



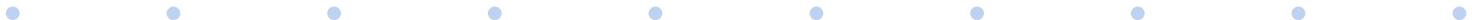
*Лекция 1:* **Понятие ОС. Функции и классификация**

*Курс «Операционные системы»*



# План лекции

- Что такое операционная система (ОС)
  - Структура вычислительной системы
  - Что же такое ОС?
- Эволюция вычислительных систем
- Основные понятия и концепции ОС
- Архитектура ОС
  - Монолитное ядро
  - Слоеные системы (Layered systems)
  - Виртуальные машины
  - Микроядерная архитектура
  - Смешанные системы
  - Структура ОС
- Классификация ОС
  - Семейства ОС
- Выводы





# Что такое операционная система



# Структура вычислительной системы

- ЭВМ = hardware + software
- *Hardware*:
  - Процессор
  - Память
  - Дисковые устройства
  - ...

} Объединенные шиной
- *Software* (программное обеспечение, ПО):
  - Прикладное
  - Системное - Инструментальные системы?





## Структура вычислительной системы- II

- Деление на прикладное и системное ПО во многом **условно**:
  - Microsoft Word
    - Для пользователя – системная программа, для программиста – приложение
  - С++ компилятор
    - Для прикладного программиста – системная программа, для системного – прикладная
- Таким образом, вычислительную систему можно представить в виде *последовательности слоев*





## Структура вычислительной системы- слои





# Что же такое ОС?

- Основные точки зрения:
  - ОС, как *виртуальная машина*
    - С помощью простых и ясных абстракций от программиста скрываются физические подробности работы компьютера
  - ОС, как *менеджер ресурсов*
    - Осуществляет упорядоченное и контролируемое распределение процессорного времени, памяти и др. между программами
  - ОС, как *защитник пользователей и программ*
    - Обеспечивает сохранность данных каждого пользователя и изолирует их выполняющиеся приложения друг от друга





## Что же такое ОС? - II

- Основные точки зрения:
  - ОС, как *постоянно функционирующее ядро*
    - Постоянно работающая на компьютере программа (или ее часть – ядро), взаимодействующая со всеми прикладными программами



**Не существует** точного адекватного определения, что такое ОС. Проще сказать, для чего нужна ОС и что делает.

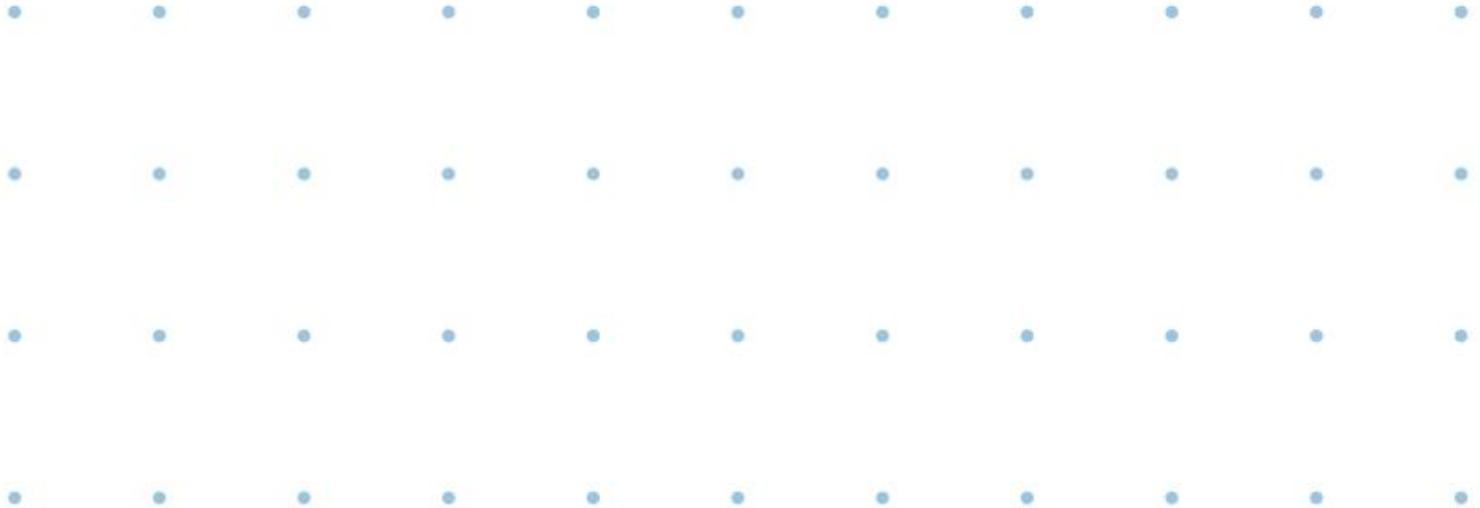




# Что же такое ОС? - определение

- Впрочем, можно довольствоваться таким определением:
  - **ОС** – это программа, загружающаяся при включении компьютера, управляющая всеми его ресурсами, реализующая диалог с пользователем, запускающая и выполняющая другие программы
  - Таким образом, ОС реализует такие *интерфейсы*:
    - Аппаратно-программный
    - Программно-программный
    - Интерфейс пользователя





# Эволюция вычислительных систем



## История эволюции ВТ

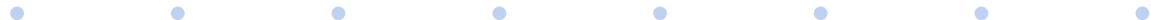
- Hardware и software эволюционируют *совместно*, влияя друг на друга
- Критерии естественного отбора в эволюции вычислительных систем:
  - Удобство
  - Эффективность
  - Безопасность
- История эволюции ВТ насчитывает *четыре периода*





