Лекция 2. Требования к современным операционным системам (ОС) . Функциональные компоненты ОС автономного компьютера

Требования к современным ОС

- функциональная полнота
- расширяемость (поддержка новых типов внешних устройств или новых технологий без переписывания кода системы»)
- переносимость («легкий перенос с одной аппаратной платформы на другую»)
- совместимость (перенос приложений, разработанных для одной ОС в среду другой ОС)
- надежность (действия ОС предсказуемы, а приложения не должны наносить вред ОС) и отказоустойчивость (поддержка аппаратных средств обеспечения отказоустойчивости, таких как дисковые массивы или источники бесперебойного питания)
- безопасность (защита данных и других ресурсов от несанкционированного доступа)
- производительность (должна обладать настолько хорошим быстродействием и временем реакции, насколько позволяют аппаратные средства)

Классификация ОС

Поддержка многозадачности

- однозадачные
- многозадачные
  - с вытесняющей многозадачностью
  - с не вытесняющей многозадачностью

Поддержка многопоточности

- многопоточные
- не поддерживают понятия потока

Классификация ОС

Поддержка многопользовательского режима:

- однопользовательские (MS-DOS, ранние версии OS/2)
- многопользовательские (UNIX, Windows NT/2000/XP)

Многопроцессорная обработка:

- поддержка мультипроцессирования
- не поддерживает мультипроцессорную обработку

Поддержка сети:

- сетевые ОС
- не сетевые ОС

Классификация ОС

Особенности аппаратных платформ

- ОС для персональных компьютеров
- ОС для мини-компьютеров
- ОС для мейнфреймов
- ОС для кластеров и сетей ЭВМ

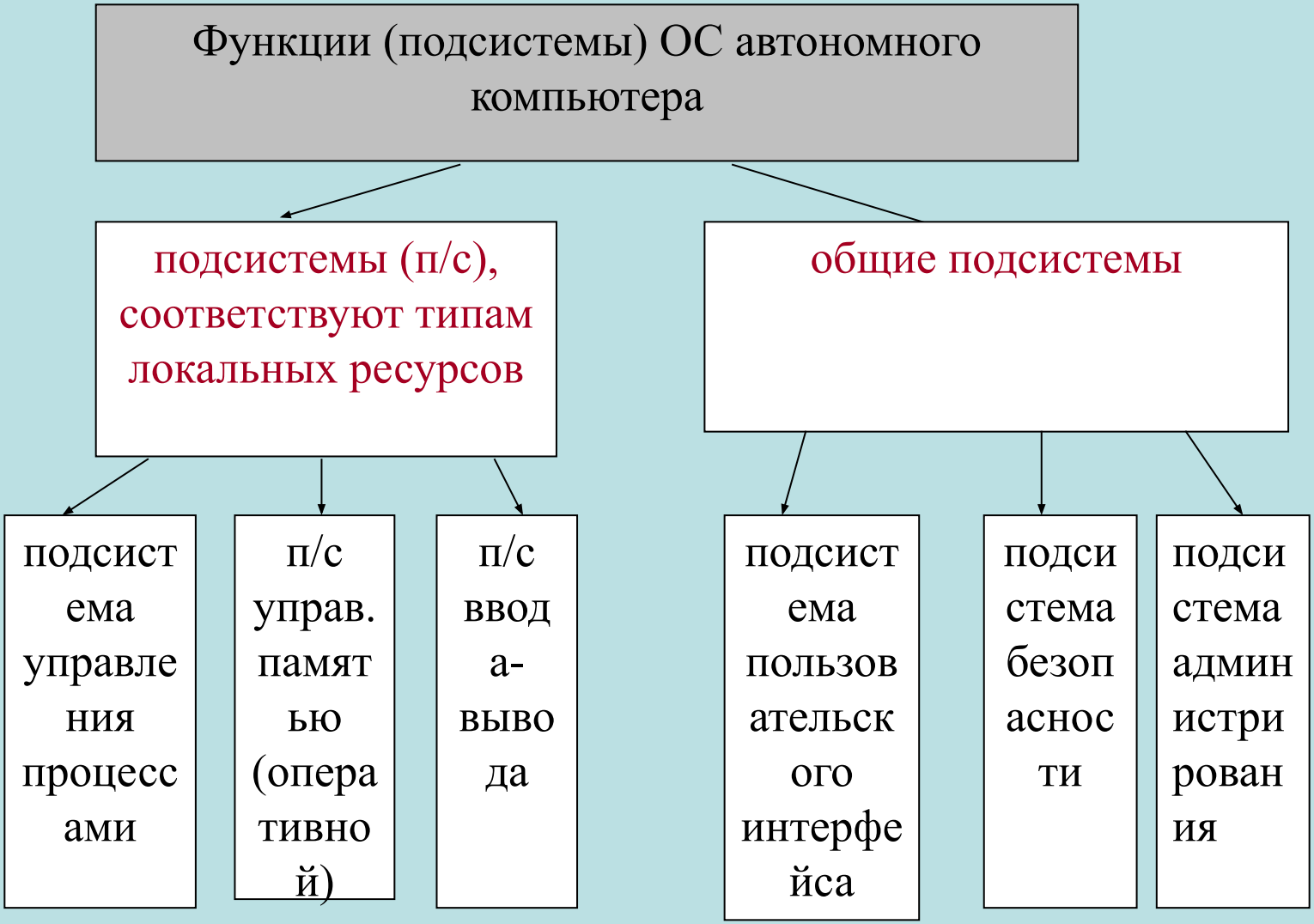
Особенности областей использования (критерий эффективности)

