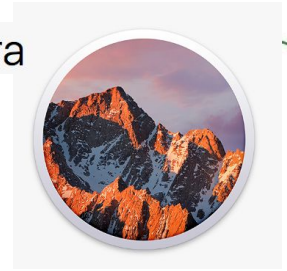
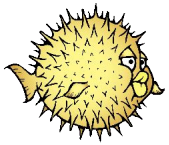
Кафедра Математической кибернетики и информационных технологий

«Операционные системы»

старший преподаватель **Королькова Татьяна Валерьевна**
+7(903)729-82-37 tvkorolkova@gmail.com



Литература

а) основная литература:

- С.В.Назаров, А.И.Широков. Современные операционные системы.- М.: Бином, 2013
- В.Г.Олифер, Н.А.Олифер. Сетевые операционные системы. – СПб.: Питер, 2009
- Э.Таненбаум, Х.Бос. Современные операционные системы. 4-изд. – СПб.: Питер, 2015

б) дополнительная литература:

- Марк Руссинович, Дэвид Соломон. Внутреннее устройство Microsoft Windows. . - СПб.: Питер, серия «Мастер-класс», 2013
- Марк Руссинович, Дэвид Соломон, А. Ионеску. Внутреннее устройство Microsoft Windows. Основные подсистемы ОС. - СПб.: Питер, серия «Мастер-класс», 2014
- Марк Руссинович, Аарон Маргозис. Утилиты Sysinternals. Справочник администратора. СПб.: БВХ-Петербург, 2012
- Петцольд Ч. Программирование для Microsoft Windows 8. 6-е изд. - СПб.: Питер, 2014
- С.В.Синицын, А.В.Батаев, Н.Ю.Налютин. Операционные системы.-М: Академия, 2012
- Джеффри Рихтер, Кристоф Назар. Windows via C/C++. – СПб.: Питер, «Русская редакция», 2009
- Джеффри Рихтер. Windows для профессионалов. Создание эффективных Win32-приложений с учетом специфики 64-разрядной версии Windows. - СПб.: Питер, «Русская редакция», 2001
- Х.М.Дейтел, П.Дж.Дейтел, Д.Р. Чофес. Операционные системы.- М.: Изд. Бином, 2009
- Д.В.Иртегов. Введение в операционные системы. СПб.: БВХ-Петербург, 2012
- С.В.Назаров. Операционные среды, системы и оболочки. Основы структурной и функциональной организации: Учебное пособие. – М.: КУДИЦ-ПРЕСС, 2007
- Сафонов В.О. Основы современные операционных систем – М.: Национальный Открытый Университет «ИНТУИТ»: БИНОМ. Лаборатория знаний, 2011
- Мартимьянов Ю.Ф., Яковлев Ал.В., Яковлев Ан..В. Операционные системы. Концепции построения и обеспечения безопасности. – М.: Горячая линия – Телеком, 2011
- С. В. Назаров, Л. П. Гудыно, А. А. Кириченко. Операционные системы. Практикум для

Раздел 1

Основные концепции операционных систем

Лекция 1

НАЗНАЧЕНИЕ И ФУНКЦИИ ОПЕРАЦИОННЫХ СИСТЕМ

Сферы применения операционных систем

- ✔ ОС мейнфреймов
- ✔ Серверные ОС
- ✔ ОС персональных компьютеров
- ✔ ОС смартфонов
- ✔ Встроенные ОС
- ✔ ОС сенсорных узлов
- ✔ ОС смарт-карт