# История эволюции ВТ – 1-й период

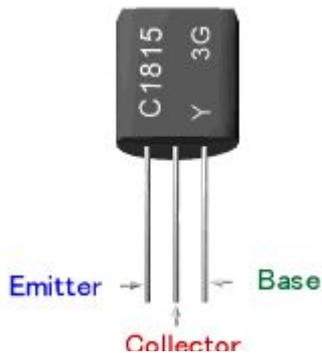
- **1-й период (1945-1955): Ламповые машины. ОС отсутствуют**
  - Июнь 1945 – принципы Джона фон Неймана
  - Программирование только на машинном языке
  - Перфокарты или панели переключателей
  - Организация вычислительного процесса выполнялась программистом с пульта управления
  - За пультом – только один человек
    - одновременно выполняется только одна операция
    - не системное ПО:
      - 1051-52 гг. – прообразы компиляторов, например, с Fortran
      - 1954г. – ассемблер для IBM-701 (Нэт Рочестер)





# История эволюции ВТ – 2-й период

- **2-й период (1955-нач. 60-х): Компьютеры на основе транзисторов. Пакетные ОС**



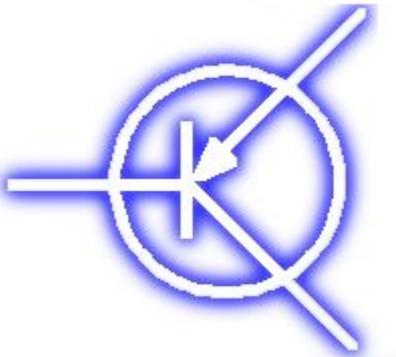
Новая элементная база – транзисторы:

- Повышение надежности ВТ
- Снижение энергопотребления
- Упрощение систем охлаждения
- Уменьшение размеров компьютеров
- Снижение стоимости их эксплуатации и обслуживания

Компьютеризация бизнеса

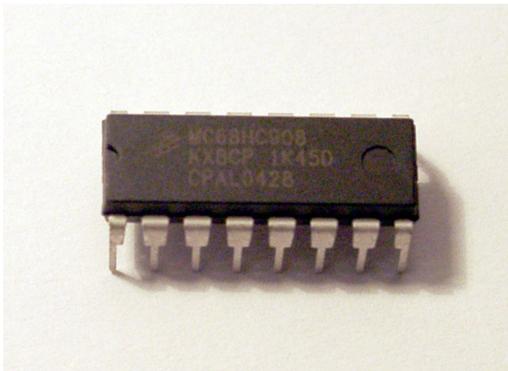
Бурное развитие алгоритмических языков (Algol-58, Lisp, Cobol, Algol-60, PL-1, ...), *настоящие* инструментальные системы

- Разделение персонала на программистов и операторов
- Задания (программа, входные данные, требуемые ресурсы), пакеты заданий
- Вычислительным процессом управляют *системы пакетной обработки*, имеющие формализованный язык управления заданиями





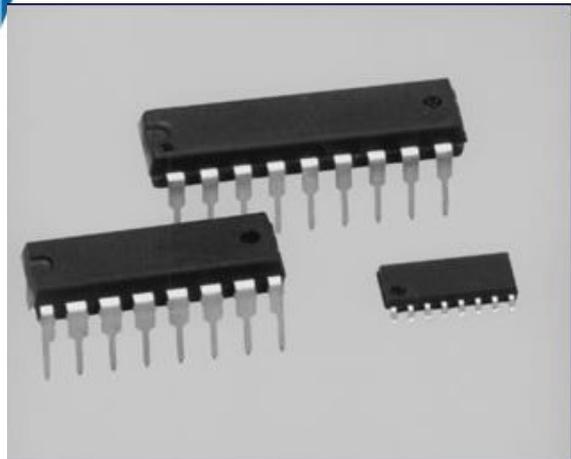
- **3-й период (нач. 60-х-1980): Компьютеры на основе ИМС. Первые многозадачные ОС**
  - Переход от транзисторов к интегральным микросхемам (ИМС):
    - ВТ становится надежной и дешевой
    - Решаются многочисленные и сложные задачи
    - Растет производительность процессоров
  - «Бутылочное горлышко» - низкая скорость устройств ввода-вывода  $\Rightarrow$  буферизация
    - Режим *off-line* – операции ввода-вывода выполняются на отдельно стоящих компьютерах
    - On-line – spooling (Simultaneous Peripheral Operation On-Line)
      - на том же компьютере, который производит вычисления
        - прерывания





## История эволюции ВТ – 3-й период II

- Магнитные диски сменили магнитную ленту
  - Планирование заданий
- Мультипрограммирование
  - Пока одна программа выполняет операцию ввода-вывода, процессор выполняет другую программу
  - *Аналогия:* преподаватель и студенты на экзамене
  - Аппаратная поддержка:
    - Защитные механизмы
      - » Привилегированные и непривилегированные команды
    - Прерывания
      - » *Внешние* – произошло асинхронное событие
      - » *Внутренние* (исключительные ситуации) – ошибки, нарушения защиты,...





