

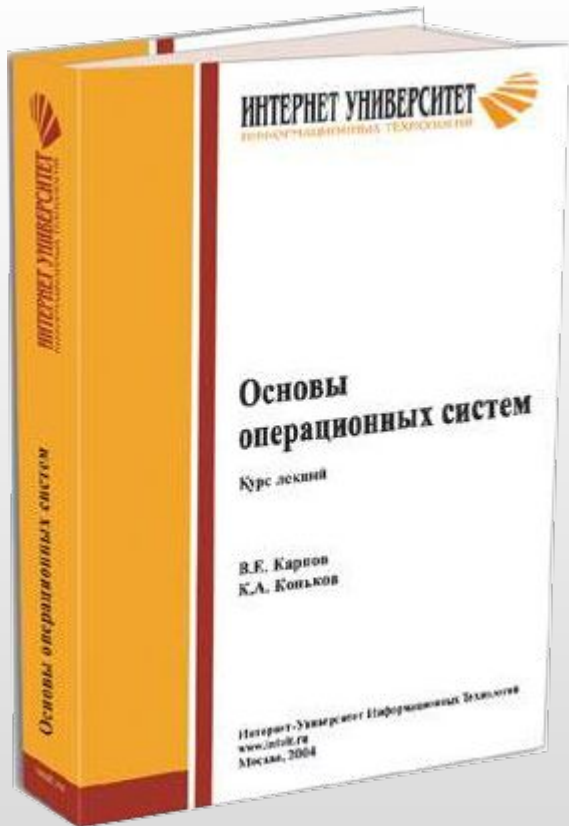
ОСНОВЫ Операционных Систем

Лектор – к.ф.-м.н, доцент Карпов Владимир Ефимович
carpson@mail.ru

ВШЭ-2019

Литература к курсу

Основная



В.Е.Карпов
К.А.Коньков

Основы операционных систем

Лекции:

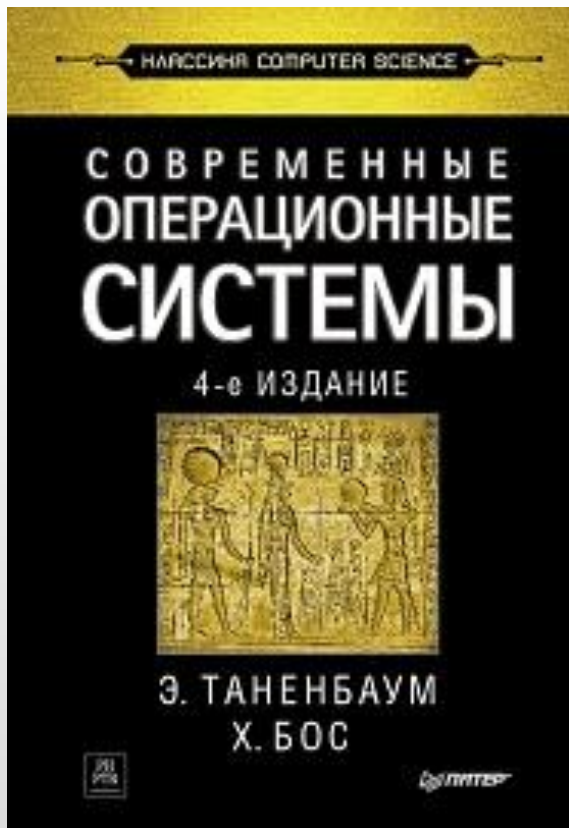
<http://www.intuit.ru/studies/courses/2192/31/info>

Практика:

