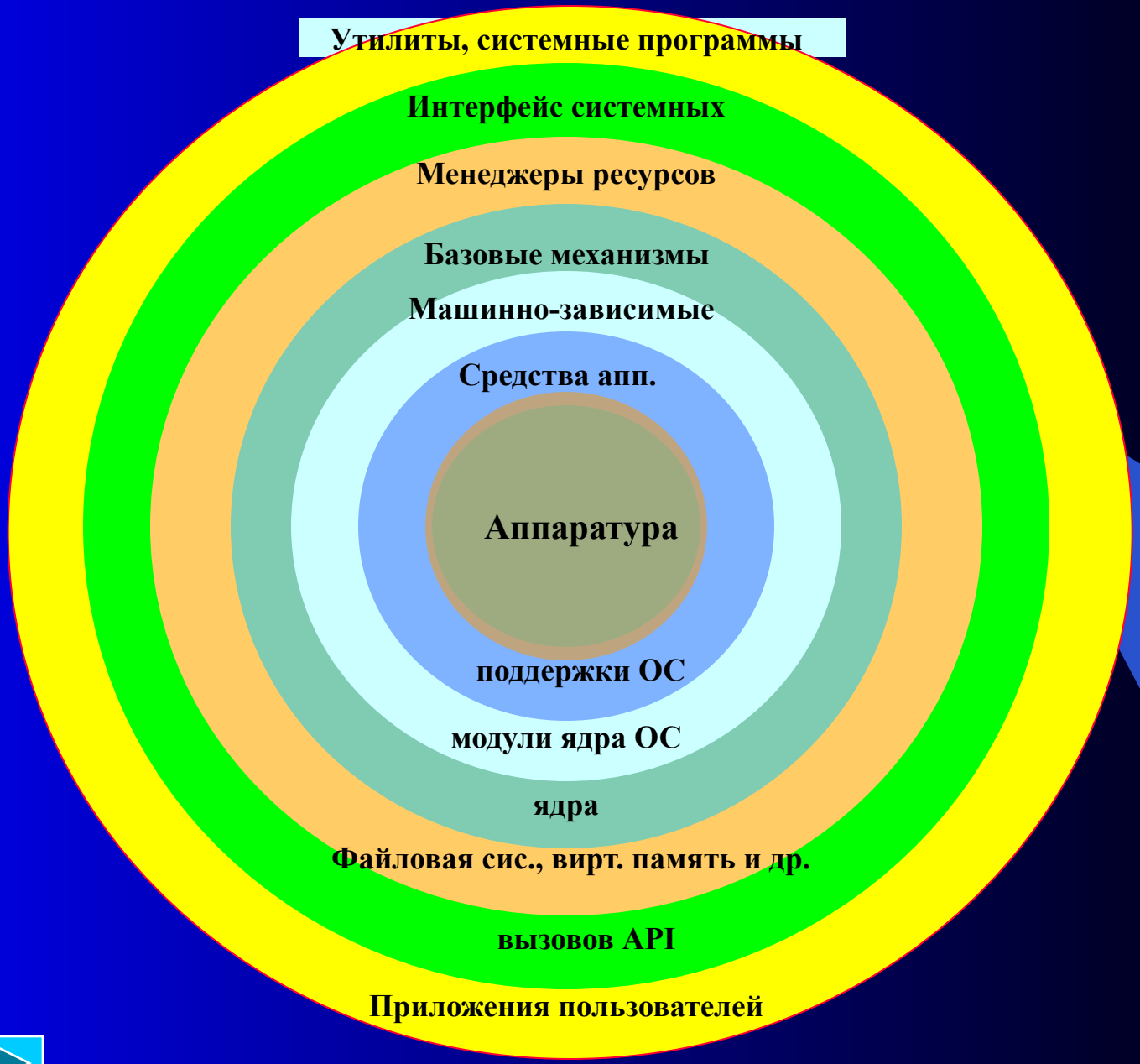
1. Между уровнями можно организовать четкий интерфейс.
2. Систему можно спроектировать методом «сверху вниз», а реализовать методом «снизу вверх».
3. Уровни реализуются в соответствии с их порядком, начиная с аппаратуры и далее вверх.
4. Каждую новую виртуальную машину можно детально проверить, после чего продолжать дальнейшую работу.
5. Любой слой достаточно просто модифицировать, не затрагивая другие слои и не меняя межслойные интерфейсы.

Монолитная архитектура операционной системы

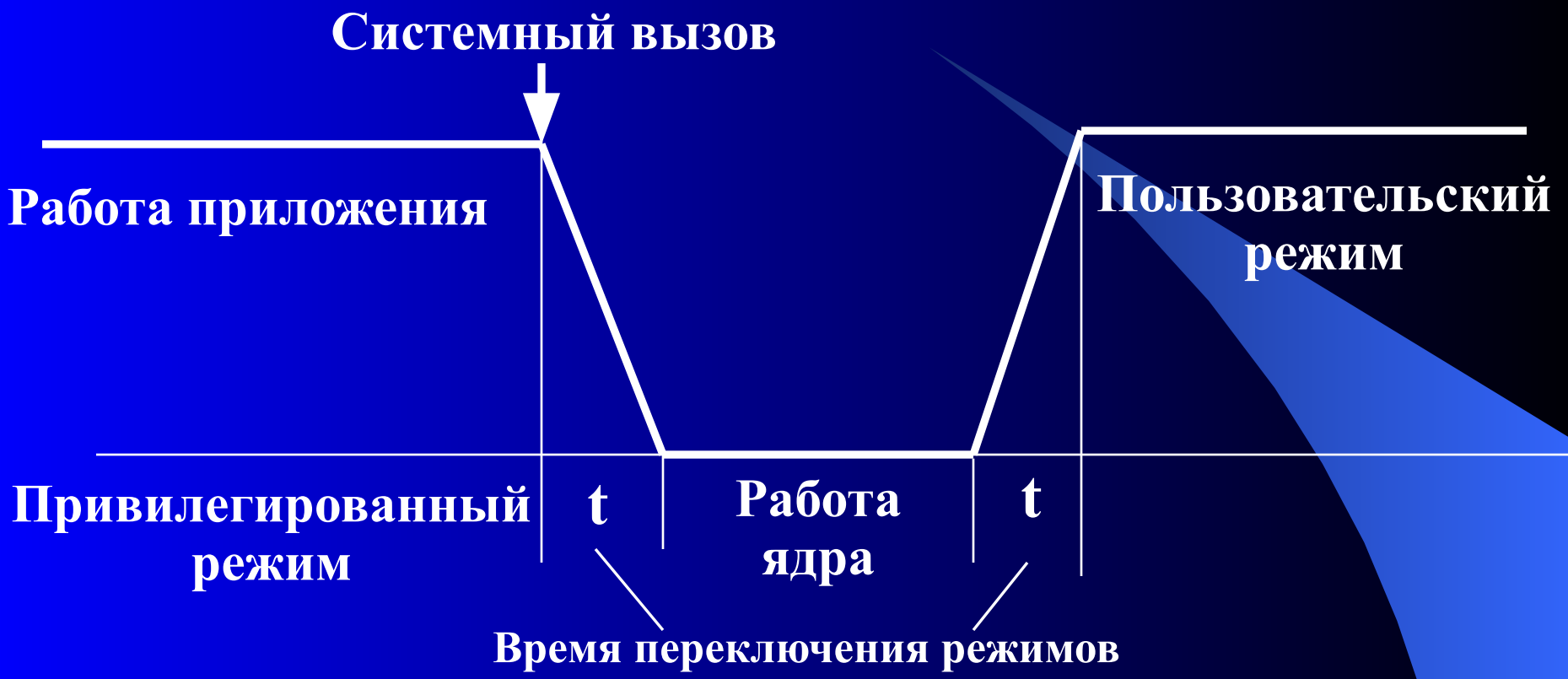


Пример: ранние версии ядра UNIX, Novell NetWare. Каждая процедура имеет хорошо определенный интерфейс в терминах параметров и результатов и может любую другую для выполнения нужной работы.