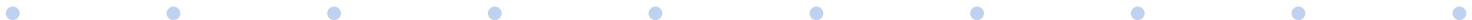
## История эволюции ВТ – 3-й период III

- Мультипрограммирование привело к изменению ОС:
  - *Системные вызовы*, как интерфейс между ОС и прикладными программами
  - *Планирование заданий*
  - Необходимость сохранения *контекста* (содержимого регистров и структур данных) при переключении задач
  - *Стратегии управления памятью*
  - Средства *коммуникации и синхронизации* между программами
- Все же пользователь **не мог** непосредственно взаимодействовать с заданием 😞





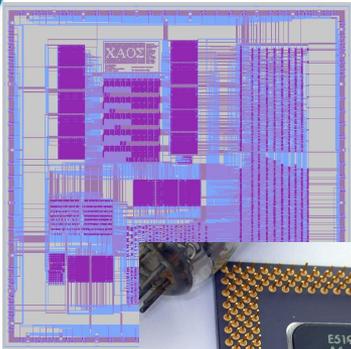
- Появились *дисплеи* на основе ЭЛТ и полноценные *клавиатуры*
- Мультипрограммирование  $\Rightarrow$  *time-sharing* -системы, или *системы разделения времени*
  - Процессор переключается между задачами не только во время операций ввода-вывода, но и просто через **определенные промежутки времени**
  - *Многопользовательский режим*
  - *Механизм виртуальной памяти*
  - *Развитые файловые системы*
- Семейства *программно-совместимых машин*
  - IBM/360 (OS/360  $\geq 1000$  известных ошибок)
  - PDP (PDP-11)





# История эволюции ВТ – 4-й период

- **4-й период (1980-наше время): Персональные компьютеры. Классические, сетевые и распределенные системы**
  - Появление БИС
    - Возрастание степени интеграции
    - Очередное удешевление
  - **Персональные компьютеры (ПК)**
    - Деградация ОС: один пользователь в однозадачном режиме
    - «Дружественный», интуитивно понятный интерфейс
    - Возрождение ОС
      - Автономные (классические) ОС
      - Сетевые ОС (сер. 80-х)
      - Распределенные ОС
    - ПК стал своего рода **бытовым прибором**

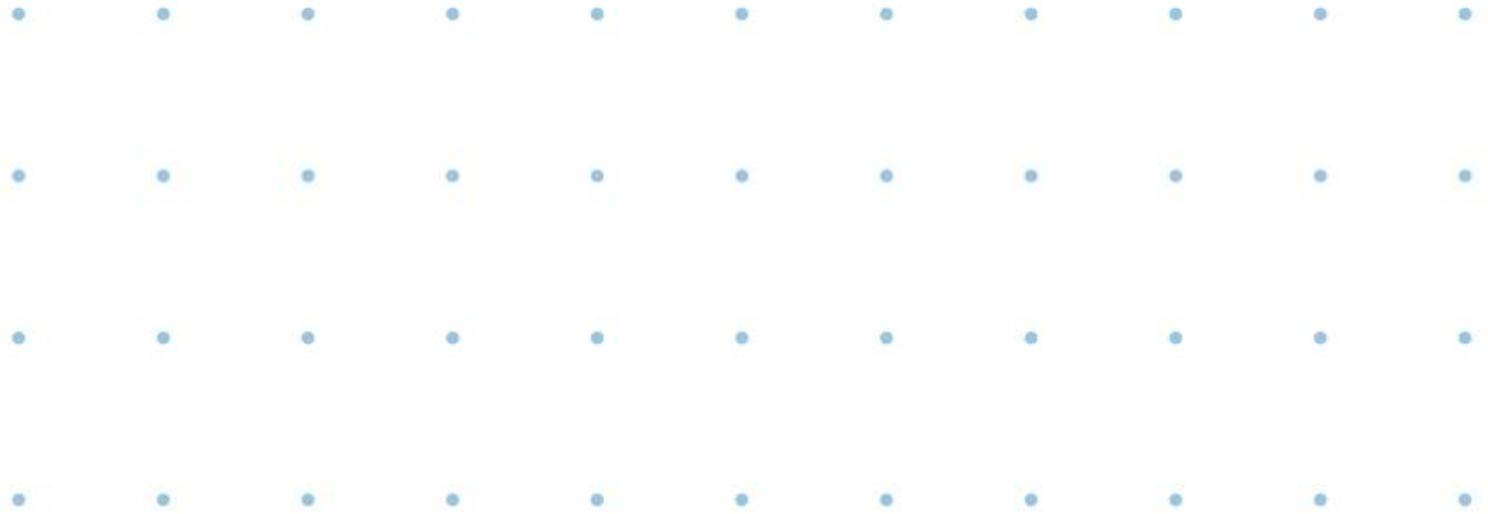




## История эволюции ВТ – выводы

- Пять основных функций классических ОС:
  - Планирование заданий
  - Средства коммуникации и синхронизации программ
  - Управление памятью
  - Файловая система
  - Управление вводом-выводом
  - Обеспечение безопасности
- Реальная эволюция проходила не так плавно и равномерно
- Это - не единственно возможный путь развития ОС!
- ОС существуют потому, что на данный момент это разумный способ использования ВТ





# Основные понятия и концепции ОС



# СИСТЕМНЫЕ ВЫЗОВЫ

- *System calls* – интерфейс между ОС и прикладной программой (сервис)
  - Создают, используют и удаляют различные объекты:
    - Процессы
    - Файлы
    - ...
  - ↳ Переход задачи в *привилегированный режим (kernel mode)*
  - ↳ осуществляется командой программного прерывания (INT) – *синхронное событие* (связано с выполняемым процессором кодом)

## System Calls

- The way user processes request services from the operating system.
- Services include:
  - Process Control
  - Load, Execute etc.
  - File Management
  - Create, Open, Close etc.
  - Communications
  - Send and receive data between processes