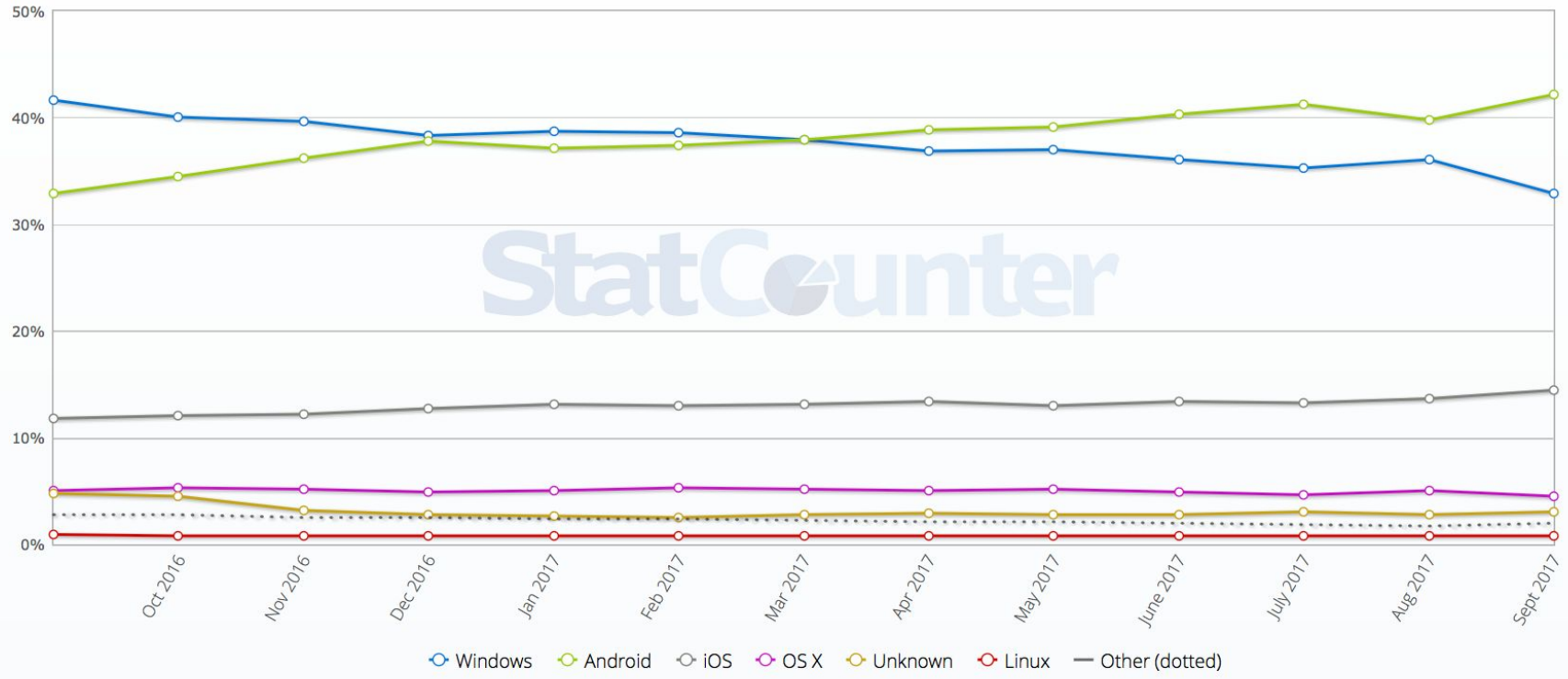
Android	Windows	iOS	OS X	Unknown	Linux
39.82%	36.1%	13.68%	5.1%	2.72%	0.8%

August 2017

Operating System Market Share Worldwide

Sept 2016 - Sept 2017

Edit Chart Data



StatCounter GlobalStats

Press Releases FAQ About Feedback

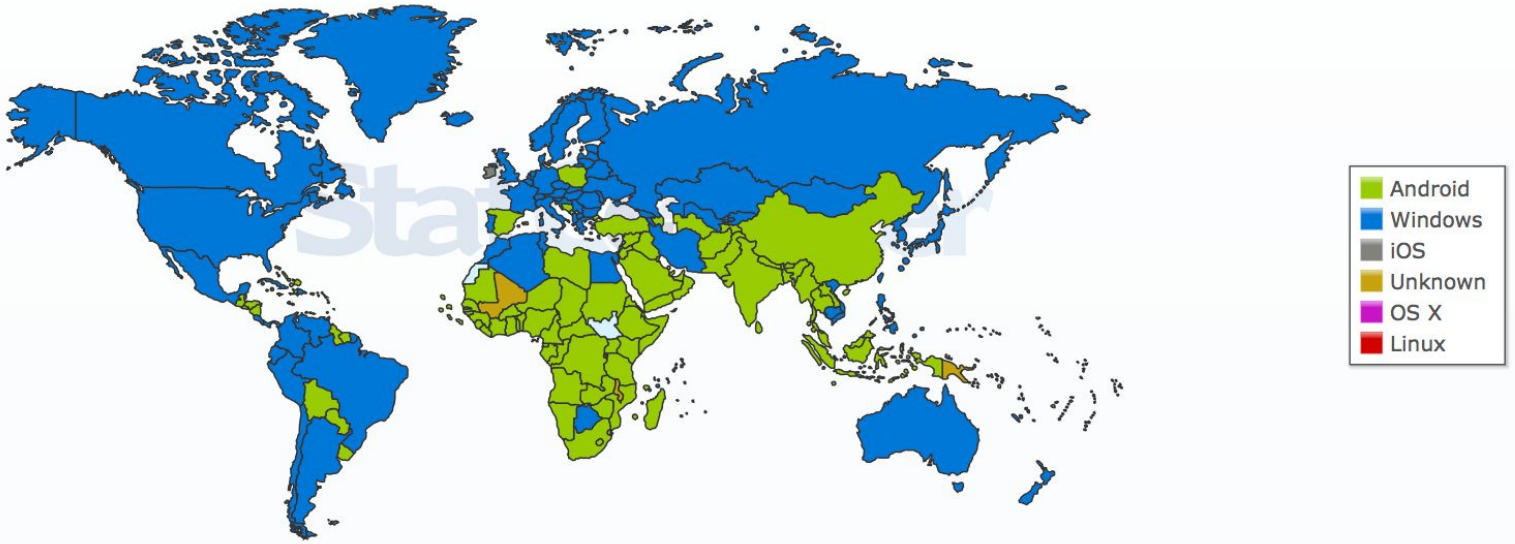
Android	Windows	iOS	OS X	Unknown	Linux
39.82%	36.1%	13.68%	5.1%	2.72%	0.8%