АРХИТЕКТУРА МНОГОУРОВНЕВОЙ ОПЕРАЦИОННОЙ СИСТЕМЫ



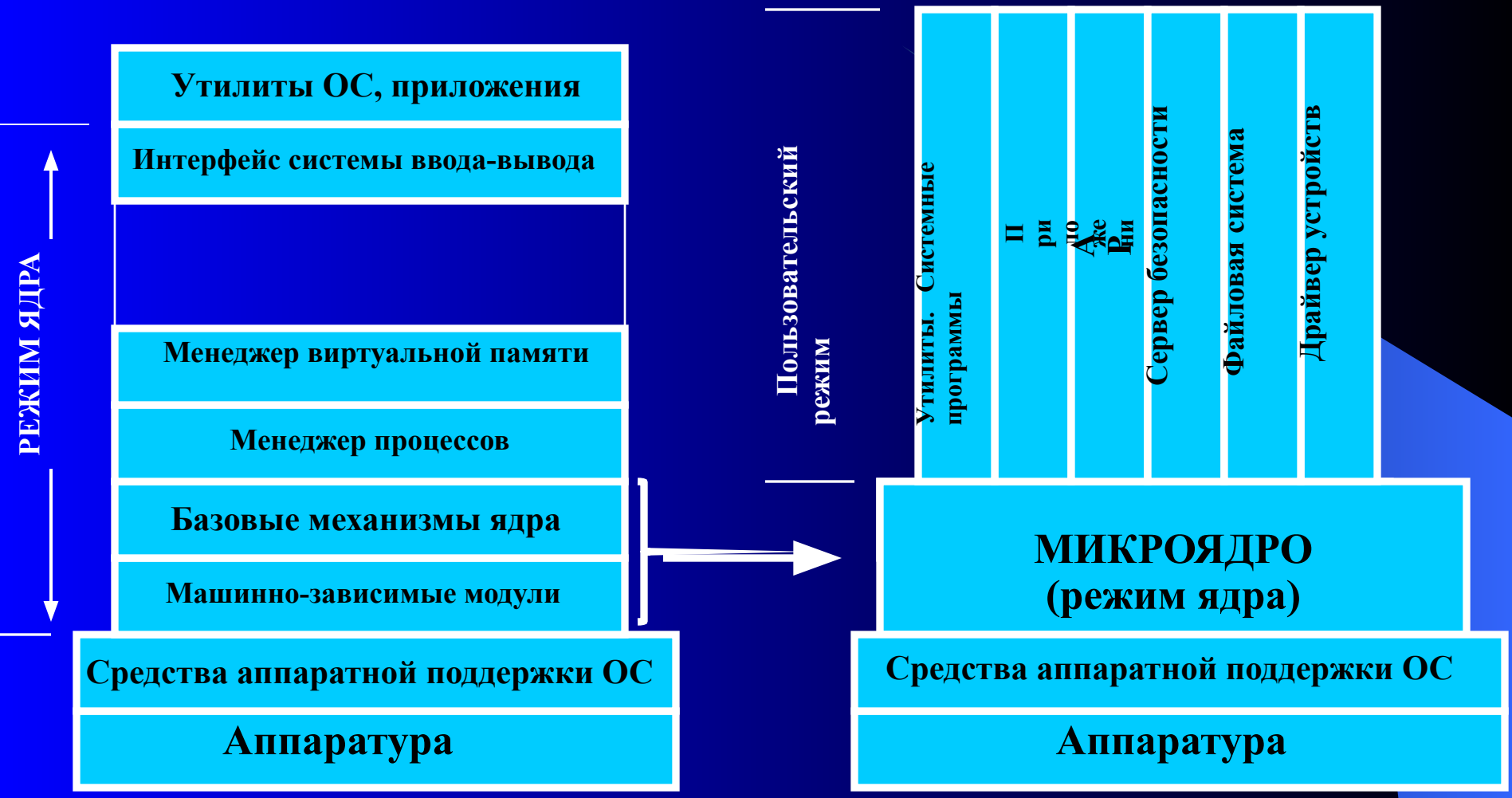
Смена режимов при выполнении вызова функции ядра



Недостатки иерархической организации ОС:

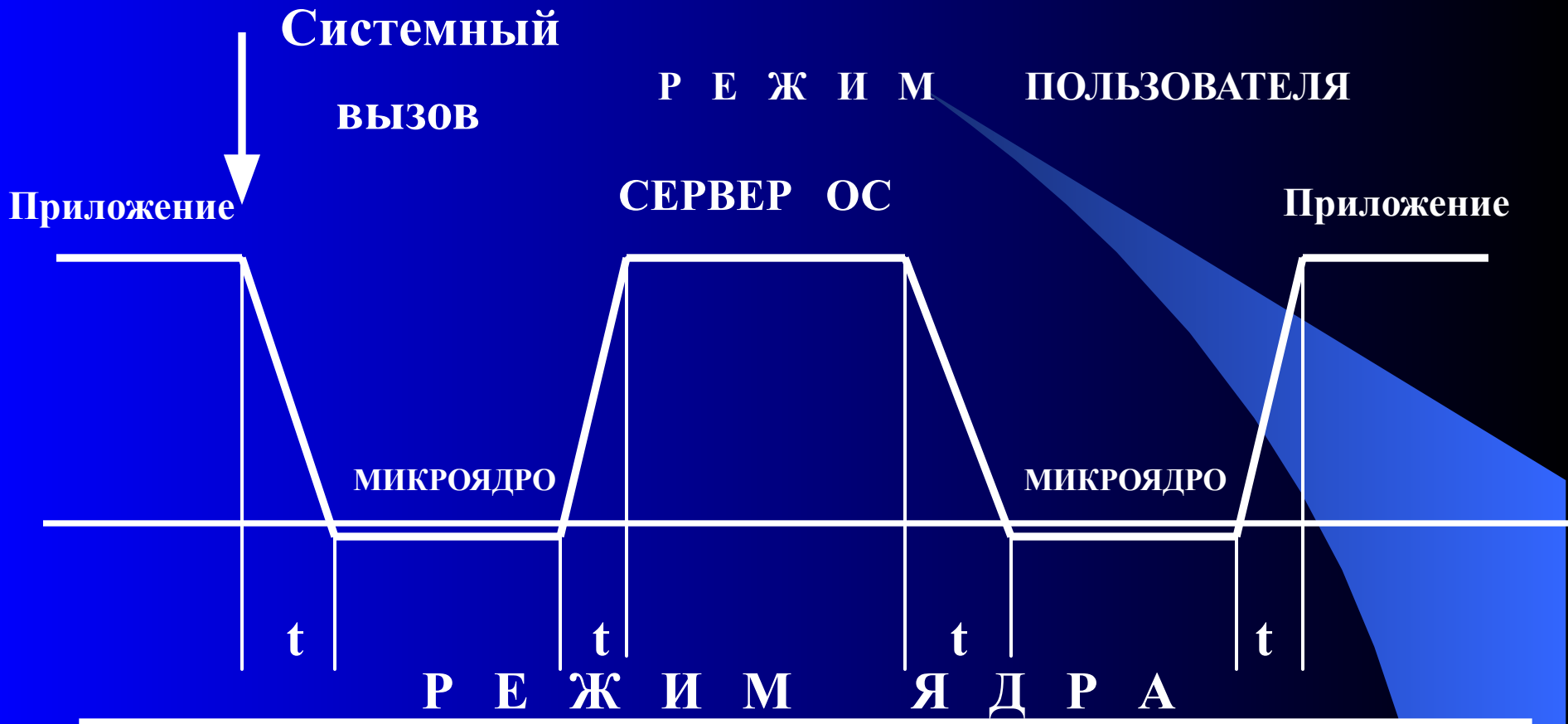
1. Значительные изменения одного из уровней могут иметь трудно предвидимое влияние на смежные уровни.
2. Многочисленные взаимодействия между соседними уровнями усложняют обеспечение безопасности.

Микроядерная архитектура ОС



Структура ОС клиент-сервер

Смена режимов при выполнении вызова функции микроядра



Достоинства: единообразные интерфейсы, расширяемость, гибкость, переносимость, надежность, поддержка распределенных систем, поддержка объектно-ориентированных ОС.

Классификация ядер операционных систем

1. **Наноядро (НЯ)** – крайне упрощённое и минимальное ядро, выполняет лишь одну задачу – обработку аппаратных прерываний, генерируемых устройствами компьютера. После обработки посылает информацию о результатах обработки вышележащему программному обеспечению. Концепция наноядра близка к концепции HAL. НЯ используются для виртуализации аппаратного обеспечения реальных компьютеров или для реализации механизма гипервизора.

2. **Микроядро (МЯ)** предоставляет только элементарные функции управления процессами и минимальный набор абстракций для работы с оборудованием. Бóльшая часть работы осуществляется с помощью специальных пользовательских процессов, называемых сервисами. В микроядерной операционной системе можно, не прерывая ее работы, загружать и выгружать новые драйверы, файловые системы и т. д. Микроядерными являются ядра ОС Minix и GNU Hurd и ядро систем семейства BSD.

3. **Экзоядро (ЭЯ)** – предоставляет лишь набор сервисов для взаимодействия между приложениями, а также необходимый минимум функций, связанных с защитой: выделение и высвобождение ресурсов, контроль прав доступа, и т. д. ЭЯ не занимается предоставлением абстракций для физических ресурсов – эти функции выносятся в библиотеку пользовательского уровня (так называемую libOS). В отличие от микроядра ОС, базирующиеся на ЭЯ, обеспечивают большую эффективность за счет отсутствия необходимости в переключении между процессами при каждом обращении к оборудованию.

4. Монолитное ядро (МЯ) предоставляет широкий набор абстракций оборудования. Все части ядра работают в одном адресном пространстве. МЯ требуют перекомпиляции при изменении состава оборудования. Компоненты операционной системы являются не самостоятельными модулями, а составными частями одной программы. МЯ более производительно, чем микроядро, поскольку работает как один большой процесс. МЯ является большинство Unix-систем и Linux. Монолитность ядер усложняет отладку, понимание кода ядра, добавление новых функций и возможностей, удаление ненужного, унаследованного от предыдущих версий, кода. «Разбухание» кода монолитных ядер также повышает требования к объёму оперативной памяти.

5. Модульное ядро (Мод. Я) – современная, усовершенствованная модификация архитектуры МЯ. В отличие от «классических» МЯ, модульные ядра не требуют полной перекомпиляции ядра при изменении состава аппаратного обеспечения компьютера. Вместо этого они предоставляют тот или иной механизм подгрузки модулей, поддерживающих то или иное аппаратное обеспечение (например, драйверов). Подгрузка модулей может быть как динамической, так и статической (при перезагрузке ОС после переконфигурирования системы). **Мод. Я** удобнее для разработки, чем традиционные монолитные ядра. Они предоставляют программный интерфейс (API) для связывания модулей с ядром, для обеспечения динамической подгрузки и выгрузки модулей. Не все части ядра могут быть сделаны модулями. Некоторые части ядра всегда обязаны присутствовать в оперативной памяти и должны быть жёстко «вшиты» в ядро.

6. Гибридное ядро (ГЯ) – модифицированные микроядра, позволяющие для ускорения работы запускать «несущественные» части в пространстве ядра. Имеют «гибридные» достоинства и недостатки. Примером смешанного подхода может служить возможность запуска операционной системы с монолитным ядром под управлением микроядра. Так устроены 4.4BSD и MkLinux, основанные на микроядре Mach. Микроядро обеспечивает управление виртуальной памятью и работу низкоуровневых драйверов. Все остальные функции, в том числе взаимодействие с прикладными программами, осуществляется монолитным ядром. Данный подход сформировался в результате попыток использовать преимущества микроядерной архитектуры, сохраняя по возможности хорошо отлаженный код монолитного ядра.