<http://www.intuit.ru/studies/courses/2249/52/info>

Литература к курсу

Дополнительная



Э.Таненбаум
Х.Бос

Современные
операционные системы

Литература к курсу

Дополнительная

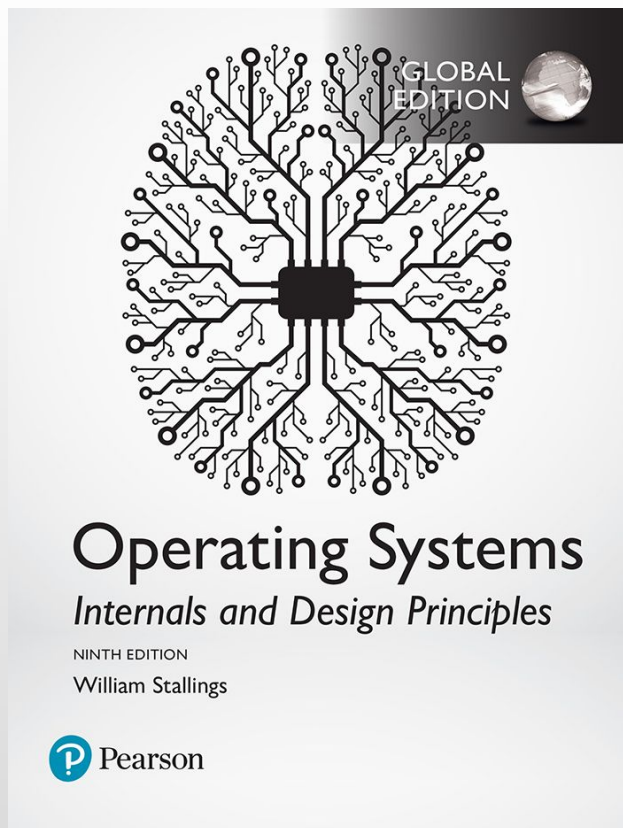


Вильям Столлингс

Операционные системы

Литература к курсу

Дополнительная

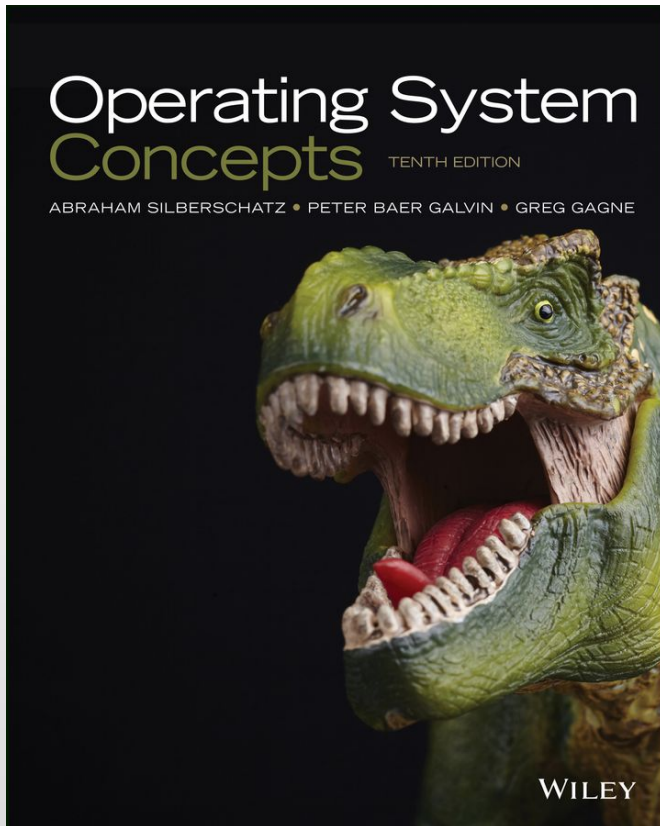


William Stallings

Operating Systems
Internals and Design Principles
Ninth Edition

Литература к курсу

Дополнительная



Avi Silberschatz
Peter Baer Galvin
Greg Gagne

Operating System Concepts
Tenth edition

Тема 1

Обзор

Введение

Структура вычислительной системы

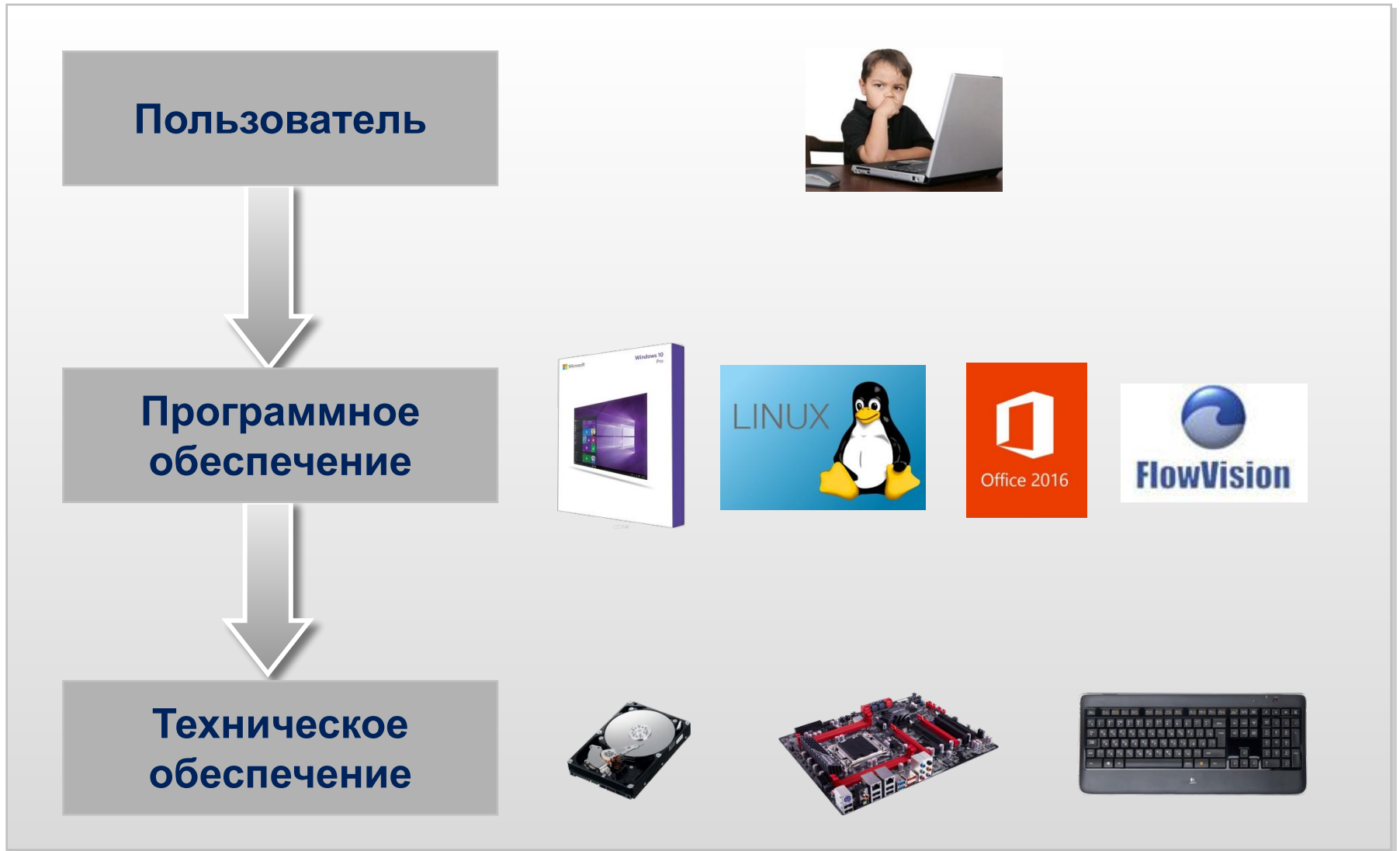
Пользователь



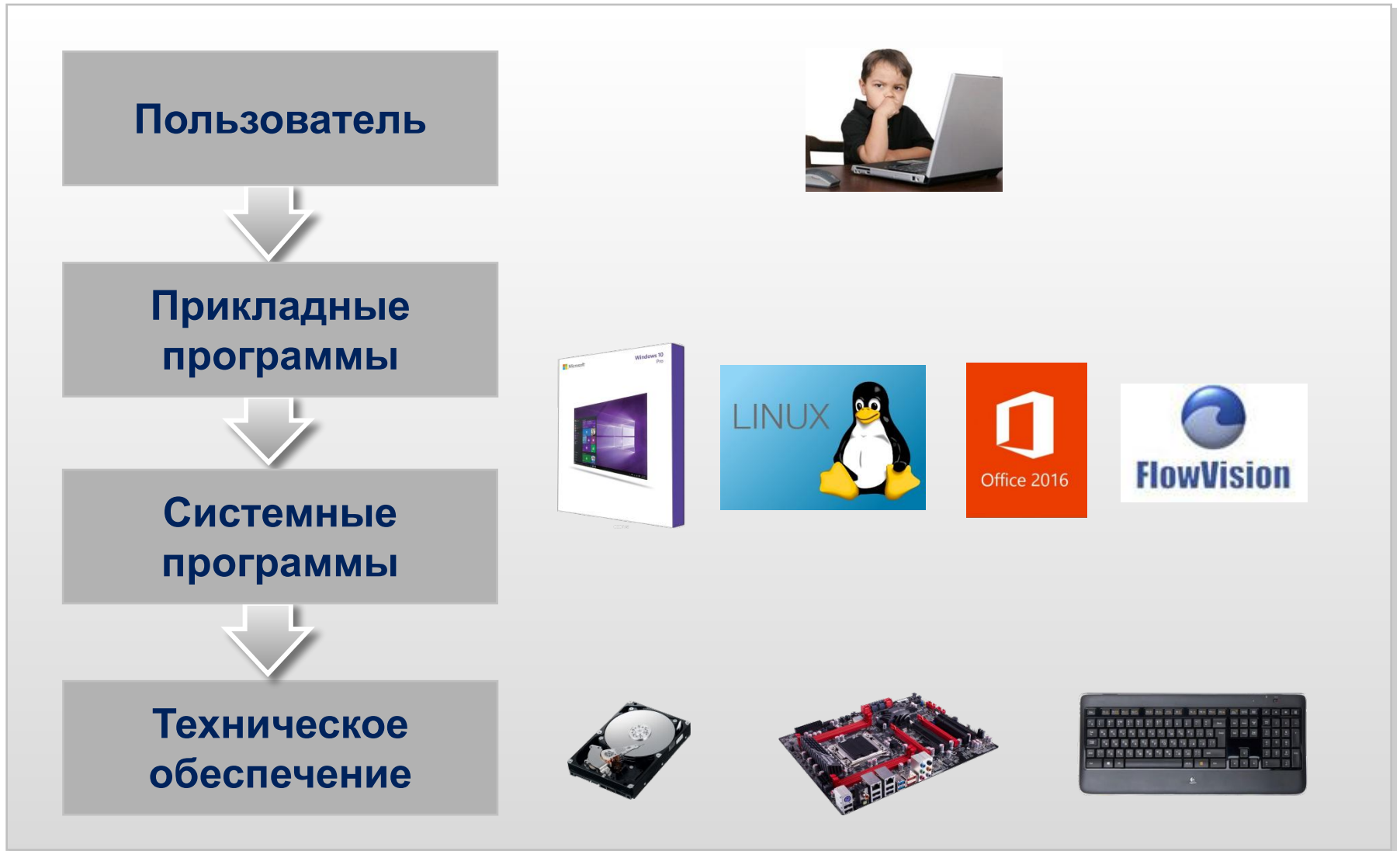
Техническое
обеспечение



Структура вычислительной системы



Структура вычислительной системы



Структура вычислительной системы

Пользователь



Прикладные программы

Прочие системные программы