- системы пакетной обработки (ОС ЕС),
- системы разделения времени (UNIX, VMS, Windows NT/2000/XP)
- системы реального времени (QNX, RT/11)

Функциональные компоненты операционной системы автономного компьютера

Функции операционной системы автономного компьютера обычно группируются либо в соответствии с типами локальных ресурсов, которыми управляет ОС, либо в соответствии со специфическими задачами, применимыми ко всем ресурсам. Иногда такие группы функций называют **подсистемами**

Классификация подсистем ОС автономного ПК



Подсистема управления процессами

Основные понятия:

- Под **процессом** в общем случае понимается программа в стадии выполнения
- **Процесс** можно также определить как некоторую заявку на потребление системных ресурсов.
- В **мультипрограммной** операционной системе одновременно может существовать несколько процессов
- Часть процессов порождается по инициативе пользователей и их приложений, такие процессы обычно называют **пользовательскими**
- Процессы, называемые **системными**, инициализируются самой операционной системой для выполнения своих функций.
- Совокупность всех областей оперативной памяти, выделенных операционной системой процессу, называется его **адресным пространством**.

Основные функции подсистемы управления процессами

- создание и уничтожение процессов (т.е. структур данных, связанных с процессами)
- поддержание очередей заявок процессов на ресурсы
- защита ресурсов, выделенных данному процессу, от остальных процессов организовывать совместное использование ресурсов
- обеспечивать прерывание и возобновление некоторого процесса
- функции синхронизации процессов, позволяющие процессу приостанавливать свое выполнение до наступления какого-либо события в системе
- предоставить средства межпроцессного взаимодействия

Подсистема управления памятью (основные функции)

- распределение имеющейся физической памяти между всеми существующими в системе в данный момент процессами (выделение и освобождение памяти)
- загрузка кодов и данных процессов в отведенные им области памяти
- настройка адресно-зависимых частей кодов процесса на физические адреса выделенной области
- защита областей памяти каждого процесса (избирательная способность предохранять выполняемую задачу от записи или чтения памяти, выделенной другой задаче)

Подсистема управления файлами и внешними устройствами

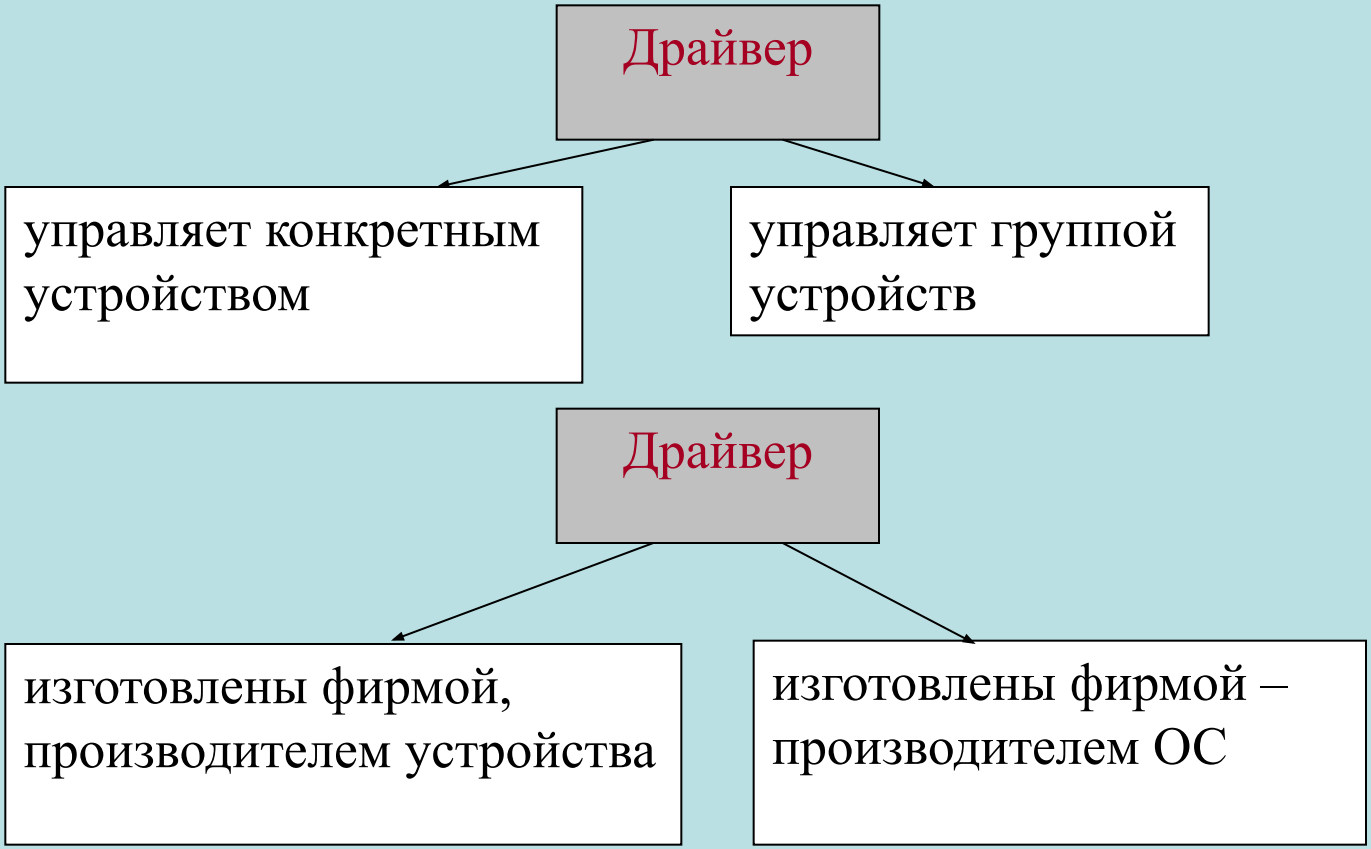
Основные определения:

- **Файл** - простая неструктурированная последовательность байтов, имеющей символьное имя.
- **Драйвер** - программа, управляющая конкретной моделью внешнего устройства и учитывающая все его особенности

Файловая система ОС выполняет:

- преобразование символьных имен файлов, с которыми работает пользователь или прикладной программист, в физические адреса данных на диске,
- организует совместный доступ к файлам,
- защищает их от несанкционированного доступа.

Классификация драйверов



Подсистемы защиты данных

Безопасность данных вычислительной системы обеспечивается:

- средствами отказоустойчивости ОС, направленными на защиту от сбоев и отказов аппаратуры и ошибок программного обеспечения
- средствами защиты от несанкционированного доступа (ОС защищает данные от ошибочного или злонамеренного поведения пользователей системы).

Отказоустойчивость

Поддержка отказоустойчивости реализуется операционной системой, как правило, на основе :

- резервирования
- использования отказоустойчивых дисковых систем
- использования восстанавливаемых ФС

Подсистема пользовательского интерфейса

Возможности операционной системы доступны прикладному программисту в виде набора функций, называющегося **интерфейсом прикладного программирования (Application Programming Interface, API).**

API- функции используются:

- когда для выполнения тех или иных действий им требуется особый статус, которым обладает только операционная система
- помимо этих функций прикладной программист может воспользоваться набором сервисных функций ОС, которые упрощают написание приложений.

Интерфейс прикладного программирования

- для Windows-систем – Win32
- для UNIX-систем - POSIX

Архитектура ОС. Многослойная и микроядерная архитектуры

Функциональная сложность операционной системы => сложность ее архитектуры

Обычный состав ОС

- исполняемые и объектные модули стандартных для данной ОС форматов
- библиотеки разных типов
- модули исходного текста программ
- программные модули специального формата (например, загрузчик ОС, драйверы ввода-вывода)
- конфигурационные файлы
- файлы документации
- модули справочной системы
- др. файлы

Обычный принцип построения ОС

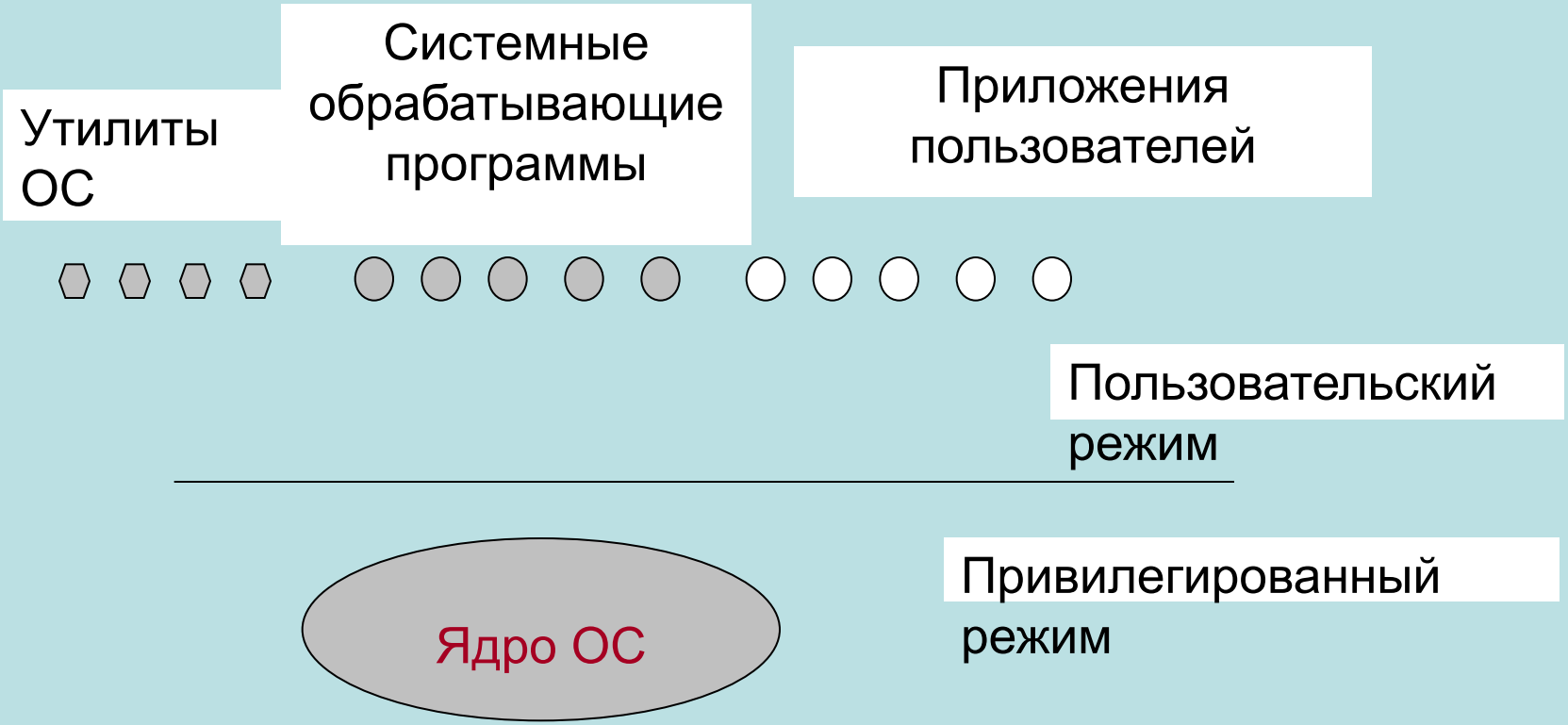
Разделение всех ее модулей на две группы:

- ядро – модули, выполняющие основные функции ОС;
- модули, выполняющие вспомогательные функции ОС.

Функции ядра

- решающие внутрисистемные задачи организации вычислительного процесса (такие как переключение контекстов, загрузка/выгрузка страниц, обработка прерываний), эти функции недоступны для приложений
- другой класс функций ядра служит для поддержки приложений, функции ядра, которые могут вызываться приложениями, образуют интерфейс прикладного программирования – API

Архитектура операционной системы с ядром в привилегированном режиме



Вспомогательные модули ОС

Вспомогательные модули ОС обычно подразделяются на следующие группы:

- *утилиты* – программы, решающие отдельные задачи управления и сопровождения компьютерной системы, такие, например, как программы работы с дисками (архиваторы, дефрагментаторы, программы разметки диска ит.д.), архивирования данных на магнитную ленту;
- *системные обрабатывающие программы* – текстовые или графические редакторы, компиляторы, компоновщики, отладчики;
- *программы предоставления пользователю дополнительных услуг* – специальный вариант пользовательского интерфейса, калькулятор и даже игры;
- *библиотеки процедур* различного назначения, упрощающие разработку приложений, например, библиотека математических функций, функций ввода-вывода и т. д.

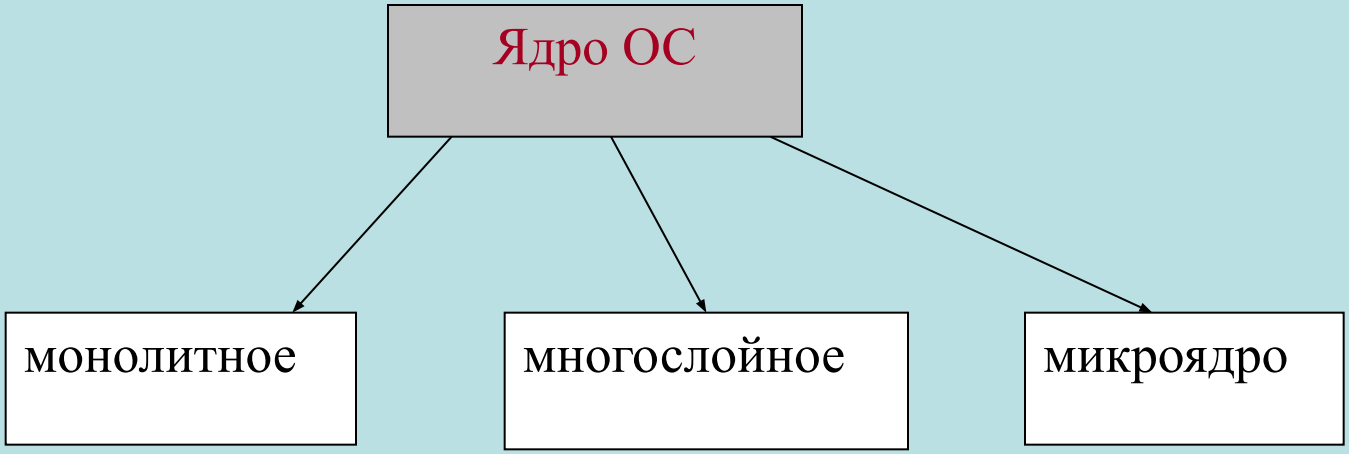
Основные свойства ядра

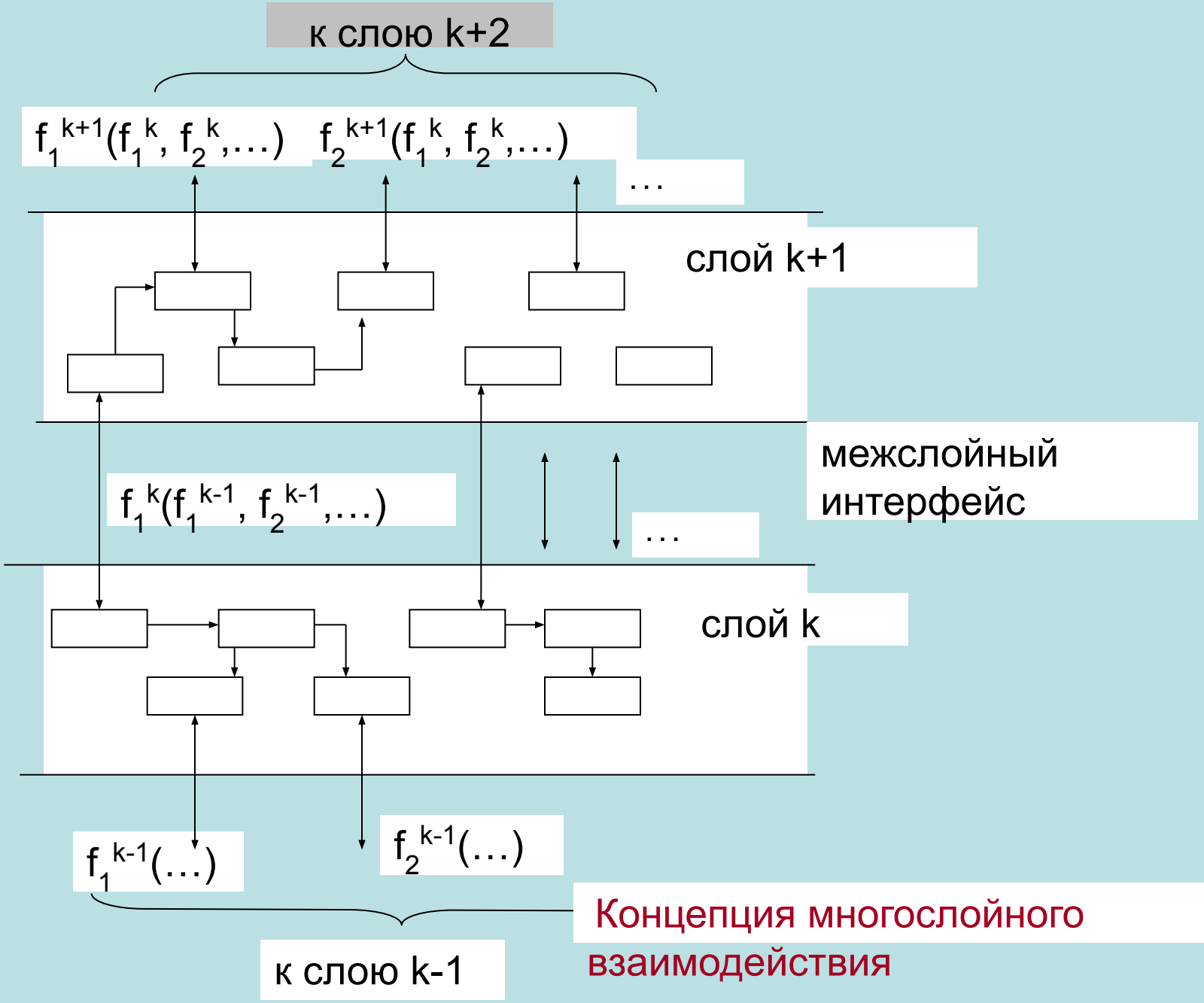
- резидентно находится в оперативной памяти (для повышение производительности работы системы)
- работает в привилегированном режиме («режиме ядра»), причины:
  - некорректно работающее приложение может вмешаться в работу ОС
  - Операционная система должна обладать исключительными полномочиями также для того, чтобы играть роль арбитра в споре приложений за ресурсы компьютера в мультипрограммном режиме

Отличия привилегированного и пользовательского режима работы

- запрет выполнения в пользовательском режиме некоторых критичных команд, связанных с переключением процессора с задачи на задачу, управлением устройствами ввода-вывода, доступом к механизмам распределения и защиты памяти
- обеспечиваются привилегии ОС при доступе к памяти (выполнение инструкции доступа к памяти для приложения разрешается, если инструкция обращается к области памяти, отведенной данному приложению операционной системой, и запрещается при обращении к областям памяти, занимаемым ОС или другими приложениями)

Разновидности ядер ОС





Концепция многослойного взаимодействия

«Многослойный» подход

- система состоит из иерархии слоев, каждый слой отвечает за выполнение функций определенного уровня иерархии
- Каждый слой обслуживает вышележащий слой, выполняя для него некоторый набор функций, которые образуют межслойный интерфейс
- строгие правила касаются только взаимодействия между слоями системы, а между модулями внутри слоя связи могут быть произвольными

Преимущества «многослойного» подхода

- Существенно упрощается разработка системы (сначала «сверху вниз»-определяются функции слоев и межслойные интерфейсы, затем «снизу вверх», детальная реализация слоев, наращивая их мощь)
- Простая модернизация системы (можно изменять модули внутри слоя без необходимости внесения изменений в других слоях)