Наиболее тесно элементы микроядерной архитектуры и элементы монолитного ядра переплетены в ядре Windows NT. Хотя Windows NT часто называют микроядерной операционной системой, это не совсем так. Микроядро NT слишком велико (более 1 Мбайт), чтобы носить приставку «микро». Компоненты ядра Windows NT располагаются в вытесняемой памяти и взаимодействуют друг с другом путем передачи сообщений, как и положено в микроядерных операционных системах. В то же время все компоненты ядра работают в одном адресном пространстве и активно используют общие структуры данных, что свойственно операционным системам с монолитным ядром

Средства аппаратной поддержки ОС

1. Средства поддержки привилегированного режима: системные регистры процессора, слово состояния процессора, привилегированные команды, привилегированные режимы.

2. Средства трансляции адресов: буферы быстрой трансляции виртуальных адресов, регистры процессора, средства поддержки сегментно-страничных таблиц.

3. Средства переключения процессов: регистры общего назначения, системные регистры и указатели, флаги операций.

4. Система прерываний: регистры и флаги прерываний, регистры масок, контроллеры прерываний.

5. Системный таймер и системные часы.

6. Средства защиты памяти: граничные регистры, ключи.

1.5. Классификация операционных систем

1. **Назначение** (универсальные, специализированные – управление производством, обучение)
2. **Способ загрузки** (загружаемые, постоянно находящиеся в памяти)
3. **Особенности алгоритмов управления ресурсами**
   - 3.1. Многозадачность: однозадачные (MS DOS), невытесняющая многозадачность (Windows 3.x, NewWare), вытесняющая многозадачность (Windows NT, OS/2, Unix)
   - 3.2. Многопользовательский режим: отсутствие (MS DOS, Windows 3.x), имеется (Windows NT, OS/2, Unix)
   - 3.3. Многопроцессорная обработка: отсутствие, асимметричные ОС, симметричные ОС
4. **По базовой технологии** (Юникс-подобные или подобные Windows)
5. **По типу лицензии** (проприетарная или открытая)
6. **По состоянию развития** (устаревшая DOS, NextStep или современные GNU/Linux и Windows)

7. Область использования и форма эксплуатации

- пакетная обработка (OS/360)
- разделение времени
- реальное время (VxWorks,QNX)

8. Аппаратная платформа

8.1. ОС для смарт-карт (с интерпретатором виртуальной Java-машины)

8.2. Встроенные ОС (Palm OS, Windows CE –Consumer Electronics)

8.3. ОС для ПК (Windows 9.x, Windows 2000, Linux, Mac OS X)

8.4. ОС мини-ЭВМ (RT-11 и RSX-11M для PDP-11, UNIX для PDP-7)

8.5. ОС мэйнфреймов (OS/390 – пакетная обработка, разделение времени, обработка транзакций)

8.6. Серверные операционные системы для ЛВС, Интранет и Интернет (UNIX, AIX, Windows 2000/2002, Linux)

8.7. Кластерные операционные системы (Windows 2000 Cluster Server, Sun Cluster (Solaris))

1.6. Эффективность и требования, предъявляемые к операционным системам

1. Эффективность – степень соответствия своему назначению, техническое совершенство и экономическая целесообразность
2. Надежность и отказоустойчивость
3. Безопасность (защищенность)
4. Предсказуемость
5. Расширяемость
6. Переносимость
7. Совместимость
8. Удобство
9. Масштабируемость

1.7. Множественные прикладные среды. Совместимость

Совместимость – возможность операционной системы выполнять приложения , разработанные для других операционных систем.

Виды совместимости:

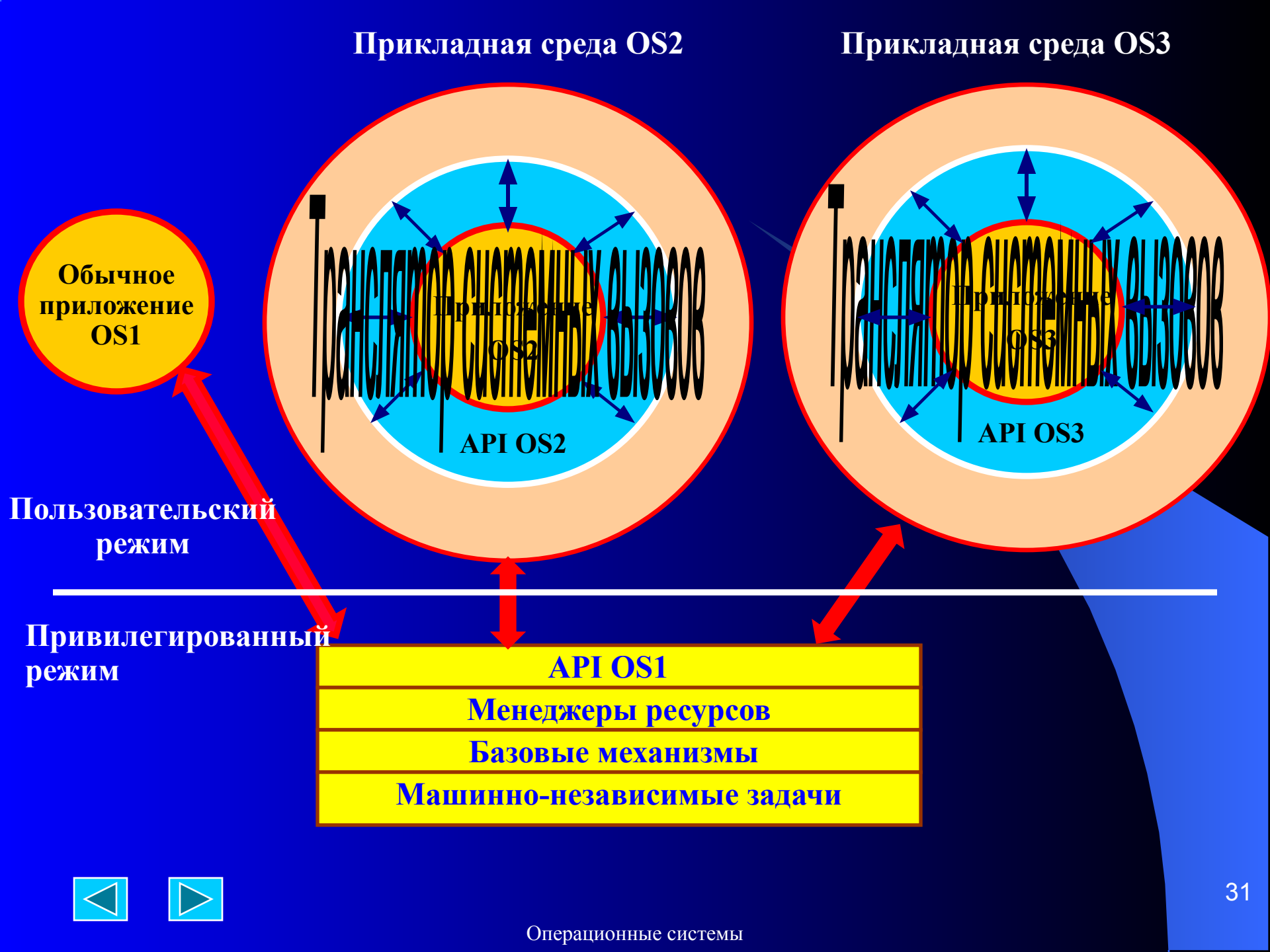
1. На двоичном уровне (уровень исполняемой программы).
2. На уровне исходных текстов (уровень исходного модуля).

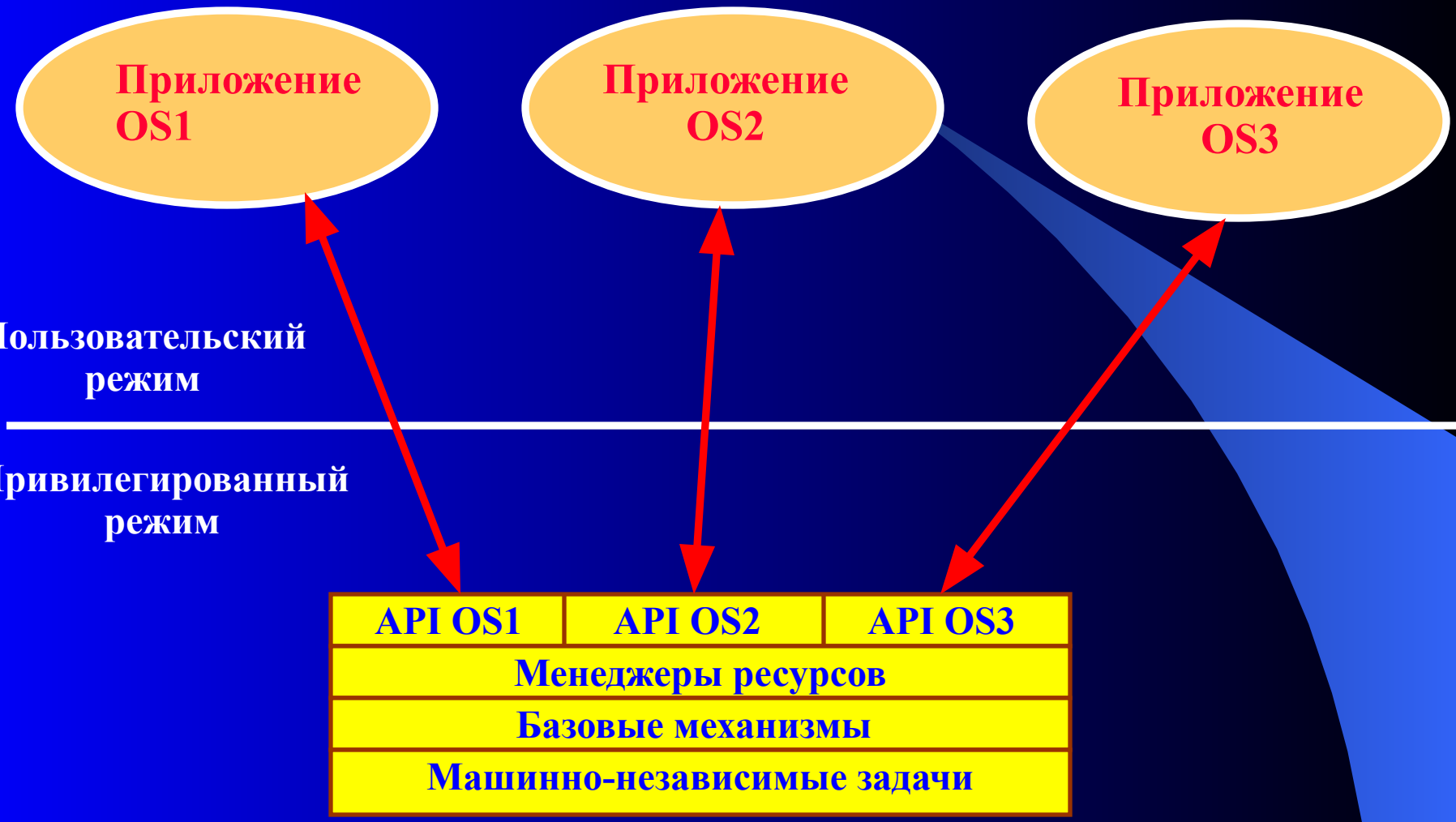
Вид совместимости определяется:

1. Архитектурой центрального процессора.
2. Интерфейсом прикладного программирования (API).
3. Внутренней структурой исполняемого файла.
4. Наличием соответствующих компиляторов и библиотек.

Способы достижения совместимости:

1. Эмуляция двоичного кода.
2. Трансляция библиотек.
3. Создание множественных прикладных сред различной архитектуры.


Прикладная среда OS2
Прикладная среда OS3
Обычное приложение OS1
Транслятор системных вызовов
Приложение OS2
API OS2
Транслятор системных вызовов
Приложение OS3
API OS3
Пользовательский режим
Привилегированный режим
API OS1
Менеджеры ресурсов
Базовые механизмы
Машинно-независимые задачи

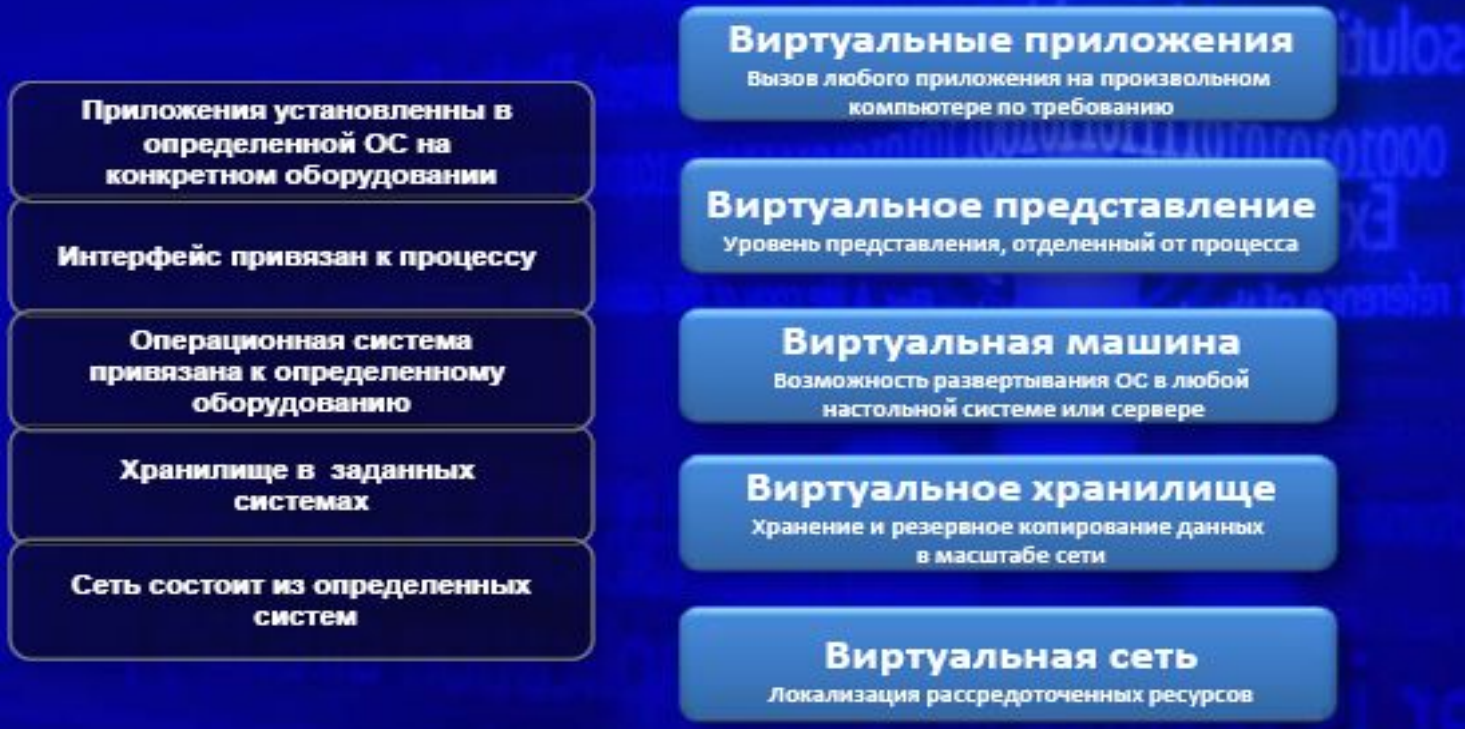

Приложение OS1
Приложение OS2
Приложение OS3
Пользовательский режим
Привилегированный режим
API OS1
API OS2
API OS3
Менеджеры ресурсов
Базовые механизмы
Машинно-независимые задачи


Приложения
Серверы ОС
Сетевой сервер
Сервер безопасности
Приложение OS2
Пользовательский режим
Приложение OS1
Приложение OS3
Прикладная программная среда OS3
Прикладная программная среда OS2
Прикладная программная среда OS1
Привилегированный режим
МИКРОЯДРО

Подсистемы среды Windows 2000

1.8. Виртуализация от Microsoft

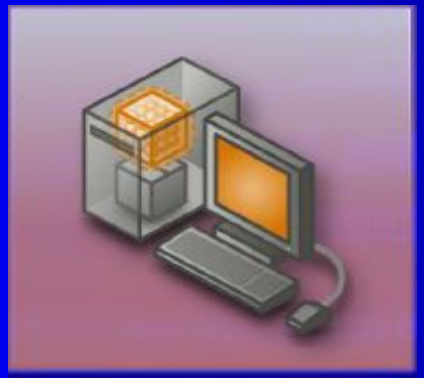
Виртуализация – это отделение одного вычислительного ресурса от других:



Виртуализация ***повышает эффективность использования ресурсов***, обеспечивает ***повышенную гибкость*** и упрощает ***управление изменениями***

Виртуализация приложений

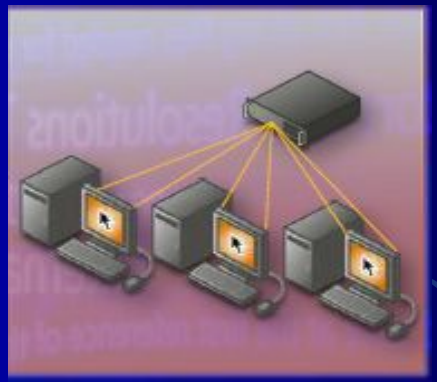
- Абстрагирование приложения от базовой операционной системы
- Исполнение приложений в виртуальной среде
- Сокращение риска нежелательного взаимодействия между приложениями и операционной системой
- Устранение конфликтов между приложениями
- Существенные преимущества на всех этапах жизненного цикла приложений
  - Управление
  - Внедрение
  - Запуск

Виртуализация настольных систем

Создание в стандартной настольной системе дополнительной, изолированной среды ОС

- Поддержка устаревших приложений в современных ОС
- Минимизация конфликтов между приложениями и ОС
- Ускорение миграции на новые ОС



Виртуализация представления

Централизованная обработка и хранение данных; локальное представление пользовательского интерфейса