# Прерывания

- *Hardware interrupt* – событие, генерируемое внешним (по отношению к процессору) устройством
  - Например, пользователь нажал клавишу
  - Завершилась асинхронная операция ввода-вывода
  - Прерывания таймера
    - Планирование процессов
  - *Асинхронное* событие – возникает независимо от того, какой код выполняется процессором в данный момент





- *Exception* – возникает в результате выполнения программой недопустимой операции, доступа к ресурсу при отсутствии необходимых привилегий, или обращения к отсутствующей странице памяти
  - *синхронные* события, возникающие в контексте текущей задачи (как и системные вызовы)
    - *Исправимые* (нет нужных данных в памяти)
    - *Неисправимые* (ошибка в программе)  $\Rightarrow$  завершение программы





# Файлы

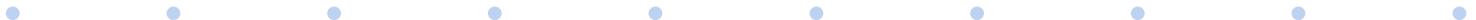
- *Файл* (file) – часть пространства на носителе информации, имеющая имя
- *Файловая система* (file system) – скрывает особенности ввода-вывода и дает программисту простую абстрактную модель файлов, независимую от устройств
  - Системные вызовы для работы с файлами
  - Понятия каталога, пути,...

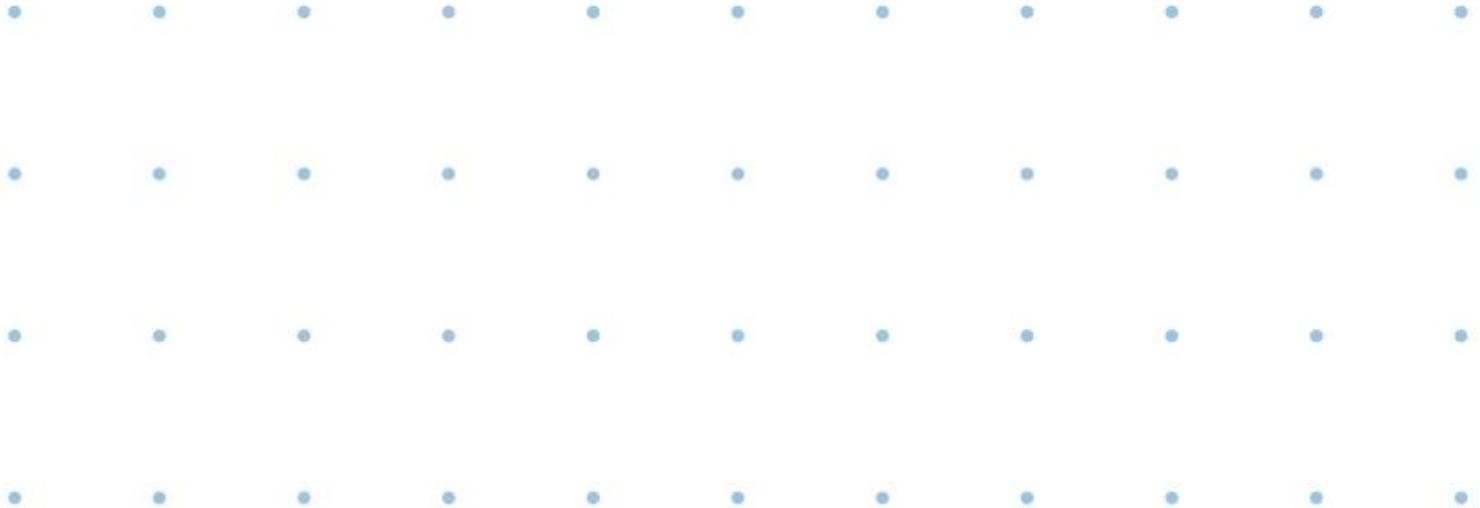




## Процессы и нити

- Процесс – наиболее фундаментальная концепция в ОС
- Нити (потoki, threads)
- Подробнее – в дальнейших лекциях