Операционная система

Техническое обеспечение



Структура вычислительной системы



Что такое операционная система?

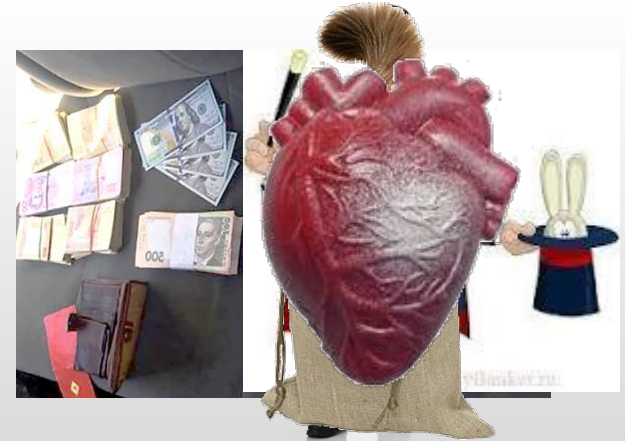
Слон и слепцы



Что такое операционная система?

Основные точки зрения

- Распорядитель ресурсов
- Защитник пользователей и программ
- Виртуальная машина
- Кот в мешке
- Постоянно функционирующее ядро

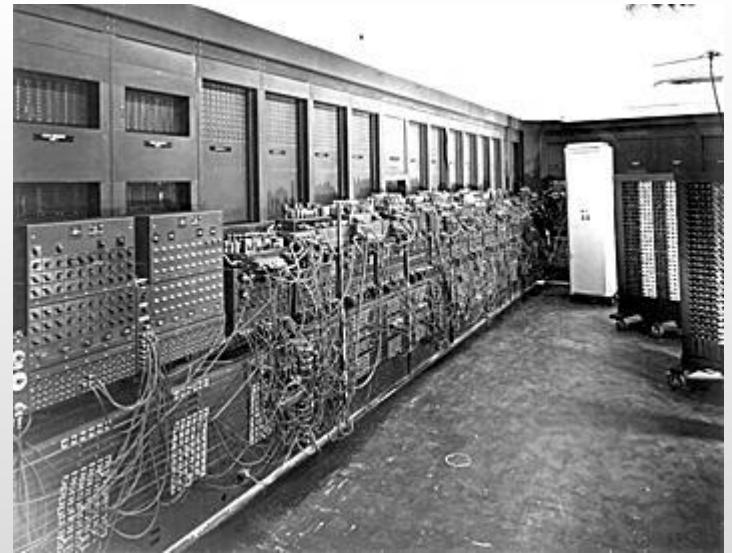


Проще сказать, не что такое есть операционная система, а для чего она нужна, и что она делает

Эволюция вычислительных систем

1-й период (1945 – 1955 гг.)

- Ламповые машины
- Нет разделения персонала
- Ввод программы коммутацией или перфокартами
- Одновременное выполнение только одной операции
- Появление прообразов первых компиляторов
- Нет операционных систем



Научно-исследовательская работа в области
вычислительной техники

Эволюция вычислительных систем

2-й период (1955 – начало 60х гг.)

- Транзисторные машины
- Происходит разделение персонала
- Бурное развитие алгоритмических языков
- Ввод задания колодой перфокарт
- Вывод результатов на печать
- Пакеты заданий и системы пакетной обработки



Начало использования ЭВМ в научных и коммерческих целях

Эволюция вычислительных систем

3-й период (начало 60х – 1980 гг.)

- Машины на интегральных схемах
- Использование спулинга (spooling)
- Планирование заданий
- Мультипрограммные пакетные системы



Мультипрограммирование и ЭВОЛЮЦИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Software

- Планирование заданий
- Управление памятью
- Сохранение контекста
- Планирование использования процессора
- Системные вызовы
- Средства коммуникации
- Средства синхронизации

Hardware

- Защита памяти
- Сохранение контекста
- Механизм прерываний
- Привилегированные команды

Эволюция вычислительных систем

3-й период (начало 60х – 1980 гг.)