August 2017

Operating System Market Share Worldwide

Aug - Sept 2017

Edit Chart Data



Save Chart Image (.png) Download Data (.csv) Embed HTML

<div id="all-os_combined-ww-monthly-201708-201709" width="600" he

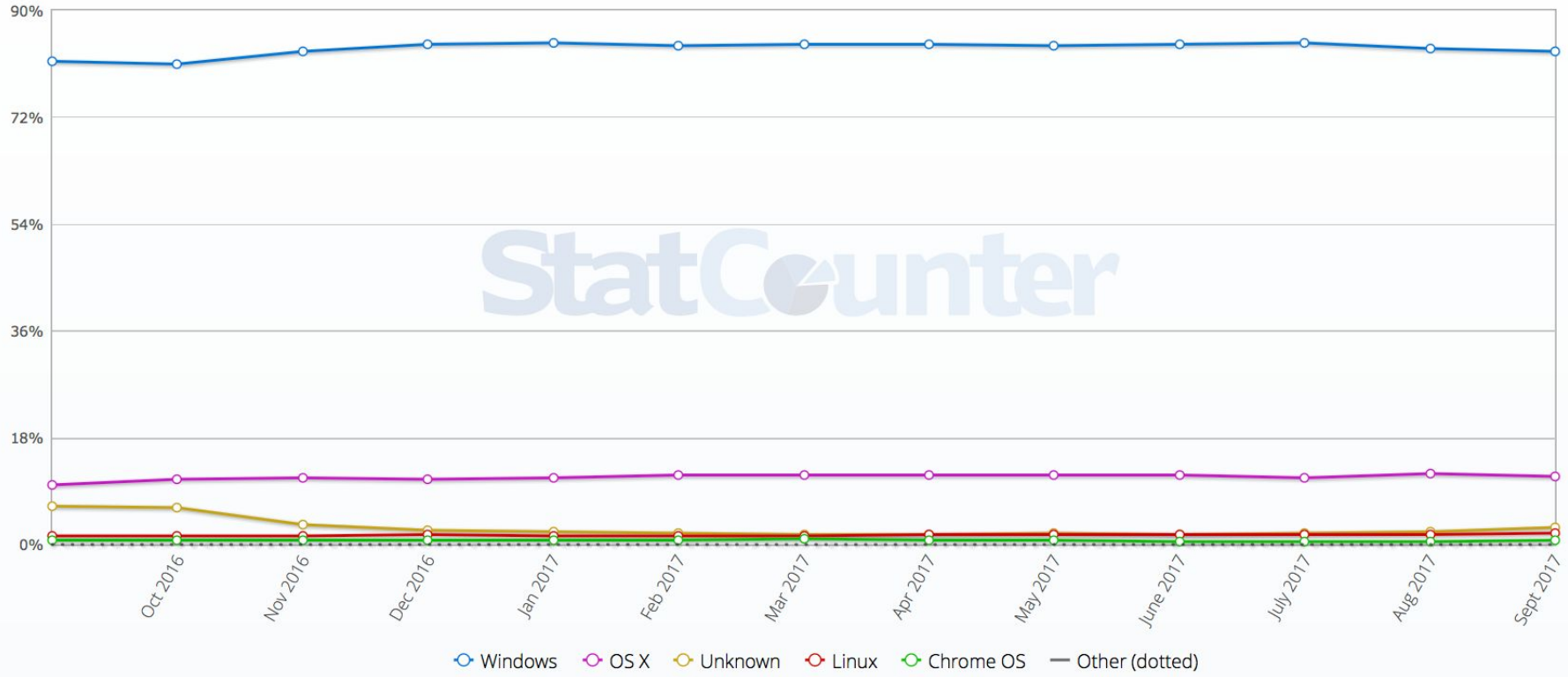
Windows	OS X	Unknown	Linux	Chrome OS	FreeBSD
83.53%	11.95%	2.15%	1.79%	0.56%	0%