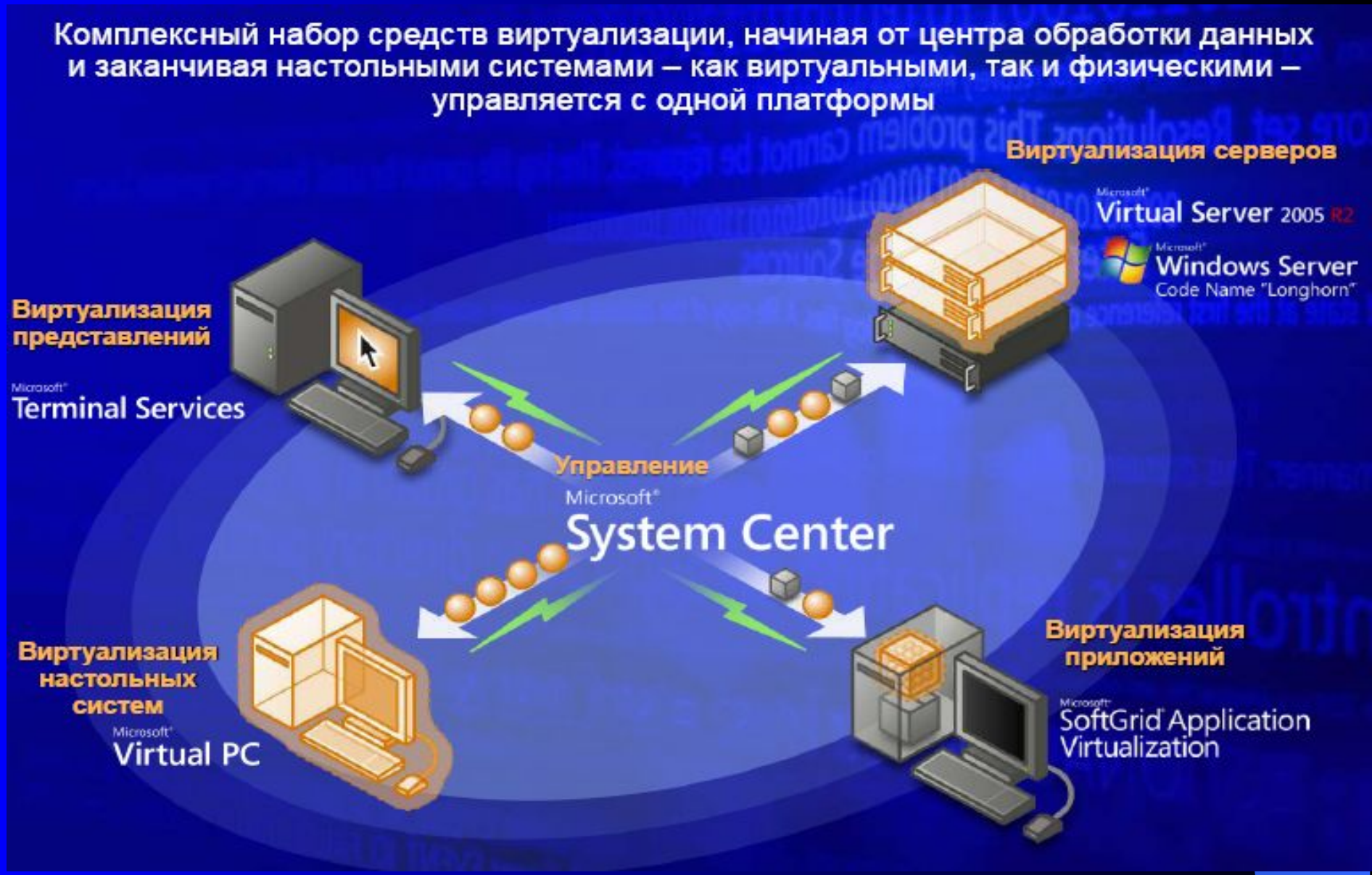
- Минимизация конфликтов между приложениями и ОС
- Упрощение процедур соответствия нормативным требованиям и обеспечения конфиденциальности данных
- Сокращение затрат на администрирование настольных систем

Виртуализация серверов

Консолидация нагрузки для более эффективного использования ресурсов

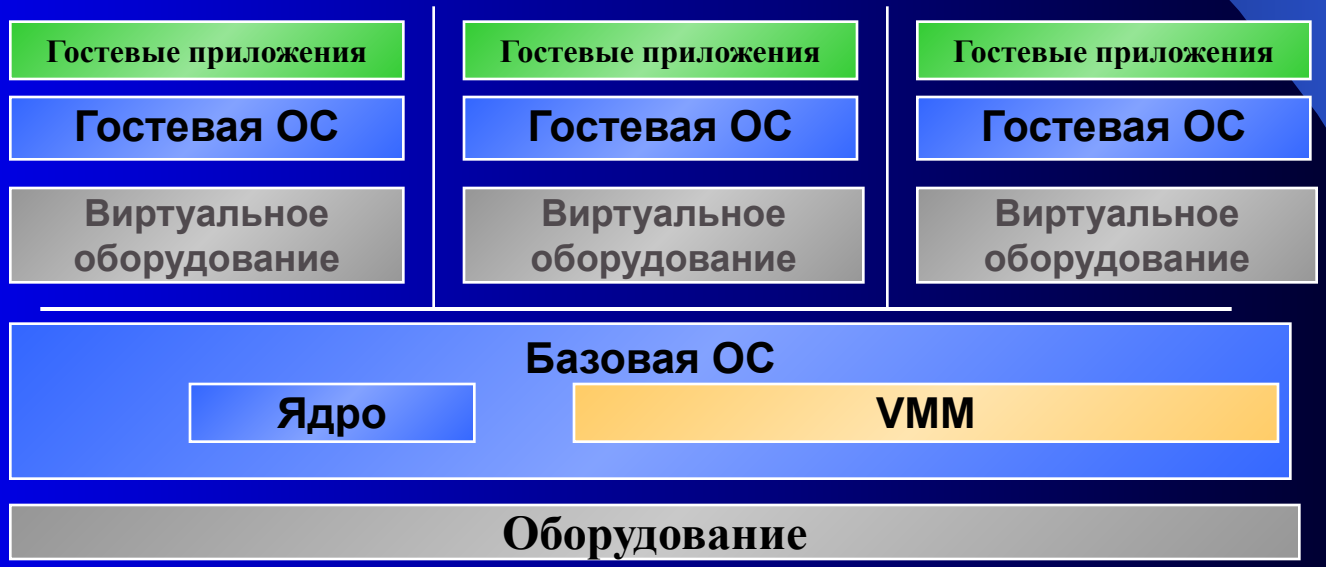
- Сокращение эксплуатационных затрат (на оборудование, энергоресурсы, площади)
- Увеличение времени работоспособного состояния и доступности
- Надежное аварийное восстановление
- Сокращение простоя из-за обслуживания
- Упрощение контроля потребления и масштабирования ресурсов

Виртуализация от Microsoft



Архитектура. Virtual Machine Monitor (VMM)

- ЦП вынужден переключаться между процессами базовой ОС и гостевой ОС
  - VMM переключает контекст между этими процессами
  - Компьютер работает в контексте хоста либо VMM
- На одном ЦП может работать только одна ОС
- Сжатие кода нулевого кольца (ring 0) гостевой ОС



Виртуализация ЦП. Проблемы

При прямом доступе гостевая ОС будет работать быстро! (99%)

Когда требуется выполнить привилегированную операцию, срабатывает ловушка, и VMM обрабатывает эту операцию в режиме ядра.

Проблема: полная виртуализация платформы x86 таким способом невозможна, так как некоторые инструкции ЦП для режима ядра, выполняющие чтение, разрешены не только в нулевом кольце

Возможные решения:

a) Перекомпилировать ОС и приложения, избегая этих 20 инструкций, т.е. исключить 20 «проблемных» инструкций.

b) Воспользоваться исполнением с трансляцией двоичного кода (модификация кода «на лету» во время выполнения на хосте).

c) Установить в гостевой системе VM Additions, что позволит модифицировать код в памяти VM.

d) Использовать аппаратную поддержку виртуализации (перехват инструкций в особом “кольце -1”).

Решения

1. Преобразование двоичного кода

Трансляция инструкций гостевой операционной системы в инструкции базовой ОС. Всегда возможна, но работает очень медленно.

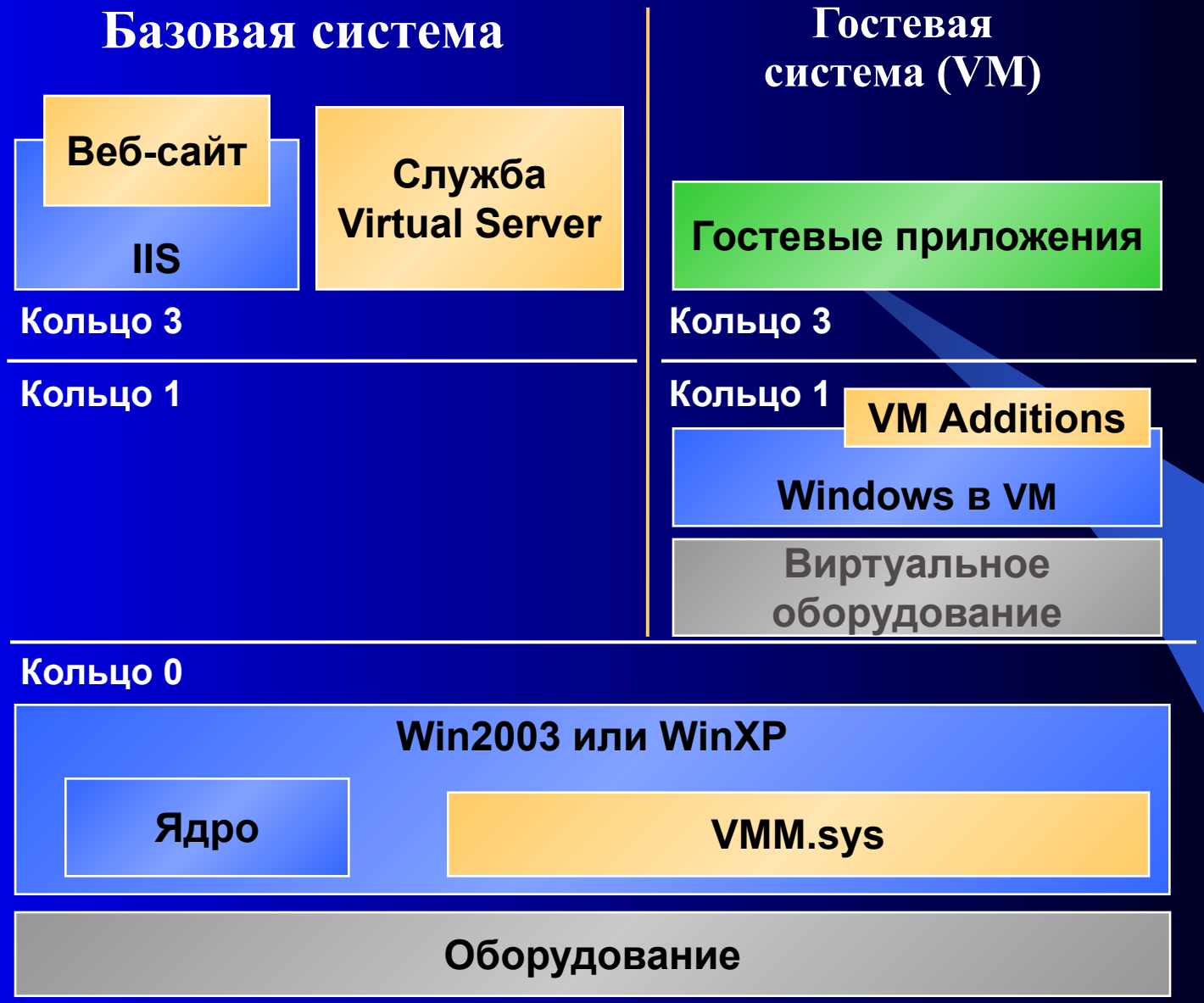
2. VM Additions

Модифицирует dll-код в памяти VM (невозможно в 64-разрядных версиях Vista и Longhorn).