Слои ядра

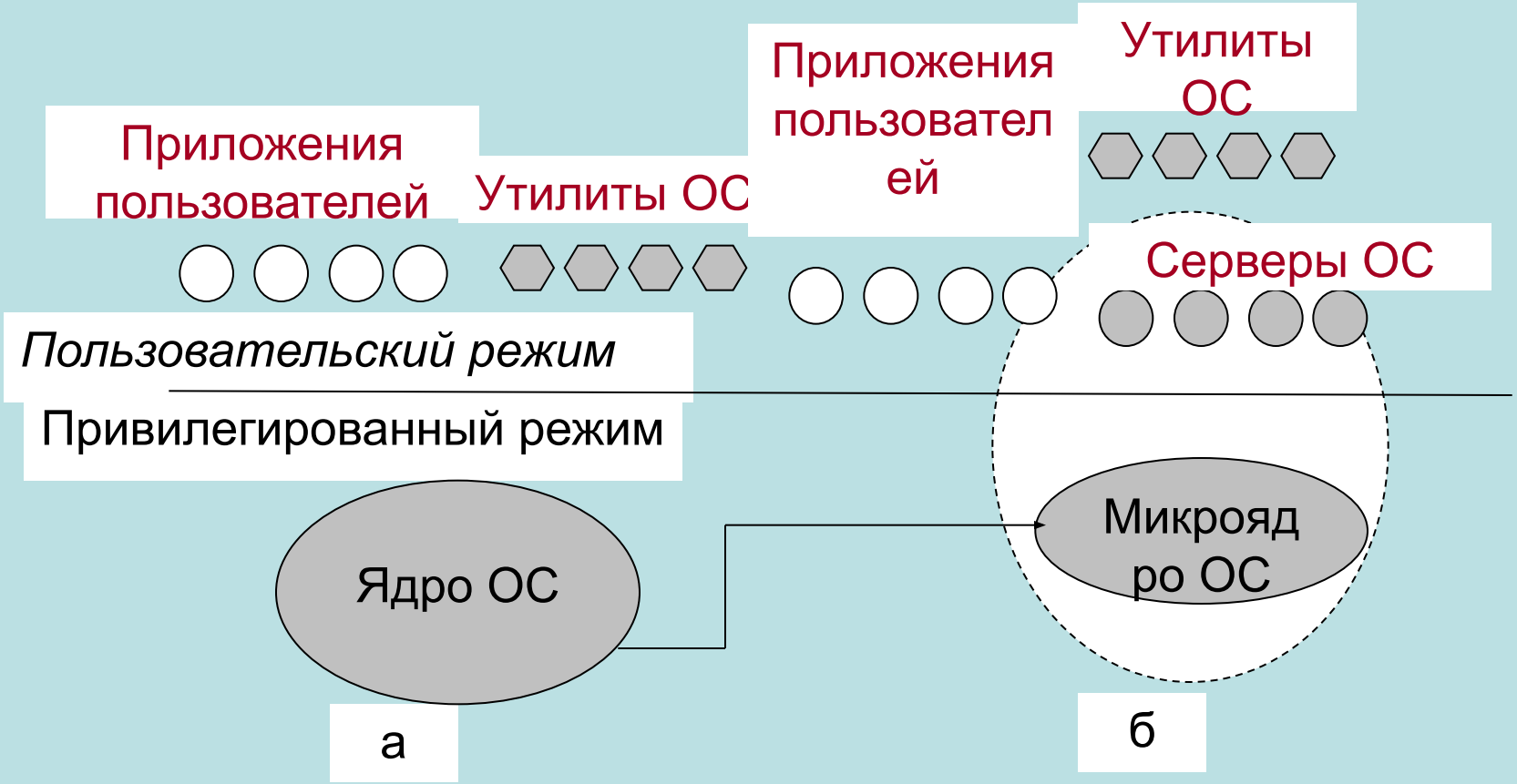
- Средства аппаратной поддержки ОС (например, средства поддержки привилегированного режима, система прерываний, средства поддержки переключения контекстов и т.д.).
- Машинно-зависимые компоненты ОС (программные модули, в которых отображается специфика аппаратной платформы ПК)
- Базовые механизмы ядра (примитивные операции ядра: переключение контекстов, диспетчеризация прерываний, перемещение страниц из памяти на диск)
- Менеджеры ресурсов (реализует стратегические задачи по управлению ресурсами)
- Интерфейс системных вызовов (непосредственно взаимодействует с приложениями и системными утилитами)

Классическая архитектура (на базе ядра)