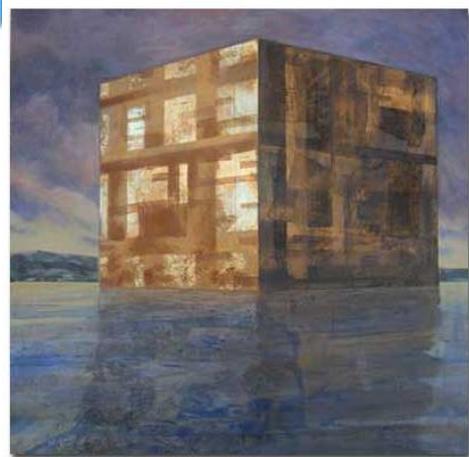


# Архитектура ОС



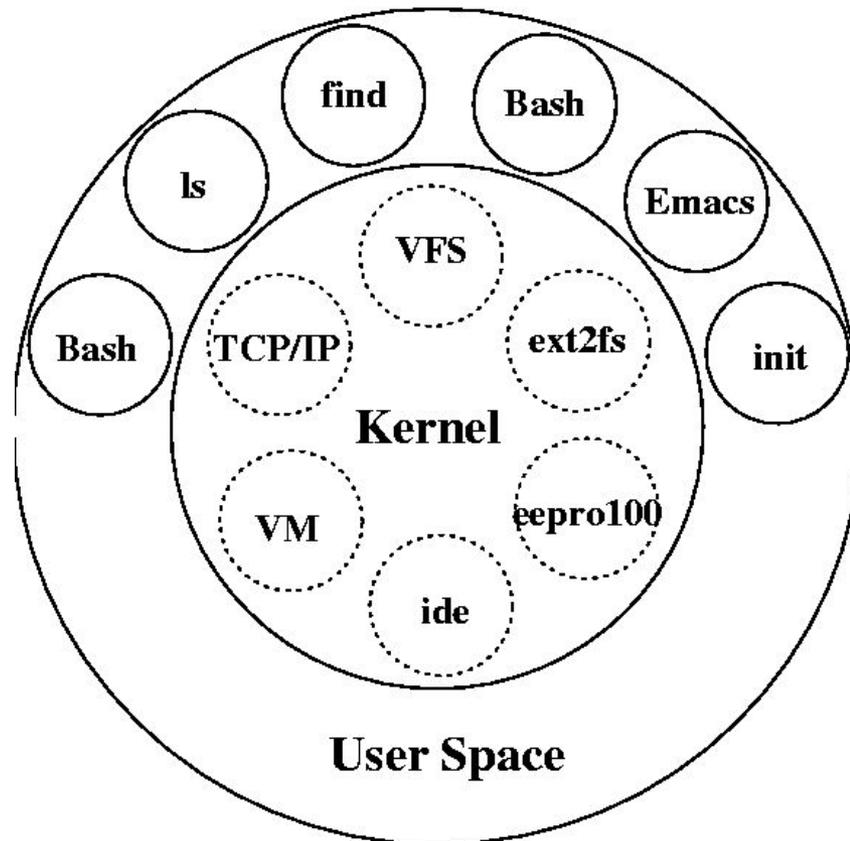
# Монолитное ядро

- *Monolithic kernel* – компоненты ОС – не отдельные модули, а части одной большой программы
  - Сборка (компиляция) ядра выполняется отдельно для каждого компьютера
    - Добавление новых компонентов и исключение неиспользуемых
    - Экономия памяти
    - Повышение надежности
  - **Быстродействие**
  - Старейший способ организации ОС (UNIX)
  - Все равно можно выделить некоторую структуру





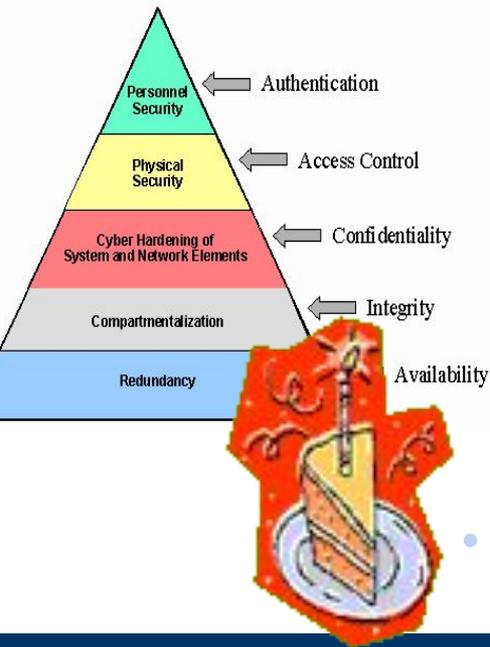
# Монолитное ядро - пример





# Слоеные системы

- *Layered system* – вычислительная система разбивается на ряд более мелких уровней с хорошо определенными связями
  - Каждый слой может вызывать объекты только из *нижнего ближайшего* к нему слоя
  - Система THE (Technische Hogeschool Eindhoven), Дейкстра и его студенты, 1968г.
  - Слоеные системы:
    - Хорошо реализуются
    - Хорошо тестируются
    - Хорошо модифицируются
    - Тяжелы в разработке (определение слоев)
    - **Менее эффективны, чем монолитные** (при вводе-выводе надо пройти все слои 😞)





# Слоеные системы - пример

- Структура ОС TNE:

5	Интерфейс пользователя
4	Управление вводом-выводом
3	Драйвер устройства связи оператора и консоли
2	Управление памятью
1	Планирование задач и процессов
0	Hardware





# Виртуальные машины

- *Виртуальная машина* для каждого пользователя – абсолютно «голая» копия всего hardware
  - Каждый может запустить свою собственную ОС
  - CP/CMS, или VM/370 (для семейства IBM/370)
  - Виртуальные машины **менее эффективны**, чем реальная
  - Громоздкость
  - Но! Можно использовать на одной машине программы, написанные для разных ОС





# Виртуальные машины - пример

- Вариант виртуальной машины:

Программа пользователя	Программа пользователя	Программа пользователя
MS-DOS	Linux	Windows-NT
Виртуальное hardware	Виртуальное hardware	Виртуальное hardware
Реальная операционная система		
Реальное hardware		