VM Additions поддерживают синхронизацию времени, «пульс», завершение работы, оптимизированный SCSI-диск, лучшие драйверы мыши и видео.

3. Аппаратная виртуализация

ЦП с поддержкой технологий Intel VT или AMD Virtualization. ЦП решает проблемы, отслеживая параметры каждой VM (фактически, это «кольцо -1»).


Базовая система
Гостевая система (VM)
Веб-сайт
IIS
Служба Virtual Server
Гостевые приложения
Кольцо 3
Кольцо 3
Кольцо 1
Кольцо 1
VM Additions
Windows в VM
Виртуальное оборудование
Кольцо 0
Win2003 или WinXP
Ядро
VMM.sys
Оборудование

Версии VM Additions

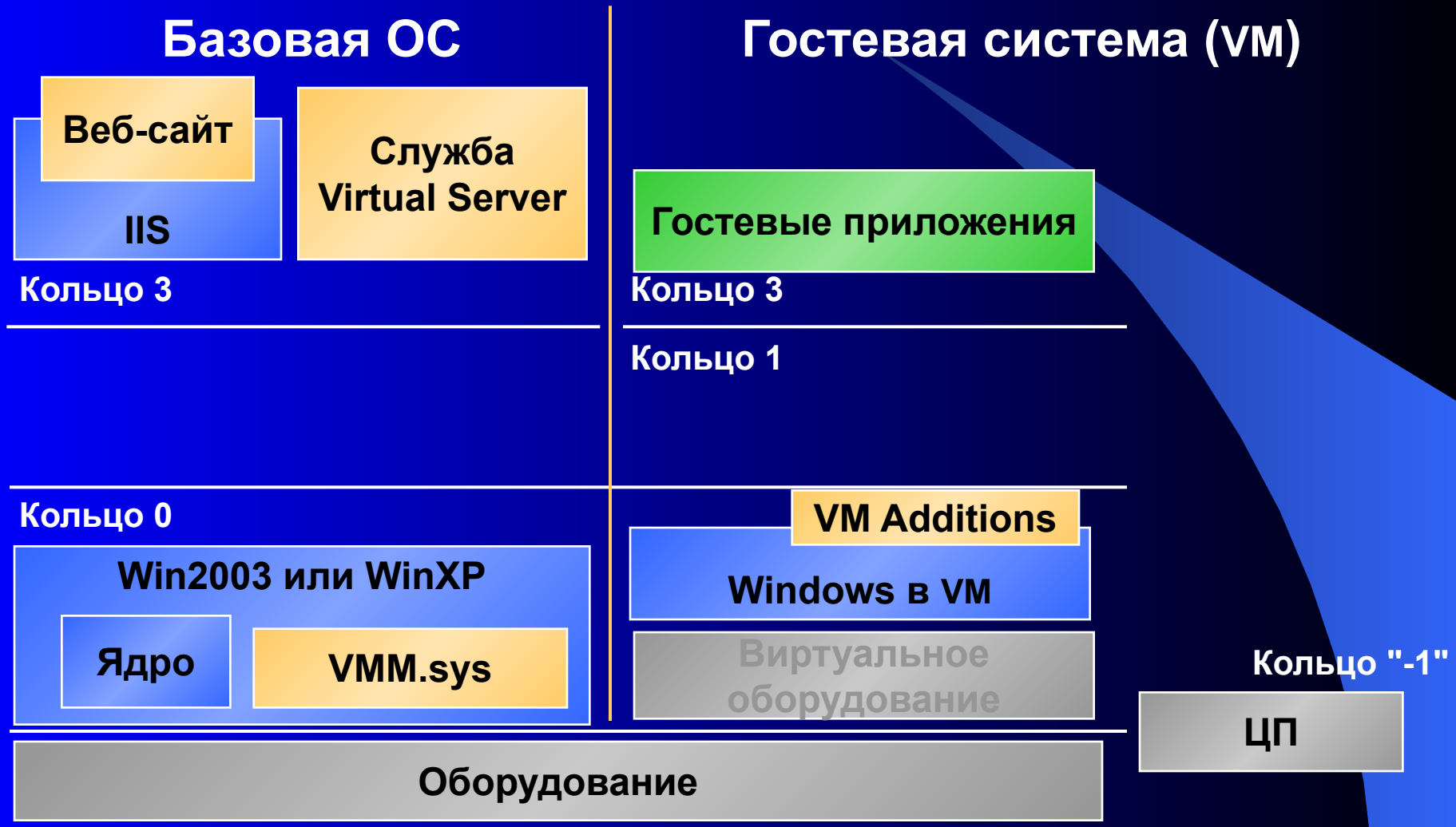
Сборка	Выпуск	Примечание
10.21	В составе Virtual PC 5.2	(дано название – Virtual PC Additions)
13.40	В составе Virtual PC 2004	
13.187	(отдельная загрузка)	Поддерживает Win XP SP2
13.206	В составе VS2005	
13.306	В составе Virtual PC 2004 SP1	
13.518	В составе VS2005 SP1 beta	
13.531	(отдельная загрузка)	Поддерживает Win2003 SP1
13.552	В составе VS2005 R2	Поддерживает Win2003 R2 и Vista (-build 5270)
13.705	В составе VS2005 R2 SP1 beta1	
13.706	(отдельная загрузка)	Поддерживает Vista B2 (-build 5384) и Longhorn
13.709	(отдельная загрузка)	Поддерживает Vista RC1
13.715	В составе VS2005 R2 SP1 beta2	Поддерживает Vista RTM
13.724	В составе Virtual PC 2007 beta	
13.803	В составе Virtual PC 2007	Загрузка – по адресу www.microsoft.com/virtualpc

Linux VM Additions

- Добавляется поддержка:
  - Синхронизации времени
  - «Пульса»
  - Завершения работы
  - SCSI-дисков
  - Драйвер мыши и видео
  - Поддержки прямого исполнения кода нет!
- Дистрибутивы (9x):
  - Red Hat 7.3/9.0, Enterprise 2.1/3/4
  - SuSE Linux 9.2/9.3/10.0, Enterprise Server 9

В выпуске VS 2005 R2 SP1 поддерживаются гостевые ОС : Red Hat Enterprise Linux 2.1 (update 7), Red Hat Enterprise Linux 3.0 (update 8), Red Hat Enterprise Linux 4.0 (update 4), Red Hat Enterprise Linux 5.0, SuSE Linux Enterprise Server 9.0, SuSE Linux Enterprise Server 10.0, Red Hat Linux 9.0, SuSE Linux 9.3, SuSE Linux 10.0, SuSE Linux 10.1, SuSE Linux 10.2.

Архитектура виртуализации с аппаратной поддержкой

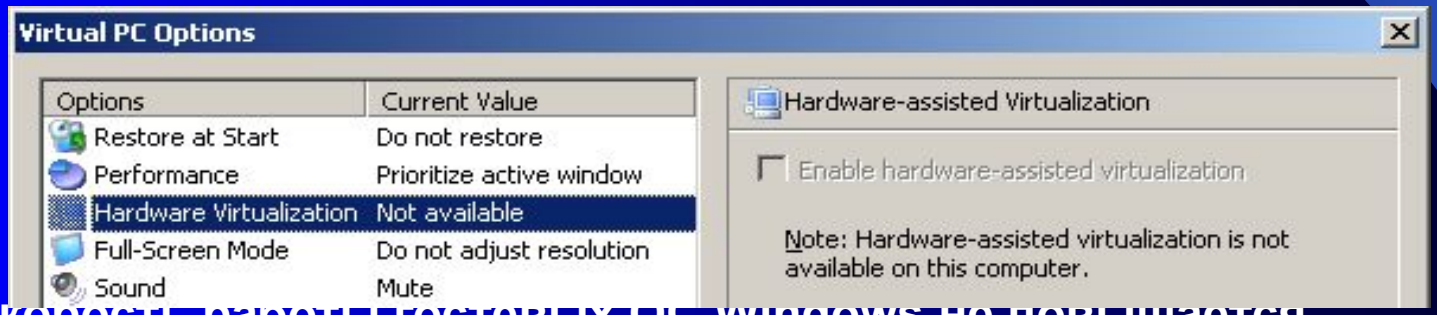


Виртуализация с аппаратной поддержкой (Intel VT или AMD Virtualization)

Поддерживается в:

- Virtual PC 2007
- Virtual Server 2005 R2 SP1
- Windows Virtualization (обязательно)

Необходимо включить в BIOS и в параметрах Virtual PC 2007



Скорость работы гостевых ОС Windows не повышается

- Последние версии VM Additions уже поддерживают прямой доступ к ЦП
- Установка Windows выполняется в 2-3 раза быстрее
- Гостевые ОС типа Linux и Netware работают быстрее

Спецификации Virtual Server 2005 R2

Базовая система:

VS2005 Standard Edition: до 4 ЦП (1- или 2-ядерные),

VS2005 Enterprise Edition: до 32 ЦП (1- или 2-ядерные),

ОЗУ: до 64 Гб

Гостевая система:

ЦП: до 1, ОЗУ: до 3,6 Гб, Сетевые адаптеры: до 4, (неограниченная пропускная способность). USB: нет, поддерживаются USB-клавиатура и USB-мышь, можно также подключить USB-устройство для чтения смарт-карт.