August 2017

Desktop Operating System Market Share Worldwide

Sept 2016 - Sept 2017

Edit Chart Data



StatCounter GlobalStats

Press Releases FAQ About Feedback

Windows	OS X	Unknown	Linux	Chrome OS	FreeBSD
83.53%	11.95%	2.15%	1.79%	0.56%	0%

August 2017

Desktop Operating System Market Share Worldwide

Aug - Sept 2017

Edit Chart Data

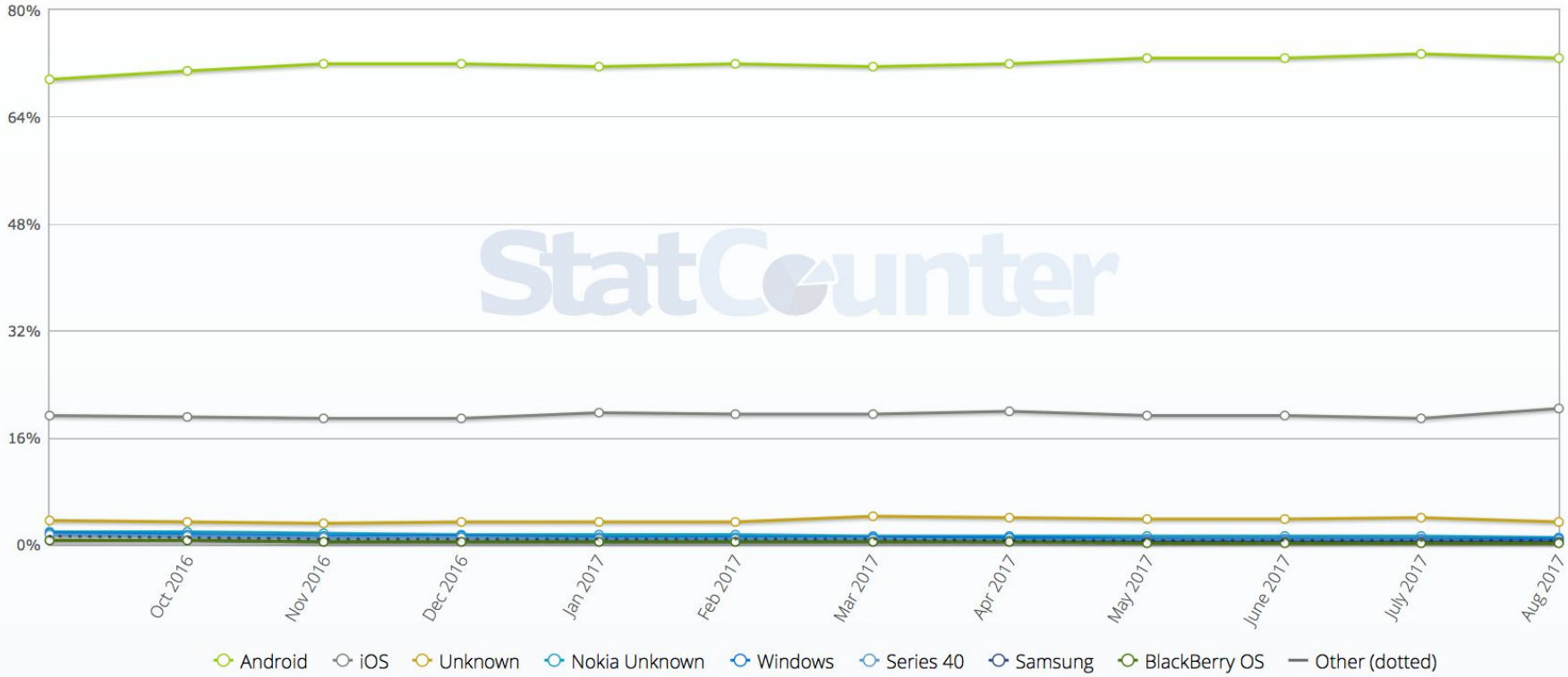
Android	iOS	Unknown	Nokia Unknown	Windows	Series 40
72.76%	20.29%	3.43%	1.06%	0.82%	0.51%

August 2017

Mobile Operating System Market Share Worldwide

Sept 2016 - Aug 2017

Edit Chart Data



StatCounter GlobalStats

Press Releases FAQ About Feedback

Android	iOS	Unknown	Nokia Unknown	Windows	Series 40
72.76%	20.29%	3.43%	1.06%	0.82%	0.51%

August 2017

Mobile Operating System Market Share Worldwide

Aug 2017

Edit Chart Data

Определение операционной системы

«Я не знаю, что это такое, но всегда узнаю ее, если увижу...»