Virtual PC running virtual machines on a Mac



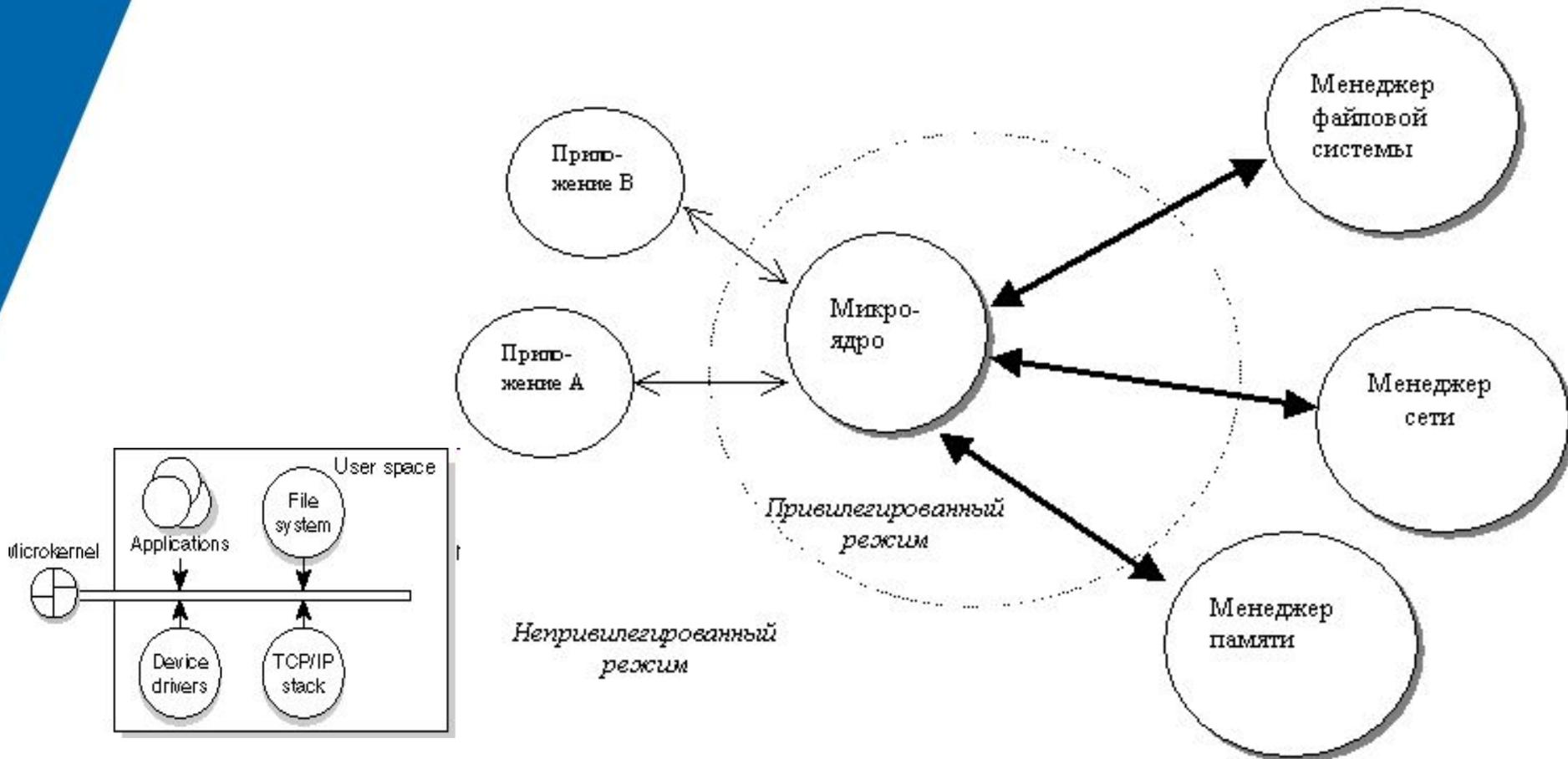
# Микроядерная архитектура

- *Microkernel architecture* – большинство составляющих ОС являются самостоятельными программами
  - Микроядро обеспечивает взаимодействие между ними через передачу *сообщений*
  - Работает в привилегированном режиме
    - Взаимодействие между программами
    - Планирование использования процессора
    - Первичная обработка прерываний
    - Операции ввода-вывода
    - Базовое управление памятью
  - Высокая степень модульности
    - Упрощается добавление новых компонент
    - Упрощается отладка
  - **Снижение производительности** из-за передачи сообщений
    - Следует аккуратно проектировать, чтобы минимизировать взаимодействие компонентов





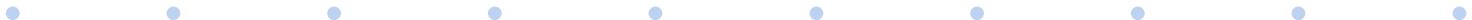
# Микроядерная архитектура - пример





# Смешанные системы

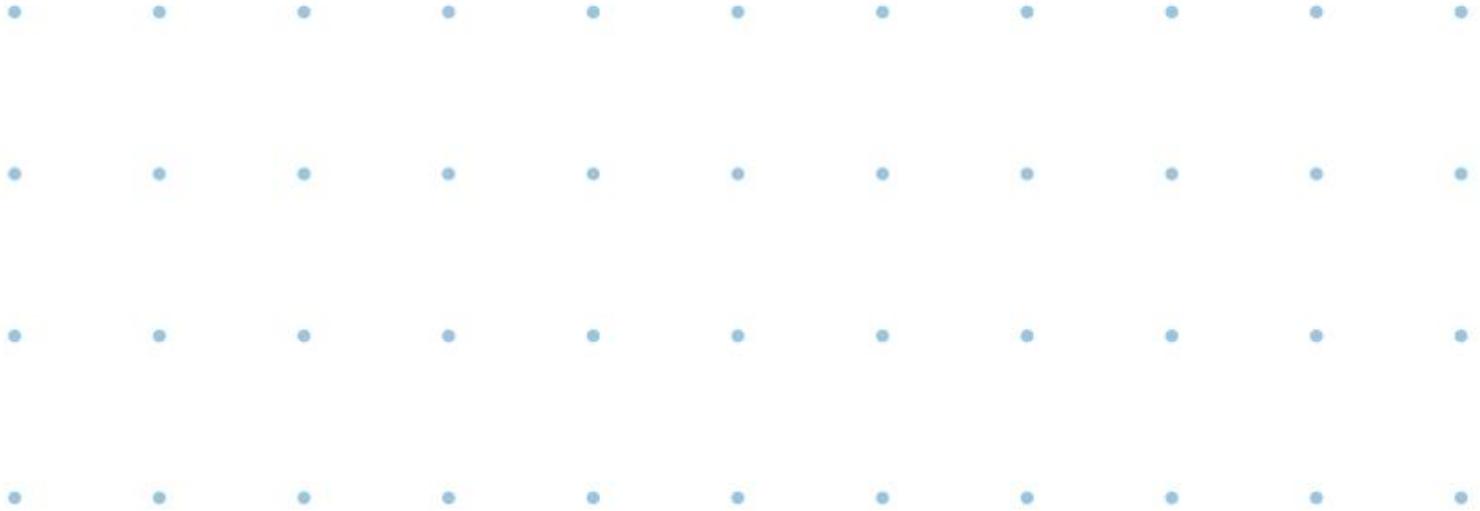
- Большинство современных ОС *комбинируют* вышеперечисленные варианты архитектуры
  - Linux – монолитная система с элементами микроядерной архитектуры
    - Модули ядра
  - Запуск ОС с монолитным ядром под управлением микроядра – 4.4 BSD, MkLinux (микроядро Mach)
  - Windows NT – тесное сплетение элементов микроядерной архитектуры и монолитного ядра
    - Сложное и большое ядро (более 1 Мб)
    - Передача сообщений между компонентами ОС
    - Но! Одно адресное пространство и общие структуры данных (как в монолитных ОС)
    - Режим ядра и режим пользователя (как в монолитных ОС)
    - Windows NT – **гибридная** ОС