Дополнительные возможности Server 2005 R2 SP1:

Поддержка Intel VT и AMD Virtualization,
Поддержка 64-х разрядных базовых систем: Win2003 и WinXP.
Поддержка теневого копирования томов (Volume Shadow Copy, VSS),
Интеграция с Active Directory средствами Service Connection Points,
Поддержка Vista как гостевой ОС,
Утилита для монтирования VHD,
Емкость по умолчанию VHD - 127 Гб (ранее – 16 Гб),
Исправление Virtual SCSI для гостевых ОС Linux 2.6.x,
Кластеризация VM,
Передача VM при ее сбое в пределах того же хоста,
Общий SCSI- (iSCSI-) диск для гостевых систем.

Virtual PC / Virtual Server 2005 R2

Windows Virtualization

Поддержка виртуализации для Windows Server

Windows Hypervisor (Гипервизор), кодовое имя - "Viridian“:

«Тонкий» (~160 Кб) программный уровень, «внутренняя базовая ОС»,
Родительский раздел – управляет дочерними разделами,
Дочерний раздел включает любое число ОС, управляемых родительским разделом.

Стек виртуализации:

Работает в корневом (= родительском) разделе,
Обеспечивает виртуализацию устройств,
WMI-интерфейс для управления

Провайдеры служб виртуализации (Virtualization Service Providers, VSPs)

Архитектура совместного использования оборудования, **В**
гостевой ОС устанавливаются драйверы "viridian«.

Windows Virtualization Server требует x64-совместимого оборудования, ЦП с поддержкой Intel VT или AMD-V

Поддерживает: 32- и 64-разрядные гостевые ОС; до 8 ЦП на VM; горячее добавление» ЦП, ОЗУ, сетевых адаптеров, дисков; > 32 Гб ОЗУ на VM; возможность переноса VM без отключения; традиционную модель драйверов; использование существующих драйверов Windows; прежний же набор эмулируемого оборудования; Server Core в качестве родительской ОС

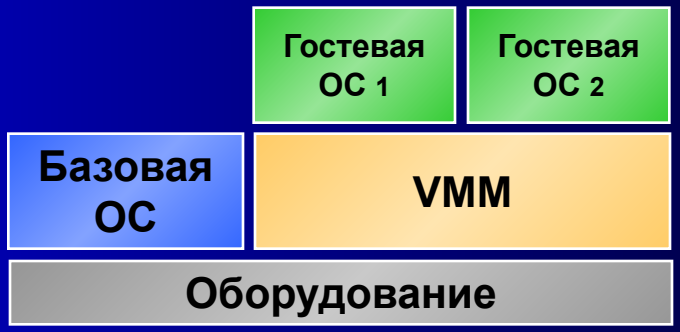
Windows Virtualization

Схемы VMM

VMM типа 2

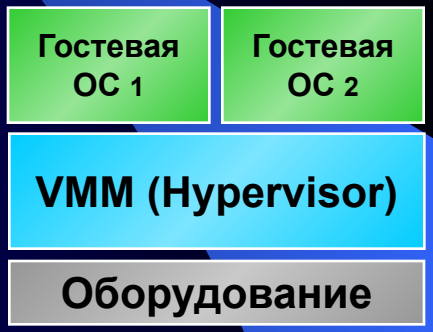
Примеры:
- JVM
- .NET CLR

Гибридный VMM



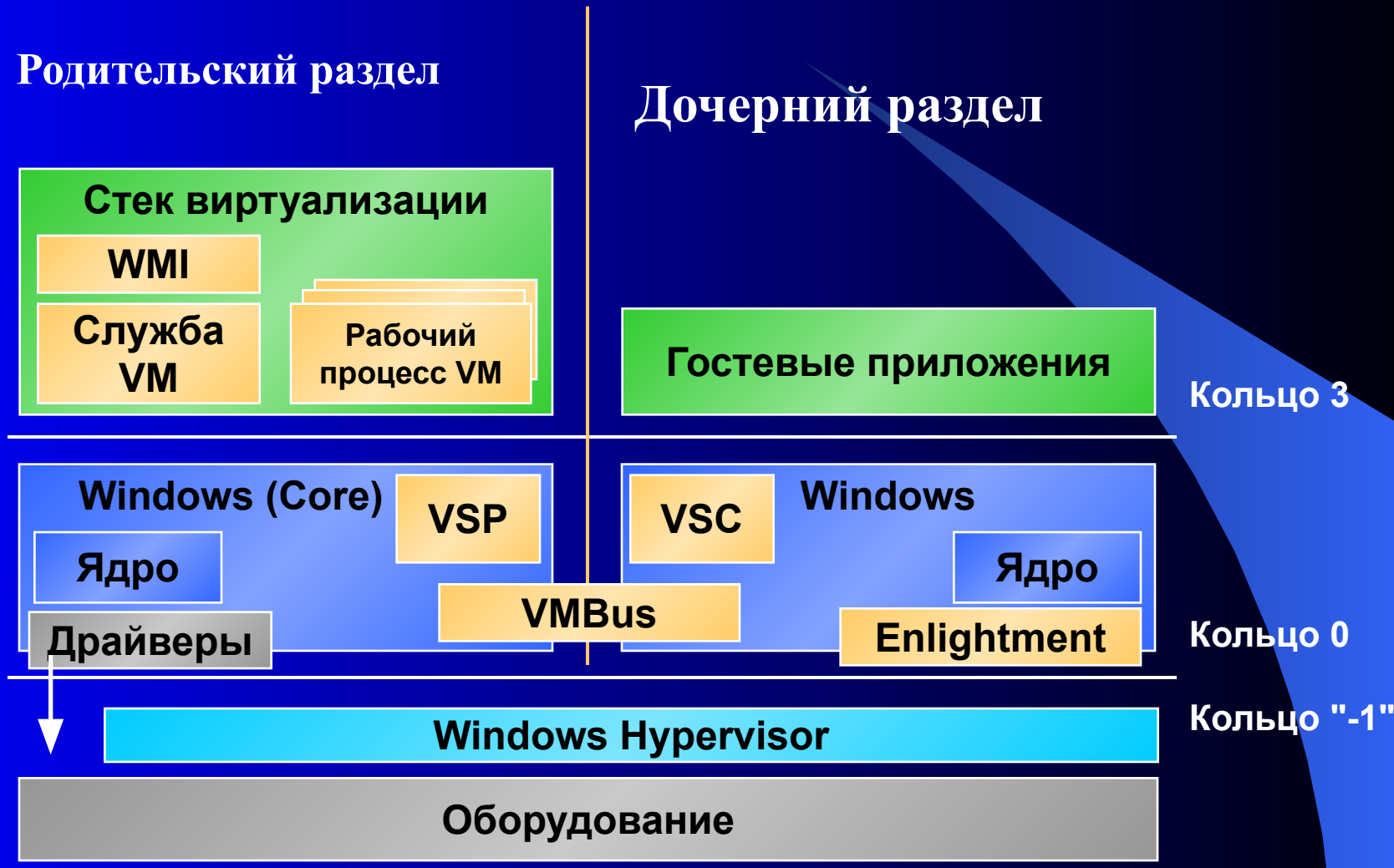
Примеры:
- Virtual PC
- Virtual Server

VMM типа 1 Hypervisor



Примеры:
- Виртуализация Windows ("Viridian")

Windows Virtualization



Версии продуктов

Продукт	Выпуск	Базовые системы	Гостевые системы **
Virtual PC 2004	**Октябрь 2003**	• **Win2000 Pro SP4** • **Win XP Pro (Tablet, SP1)**	• MS-DOS 6.22 * / OS/2 • Win 95, 98, 98SE, ME * • Win NT4 SP6a (workstation) * • Win2000 Pro SP4 • Win XP (Tablet, SP1)
Virtual Server 2005	**Июль 2004**	• Win XP Pro • Win2003 SBS • Win2003 (SE, EE, Data)	• Win NT4 SP6a (server) * • Win2000 Server • Win2003 (SE, EE, Web)
Virtual PC 2004 SP1	**Октябрь 2004**	То же, что и для Virtual PC 2004 + Win2003 SE	То же, что и для Virtual PC 2004 + Win XP SP2
Virtual Server 2005 R2	**Ноябрь 2005**	То же, что и для Virtual Server 2005 + Win XP Pro SP2 (non prod) + Win2003 (SP1, R2) + Win XP / Win2003 x64	То же, что и для Virtual Server 2005 + Win XP Pro SP2 + Win2003 (SP1, R2) + Linux (9x) - Apr 2006
Virtual PC 2004 Express	**Март 2006**	То же, что и для Virtual PC 2004 SP1 + Поддерживает не более одной VM + в Vista Enterprise / только для участников программы Software Assurance	
Virtual PC 2007	**19 февраля 2007**	+ Поддержка ЦП с технологиями Intel VT и AMD Virtualization + Поддержка Vista (гостевые и хост-системы)	
Virtual Server 2005 R2 SP1	Март 2007	+Поддержка виртуализации процессоров Intel VT и AMD Virtualization +Поддержка Volume Shadow Copy Service (для резервного копирования)	
Windows Virtualization	Longhorn + < 180 дней	Реализация Windows Hypervisor Новая модель виртуализации, требует аппаратной поддержки VT/Virtualization Кодовое имя "Viridian"	