- Машины на интегральных схемах
- Использование спулинга (spooling)
- Планирование заданий
- Мультипрограммные пакетные системы
- Системы разделения времени (time-sharing)
- Виртуальная память
- Интерактивная отладка программ, файловые системы
- Семейства ЭВМ



Широкое использования ЭВМ в научных и коммерческих целях

Эволюция вычислительных систем

4-й период (1980 – 2005 гг.)

- Машины на больших интегральных схемах (БИС)
- Персональные ЭВМ
- Дружественное программное обеспечение
- Сетевые и распределенные операционные системы



Широкое использования ЭВМ в быту, в образовании, на производстве

Эволюция вычислительных систем

5-й период (2005 - ?? гг.)

- Машины на многоядерных процессорах
- Мобильные компьютеры
- Высокопроизводительные вычислительные системы
- Облачные технологии
- Виртуализация выполнения программ



Глобальная компьютеризация

Основные функции ОС

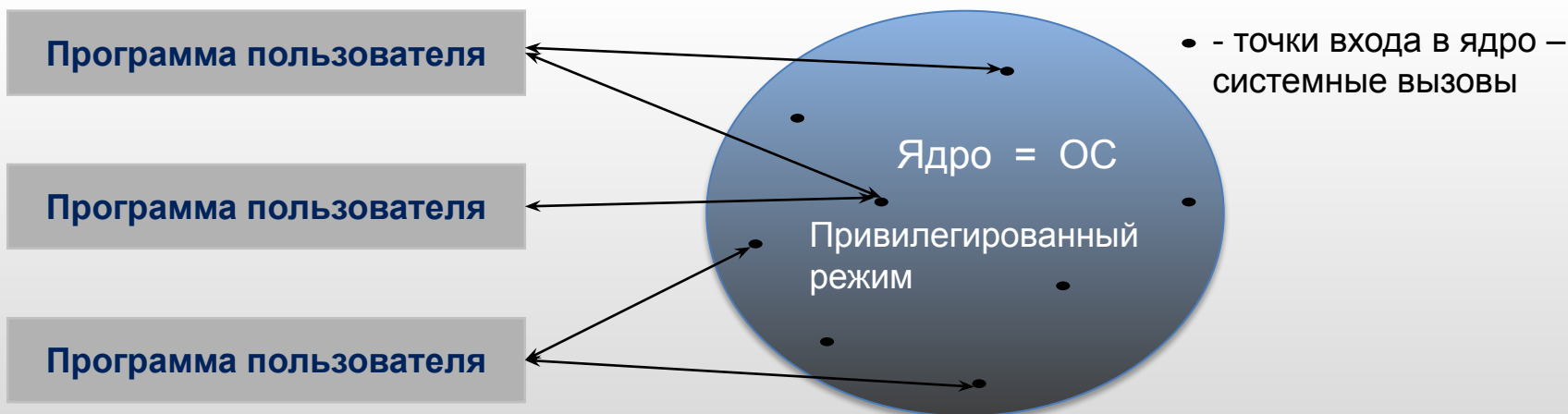
- Планирование заданий и использования процессора
- Обеспечение программ средствами коммуникации и синхронизации
- Управление памятью
- Управление файловой системой
- Управление вводом-выводом
- Обеспечение безопасности

Операционные системы существуют потому, что на данный момент их существование – это разумный способ использования вычислительных систем

Внутреннее строение ОС

Монолитное ядро

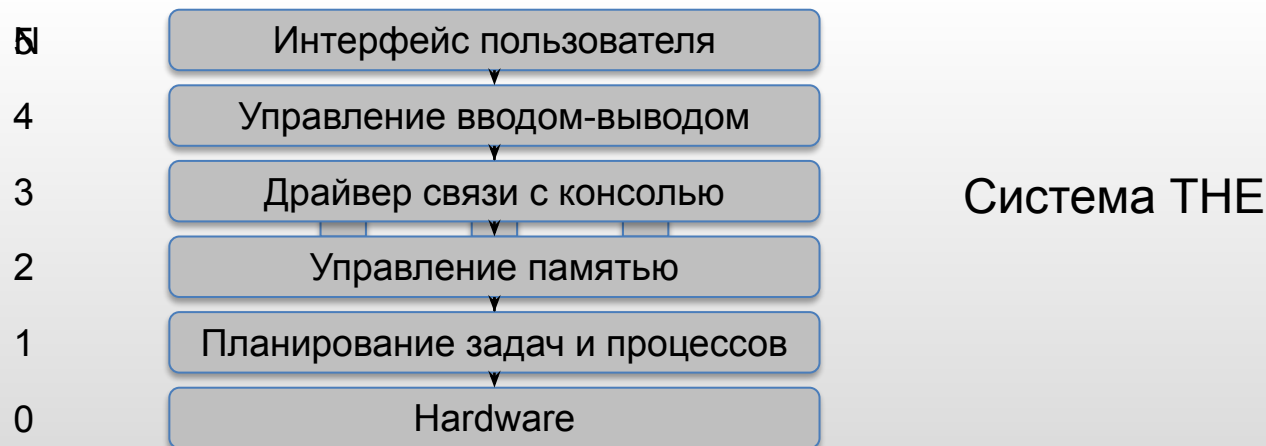
- Каждая процедура может вызывать каждую
- Все процедуры работают в привилегированном режиме
- Ядро совпадает со всей операционной системой
- Пользовательские программы взаимодействуют с ядром через системные вызовы