# Структура ОС

- Даже в монолитных ОС можно выделить структуру
- **MS-DOS:**
  - *BIOS* – базовая система ввода-вывода. Встроена в ПЗУ машины. Выполняет наиболее простые и универсальные услуги ОС по вводу-выводу (Windows NT не использует BIOS)
  - *Загрузчик* – в 1-м секторе нулевой дорожки системного диска (для HDD это не так). Считывает в память еще два модуля ОС
  - *Резидентная часть* – файлы IO.SYS и MSDOS.SYS – остаются в памяти постоянно. IO.SYS – дополнение к BIOS, MSDOS.SYS – реализует основные высокоуровневые услуги ОС
  - *Командный процессор* (интерпретатор) – файл COMMAND.COM). Обрабатывает команды, вводимые пользователем
  - *Команды*
    - *Внутренние* – их командный процессор выполняет сам
    - *Внешние* – поставляются с ОС в виде отдельных программ
  - *Драйверы устройств* – расширяют возможности ОС по управлению устройствами, позволяют подключить новые устройства или нестандартным образом использовать имеющиеся
  - + *оболочки* (DOS Navigator, Norton Commander,...) и *операционные оболочки* (Windows, SEAL, Polit,...)



# Классификация ОС



# Классификация ОС

- Существует несколько схем классификации
- Будем классифицировать с точки зрения пользователя по нескольким критериям





# Реализация многозадачности

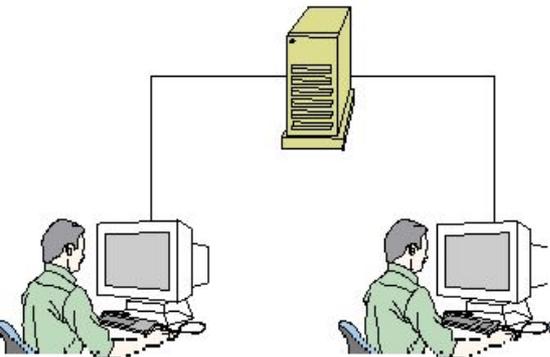
- Многозадачные
  - UNIX
  - OS/2
  - Windows NT
    - мультипрограммирование
- Однозадачные
  - MS-DOS
    - Отсутствуют защитные механизмы и коммуникационные возможности (между программами)





## Поддержка многопользовательского режима

- Однопользовательские
  - MS-DOS
  - DOS + Windows 3.x
- Многопользовательские
  - Windows NT
  - UNIX
    - Механизмы защиты персональных данных каждого пользователя





# Многопроцессорная обработка

- Два или более процессоров, *параллельно* выполняющих команды
  - Усложнение всех алгоритмов управления ресурсами
  - Linux, Solaris, Windows NT
  - Многопроцессорные ОС:
    - *Симметричные*
      - Обработка полностью децентрализована
    - *Асимметричные*
      - Процессоры неравноправны
        - » Master
        - » Slave





# Системы реального времени

- Многозадачные системы
  - Пакетные системы
  - Системы разделения времени
  - Системы реального времени
- Предназначены в первую очередь для управления техническими объектами (АЭС) и технологическими процессами
  - Характеризуются предельно допустимым временем реакции на внешнее событие
    - Нет виртуальной памяти
  - QNX





# Графический интерфейс пользователя

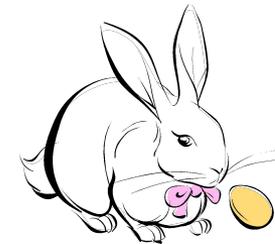
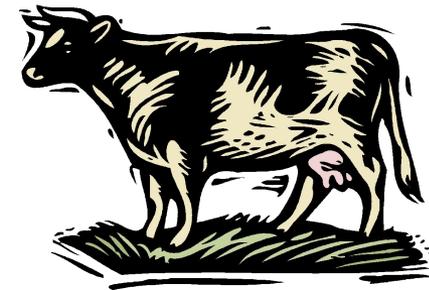
- Неграфические ОС
  - *Интерфейс командной строки* (клавиатура)
  - MS-DOS – использовалась более 15 лет!
  - Оболочки
  - Операционные оболочки
    - Многозадачность
    - Расширенные средства обмена данными между программами
    - Графический интерфейс пользователя
- Графические ОС
  - *Графический интерфейс пользователя* (манипулятор)
  - Windows'9x
  - Windows NT
  - Linux (на самом деле это и не совсем так)