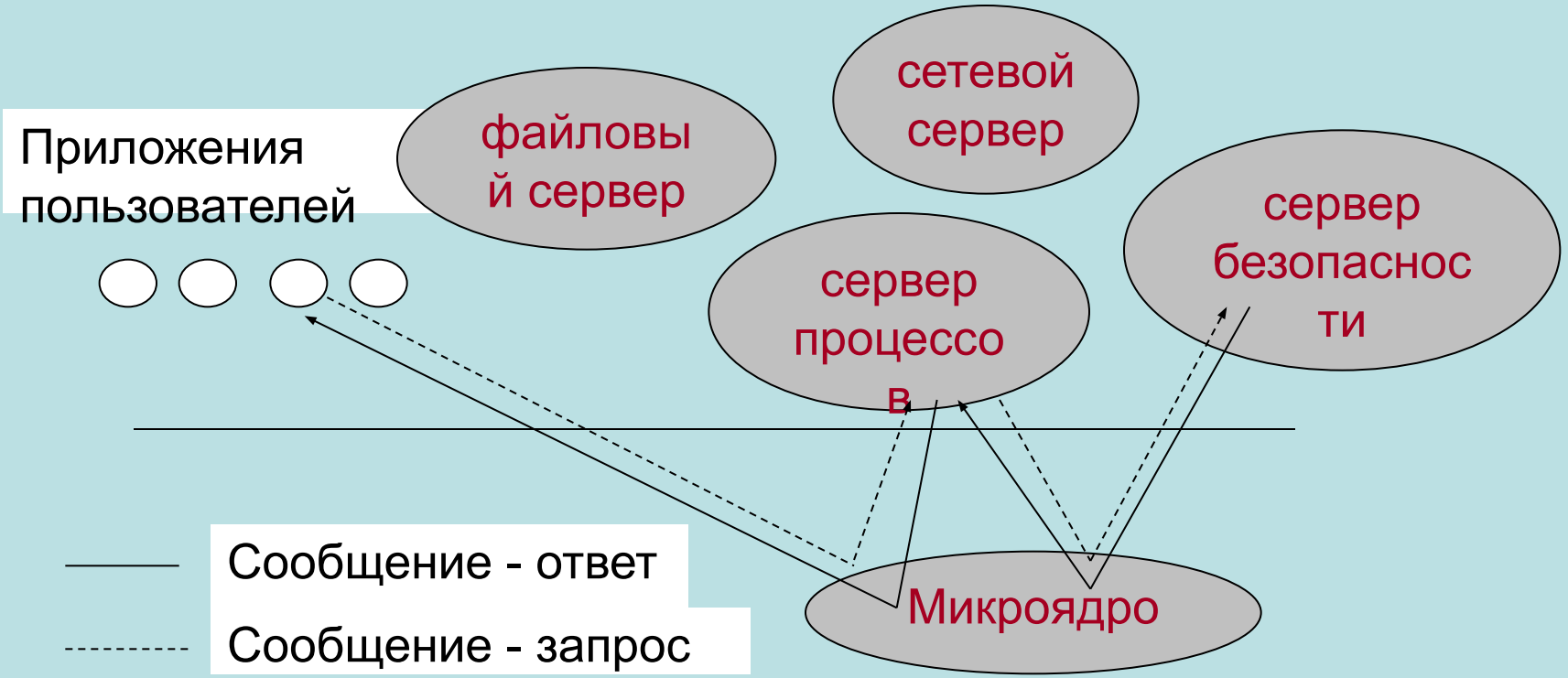
Выводы:

- Все основные функции операционной системы, составляющие многослойное ядро, выполняются в привилегированном режиме.
- Некоторые вспомогательные функции ОС оформляются в виде приложений и выполняются в пользовательском режиме наряду с обычными пользовательскими программами (становясь системными утилитами или обрабатывающими программами).
- Каждое приложение пользовательского режима работает в собственном адресном пространстве и защищено тем самым от какого-либо вмешательства других приложений.
- Код ядра, выполняемый в привилегированном режиме, имеет доступ к областям памяти всех приложений, но сам полностью от них защищен. Все основные функции операционной системы, составляющие многослойное ядро, выполняются в привилегированном режиме.
- Приложения обращаются к ядру для выполнения системных функций

Перенос функций ядра в пользовательское пространство (микроядро)



Микроядерная архитектура (реализация системного вызова)



Преимущества и недостатки микроядерной архитектуры

Преимущества:

- переносимость
- расширяемость
- повышение надежности

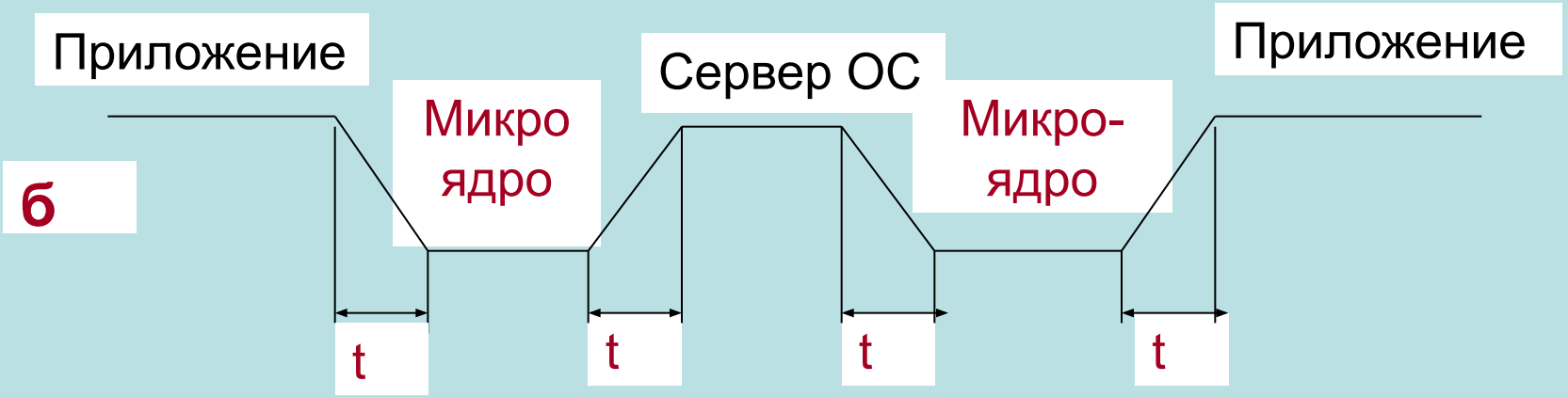
Недостаток:

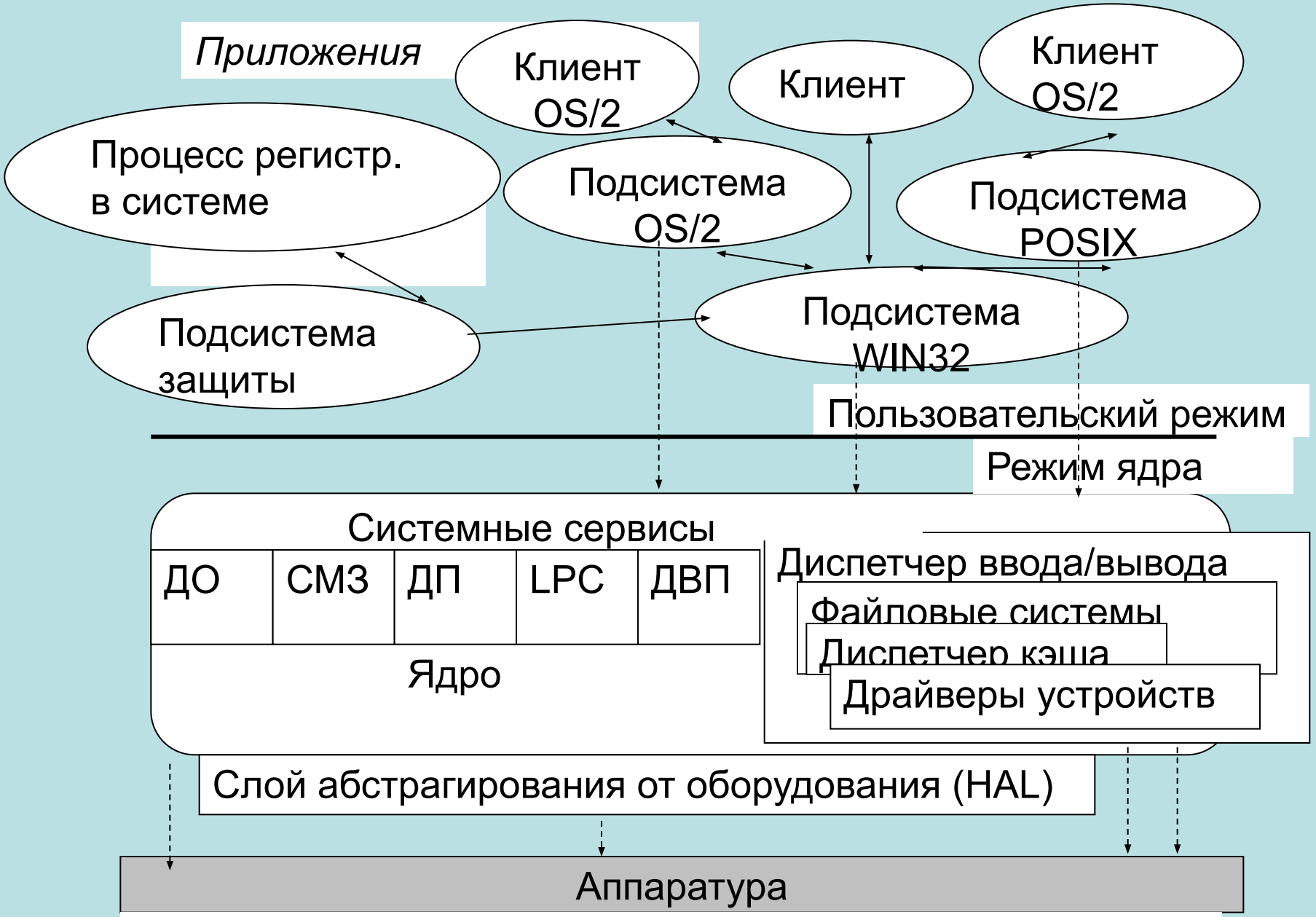
- понижение производительности

Причины потери производительности при микроядерной архитектуре

а

б





. Структурная схема операционной системы Windows NT

Исполнительная подсистема

- Диспетчер объектов (ДО). Создает, поддерживает и уничтожает объекты исполнительной системы NT
- Справочный монитор защиты (СМЗ). Гарантирует выполнение политики защиты на локальном компьютере
- Диспетчер процессов (ДП). Создает и завершает процессы и потоки
- Средство локального вызова процедур (LPC). Передает сообщения между клиентскими и серверными процессами, расположенными на одном и том же компьютере
- Диспетчер виртуальной памяти (ДВП). Реализует *виртуальную память*
- Слой абстрагирования от оборудования (HAL). Помещает кодовую прослойку между исполнительной системой NT и аппаратной платформой, на которой работает ОС, скрывает аппаратно – зависимые детали