Внутреннее строение ОС

Многоуровневые (Layered) системы

- Процедура уровня K может вызывать только процедуры уровня K - 1
- Все или почти все уровни работают в привилегированном режиме
- Ядро совпадает или почти совпадает со всей операционной системой
- Пользовательские программы взаимодействуют с ОС через интерфейс пользователя

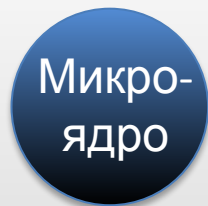


Внутреннее строение ОС

Микроядерная (microkernel) архитектура

Функции микроядра:

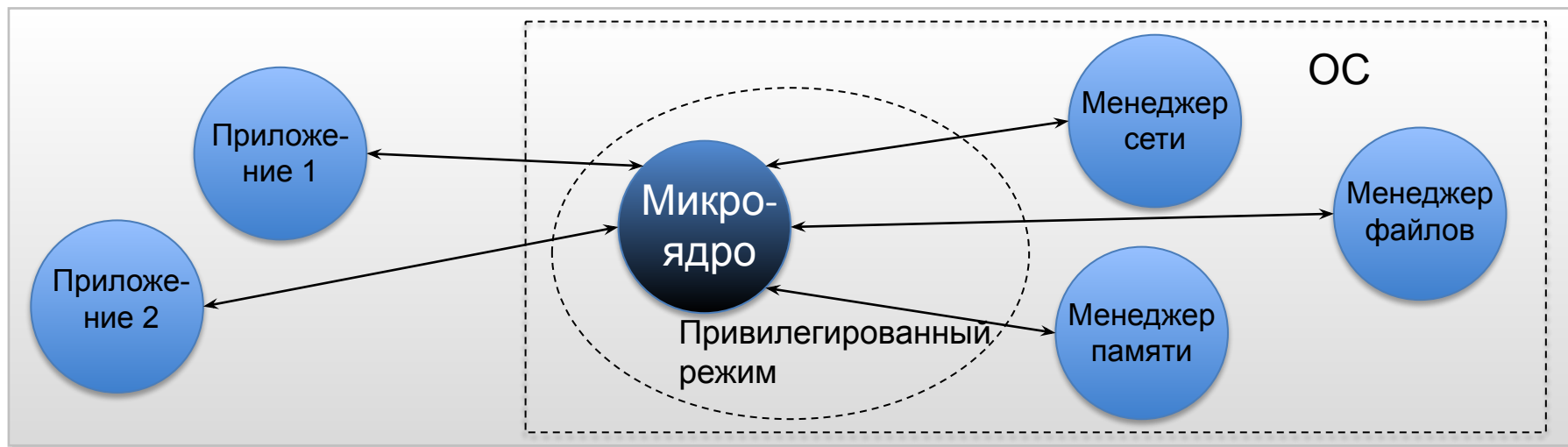
- взаимодействие между программами
- планирование использования процессора
- первичная обработка прерываний и операций ввода-вывода
- базовое управление памятью