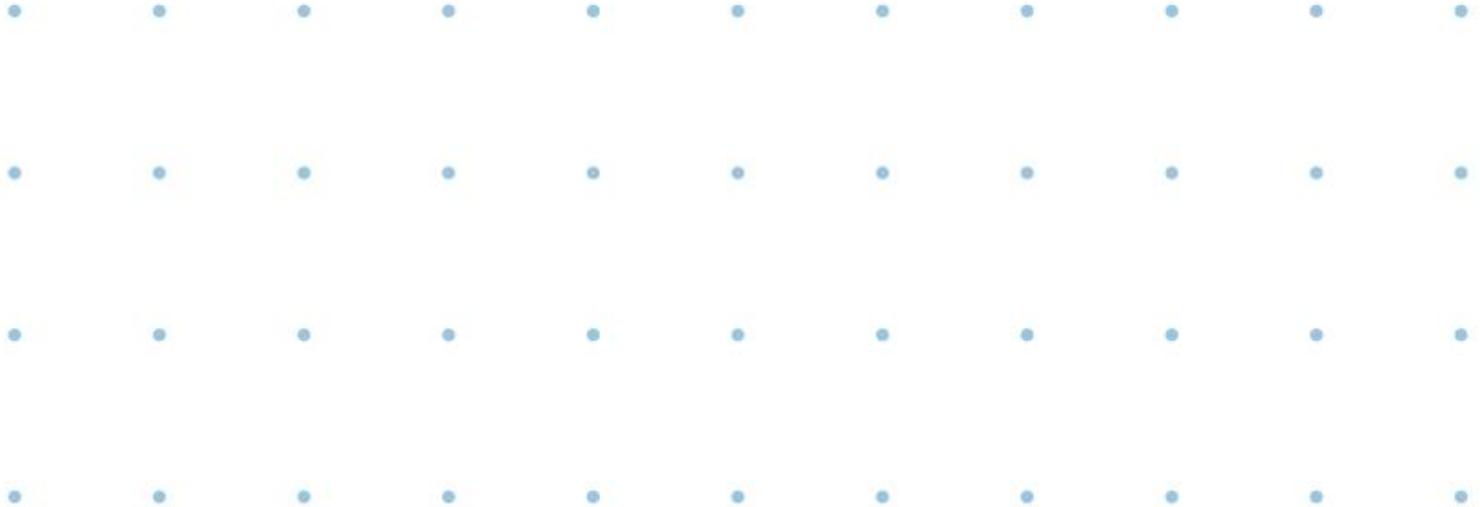




# Семейства ОС

- **CP/M**
  - Примитивная операционная среда
- **MS-DOS, PC-DOS, DR-DOS, PTS-DOS**
  - Иерархическая файловая система
  - Большое число внешних команд
- **OS/2**
  - Графический интерфейс пользователя
- **Windows'9x**
  - Псевдомногозадачные ОС
  - 32-х разрядные, но с сохранением 16-ти разрядной подсистемы
- **Windows NT, 2k, XP, 2003 Server, Longhorn**
  - Многопользовательские
  - Истинно многозадачные
  - Полностью 32-х разрядные
- **UNIX, XENIX, Linux**
  - Многопользовательские
  - Истинно многозадачные
  - Полностью 32-х разрядные
  - сверхзащищенные
- **QNX, WebOS, BeOS**
  - Нестандартные ОС
  - ОС реального времени
  - ...



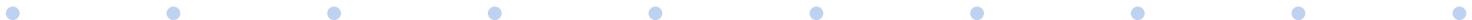


**Выводы**



# Выводы

- ОС обеспечивает *интерфейсы* (связь) между пользователем, программами и аппаратурой
- Вычислительные системы прошли в своей эволюции четыре этапа
- ОС оперируют такими *понятиями и концепциями*, как
  - Системные вызовы
  - Прерывания
  - Исключения
  - Файлы
  - Процессы и нити
- ОС может быть построена на основе:
  - Монолитного ядра
  - Слоеной архитектуры
  - Виртуальных машин
  - Микроядерной архитектуры
  - Смешанной архитектуры
- Существует несколько *классификаций* ОС





## Вопросы?

Возникли ли у Вас какие-нибудь вопросы после прослушивания этой лекции?





## Контрольные вопросы

- Из каких слоев состоит вычислительная система?
- Какие виды ПО Вы знаете?
- Каковы основные точки зрения на то, что такое ОС?
- Какие этапы (и сколько) прошли вычислительные системы своей эволюции?
- Перечислите основные концепции и понятия ОС?
- Какие архитектуры ОС Вы знаете?
- По каким критериям можно классифицировать ОС?
- Какие семейства ОС Вы знаете?



## Использованные материалы

- В.Е. Карпов, К.А. Коньков. Курс «Введение в операционные системы» // Общая редакция член-корр. РАН В.П. Иванников. Курс создан при поддержке корпорации Intel. ©2001-2002г.
- Симонович С.В., Евсеев Г.А., Алексеев А.Г. Общая информатика: учебное пособие для средней школы. – М.: АСТ-ПРЕСС, 1998.
- Фигурнов В.Э. IBM PC для пользователя. Изд. 5-е, исправл. и доп. – М.: Финансы и статистика, НПО «Информатика и компьютеры», 1994.