* Жизненный цикл этих продуктов близок к завершению

** Server На http://vpc.visualwin.com находится список из > 1200 (!) ОС, совместимых с Virtual PC и Virtual

В статье KB 867572 см. список ОС, поддерживаемых Virtual Server 2005 R2

Основные области применения:

- **Тестирование программного обеспечения и средств разработки (тестирование создаваемых приложений, тестирование конфигураций и настроек готового программного обеспечения, а также действий администраторов серверов и сети с целью проверки работоспособности той или иной конфигурации серверного ПО перед началом ввода его в реальную эксплуатацию.**
- **Хостинг унаследованных приложений. Зачастую наиболее удачные бизнес-приложения эксплуатируются десятилетиями, поэтому вполне может случиться так, что платформа, для которой они написаны, в компании уже практически не применяется из-за отсутствия нормальной технической поддержки со стороны производителей оборудования.**
- **Консолидация загрузки серверов. Идея консолидации загрузки серверов заключается в создании виртуальных машин с разными операционными системами и программным обеспечением, реализующими выполнение указанных задач, и в размещении одного и того же набора этих виртуальных машин на нескольких физических серверах. Благодаря этому число самих серверов можно уменьшить, да и выход из строя одного из серверов не будет столь критичен для компании, поскольку его нагрузку может взять на себя виртуальная машина на каком-либо другом сервере.**
- **Моделирование распределенных серверных приложений на одном физическом сервере. Данный способ применения серверных виртуальных машин предназначен для разработчиков, специалистов по тестированию и специалистов по внедрению приложений масштаба предприятия. С его помощью можно создавать распределенные приложения, тестировать их, а также моделировать реальные условия внедрения, используя для этой цели один-единственный компьютер, что позволяет сократить расходы на приобретение аппаратного обеспечения для разработки приложений.**

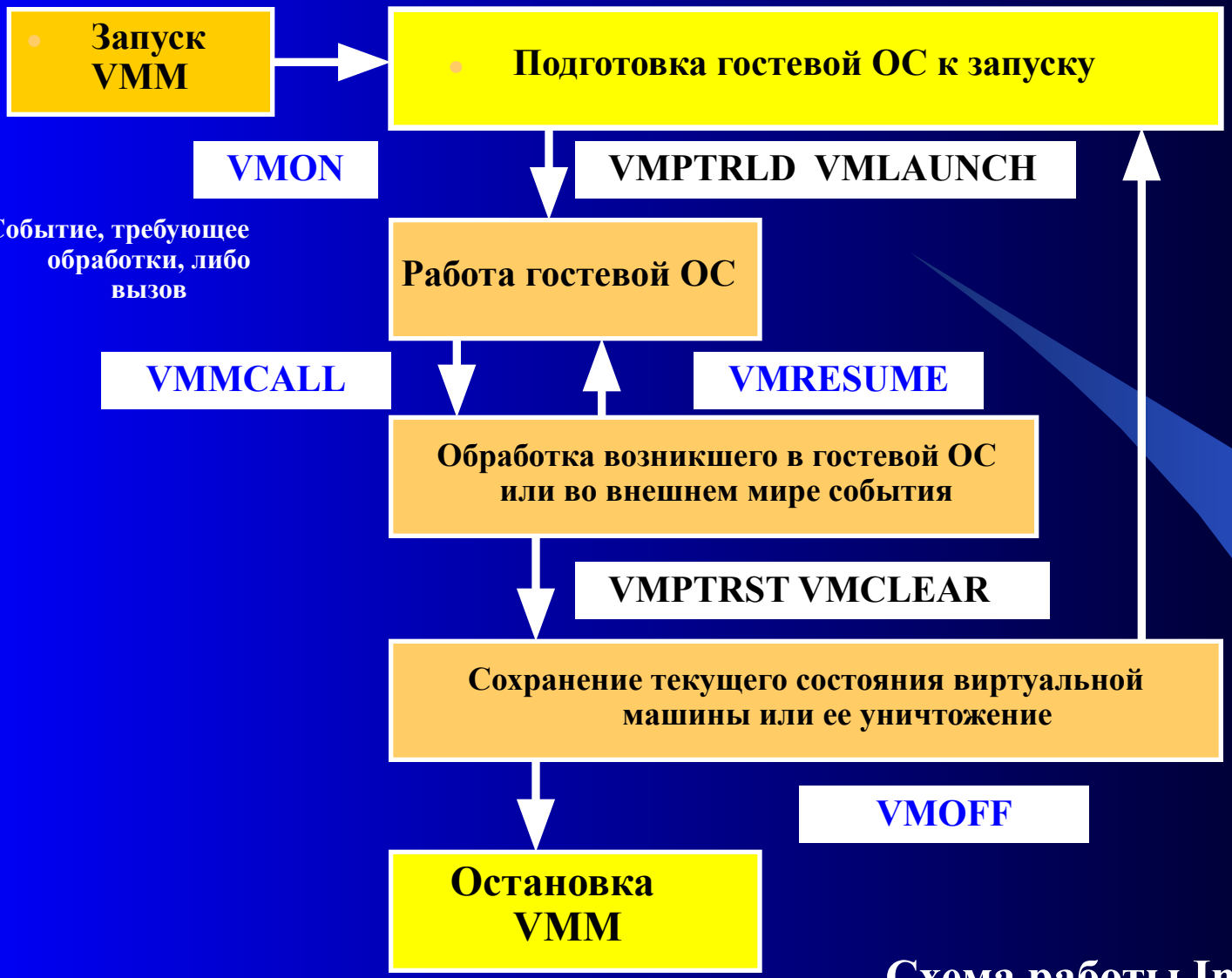
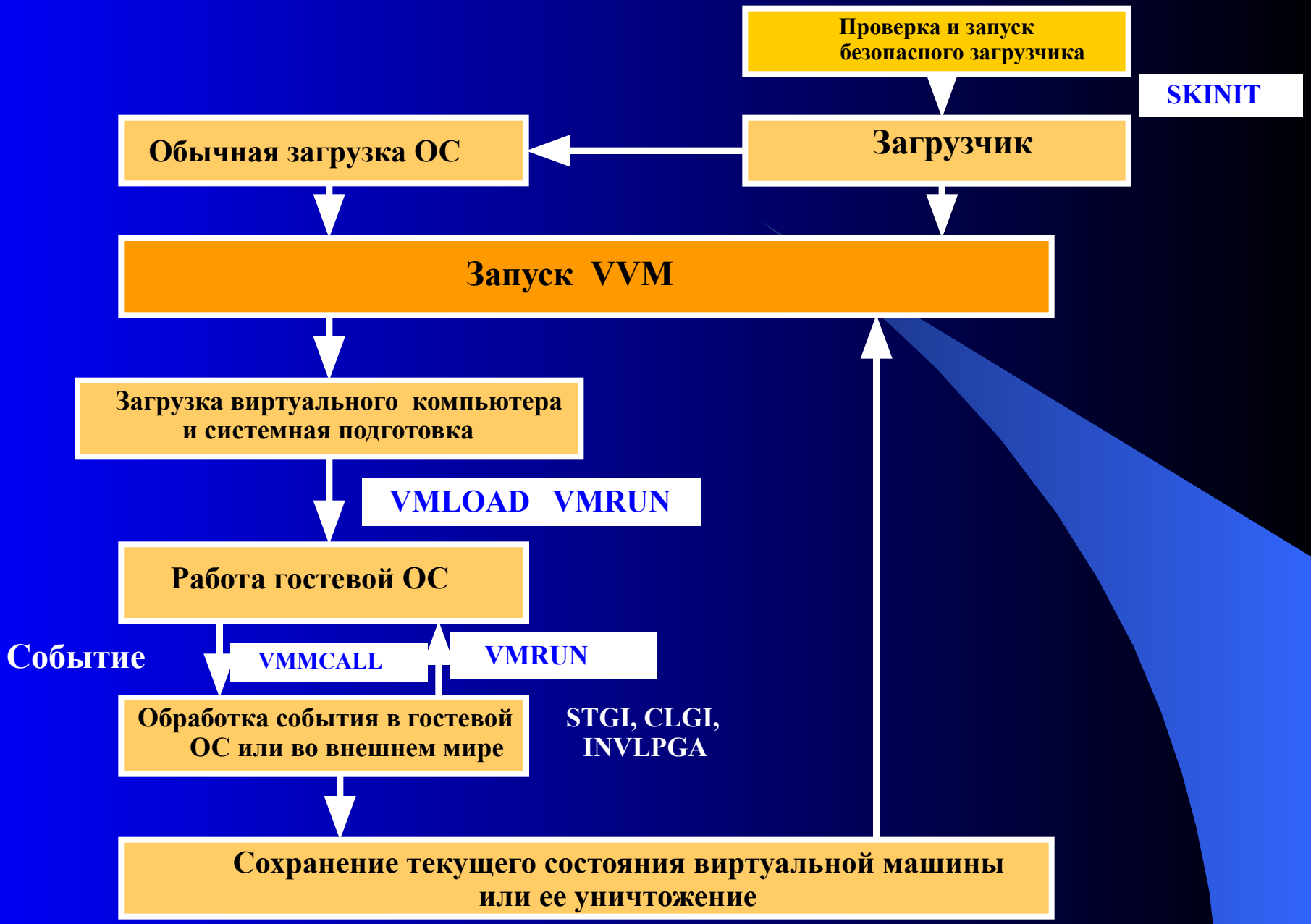


Схема работы Intel Virtualization Technology


Проверка и запуск безопасного загрузчика
SKINIT
Загрузчик
Обычная загрузка ОС
Запуск VVM
Загрузка виртуального компьютера и системная подготовка
VMLOAD VMRUN
Работа гостевой ОС
Событие
VMMCALL
VMRUN
Обработка события в гостевой ОС или во внешнем мире
STGI, CLGI, INVLPGA
Сохранение текущего состояния виртуальной машины или ее уничтожение