Внутреннее строение ОС

Микроядерная (microkernel) архитектура

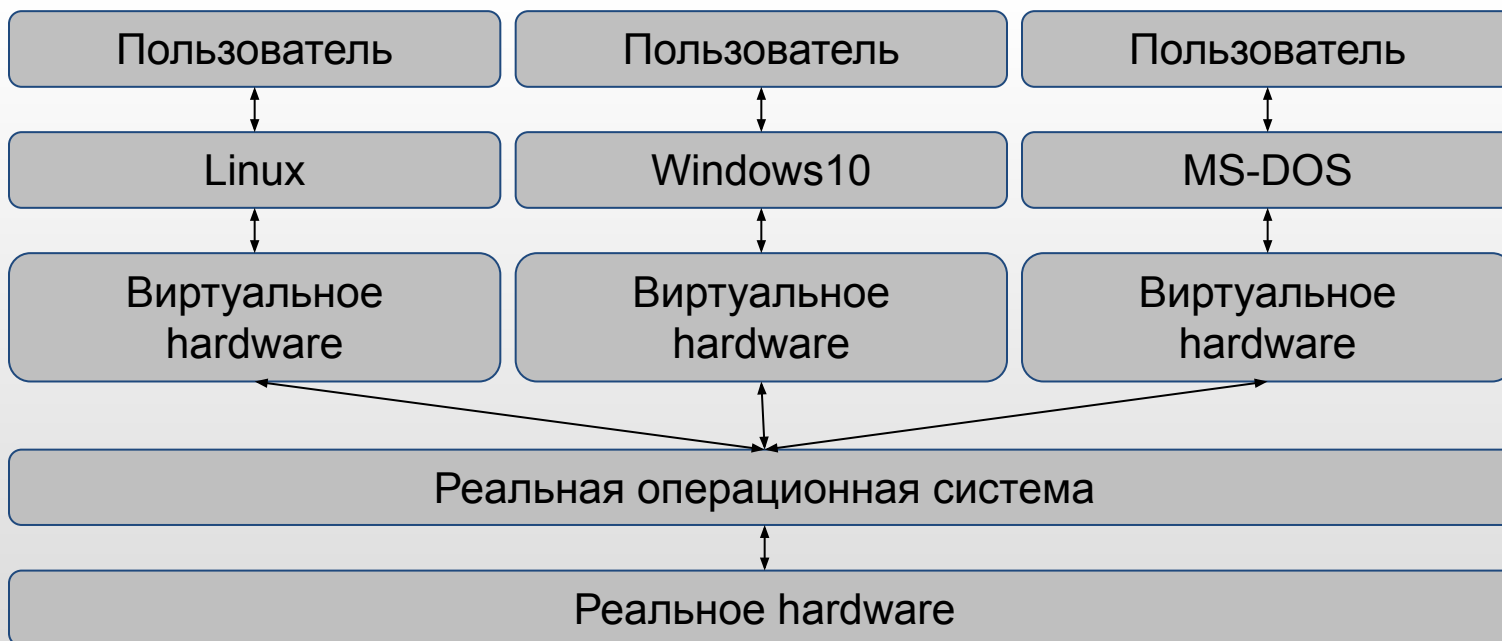
- Микроядро составляет лишь малую часть ОС
- В привилегированном режиме работает только микроядро
- Взаимодействие частей ОС между собой и с программами пользователей путем передачи сообщений через микроядро



Внутреннее строение ОС

Виртуальные машины

Каждому пользователю предоставляется своя копия виртуального hardware

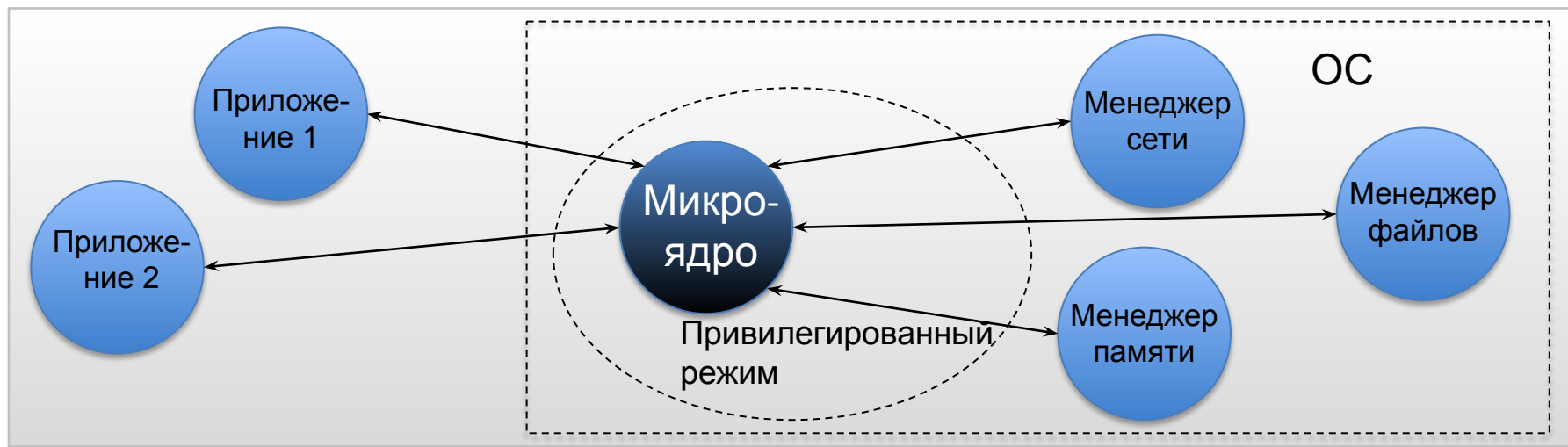


Внутреннее строение ОС

Новая микроядерная архитектура

Функции микроядра:

- взаимодействие между программами
- планирование использования процессора
- первичная обработка прерываний и операций ввода-вывода
- базовое управление памятью

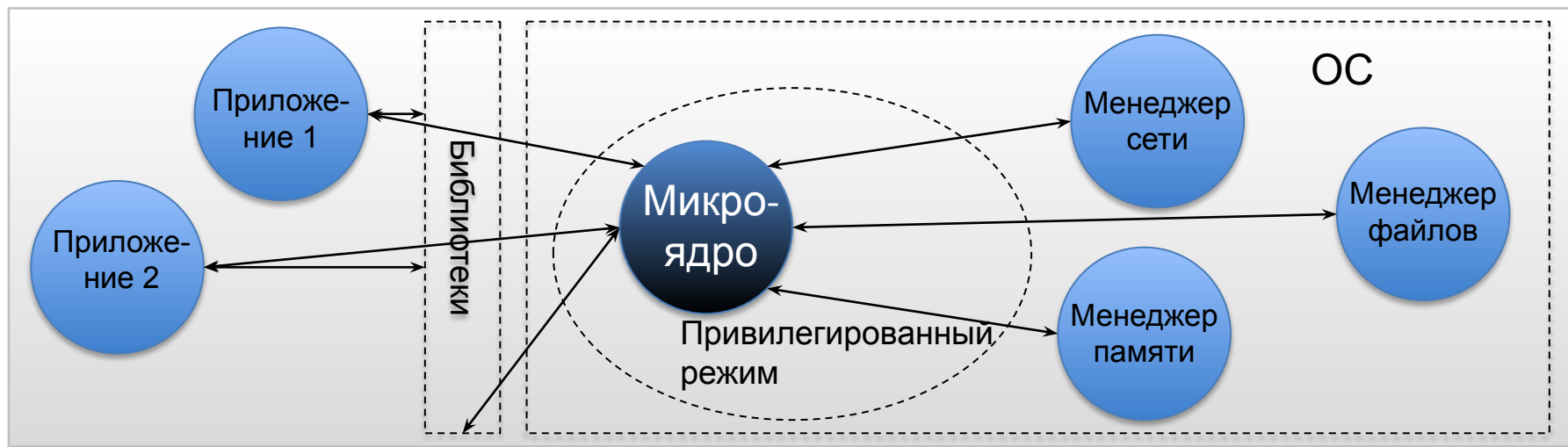


Внутреннее строение ОС

Новая микроядерная архитектура

Функции микроядра:

- взаимодействие между программами
- выделение и высвобождение физических ресурсов
- контроль прав доступа

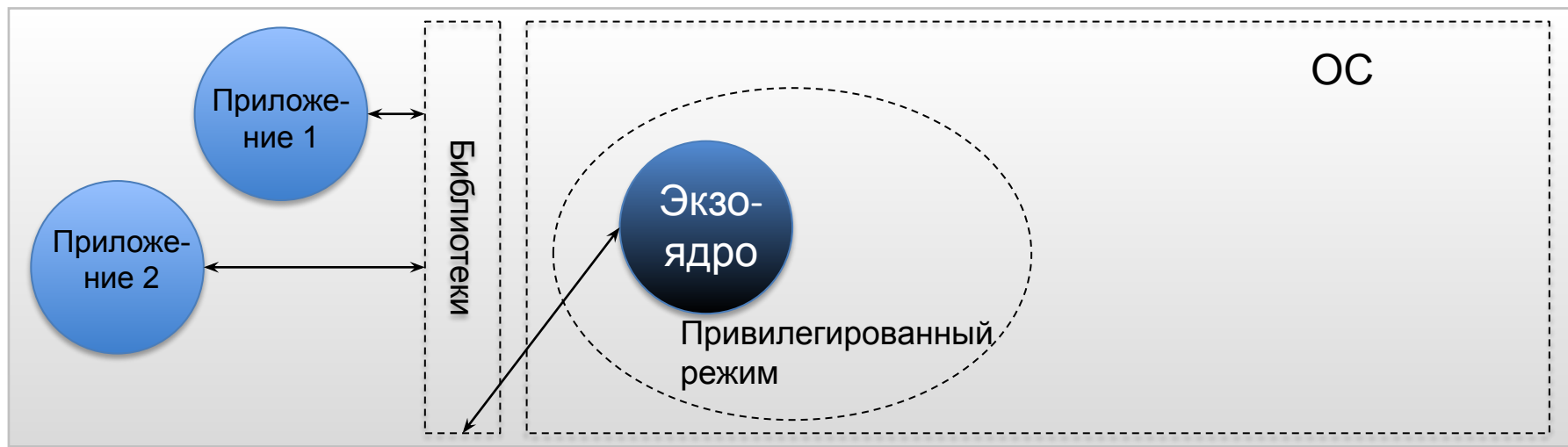


Внутреннее строение ОС

Экзоядерная архитектура

Функции экзоядра:

- взаимодействие между программами
- выделение и высвобождение физических ресурсов
- контроль прав доступа



Внутреннее строение ОС

Смешанные системы – почему?

- Монолитное ядро – необходимость перекомпиляции при каждом изменении, сложность отладки, высокая скорость работы.
- Многоуровневые системы – необходимость перекомпиляции при изменениях, отлаживается только измененный уровень, меньшая скорость работы
- Микроядро – простота отладки, возможность замены компонент без перекомпиляции и остановки системы, очень медленные