Barron D.W.,
Computer Operating Systems, 1971 год

ОС компьютера – это комплекс взаимосвязанных программ, обеспечивающий взаимодействие пользователя с вычислительной системой, а также управляющий ресурсами вычислительной системы.

Ресурс - всякий объект, который может распределяться внутри вычислительной системы.

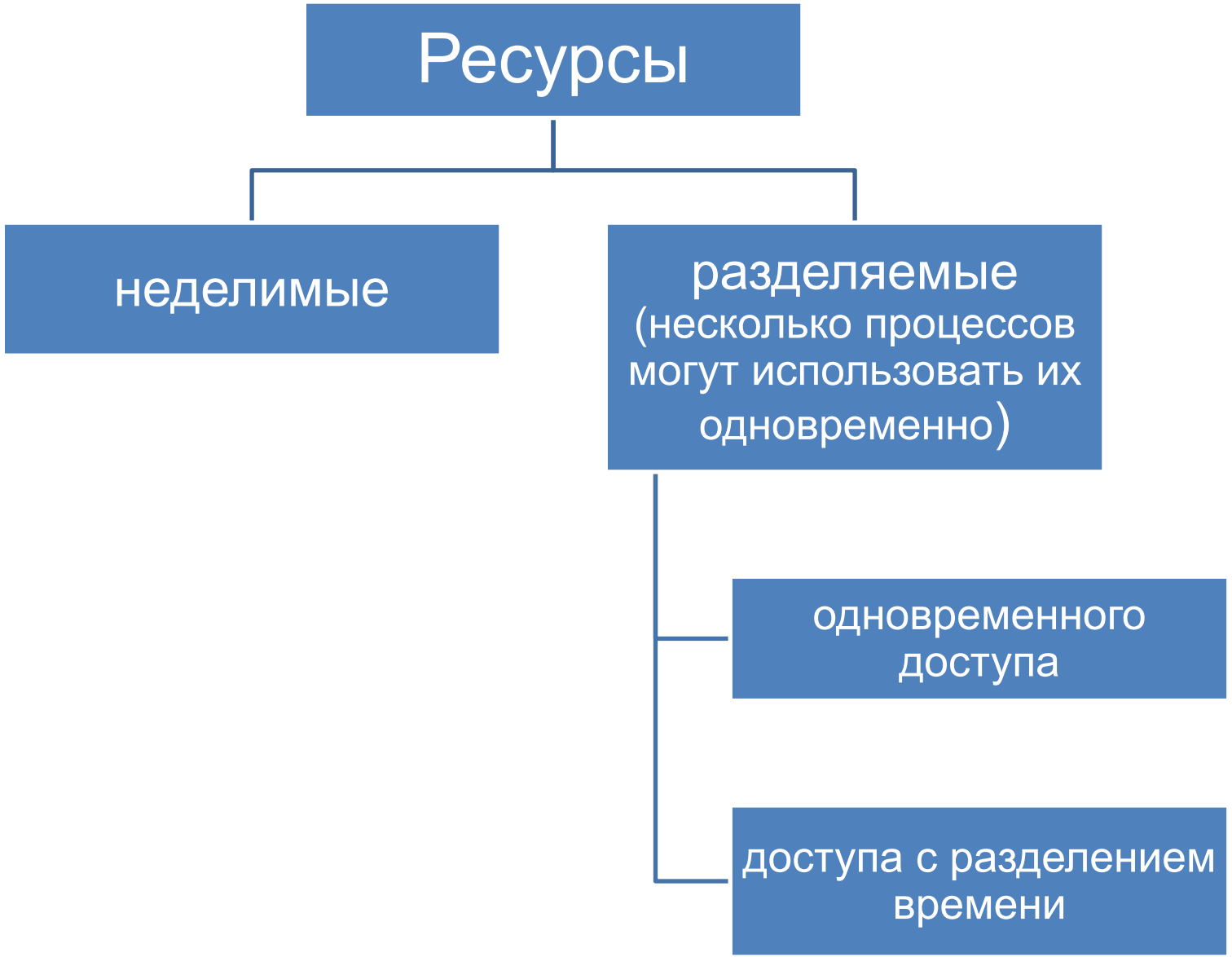
Основные ресурсы вычислительной системы:

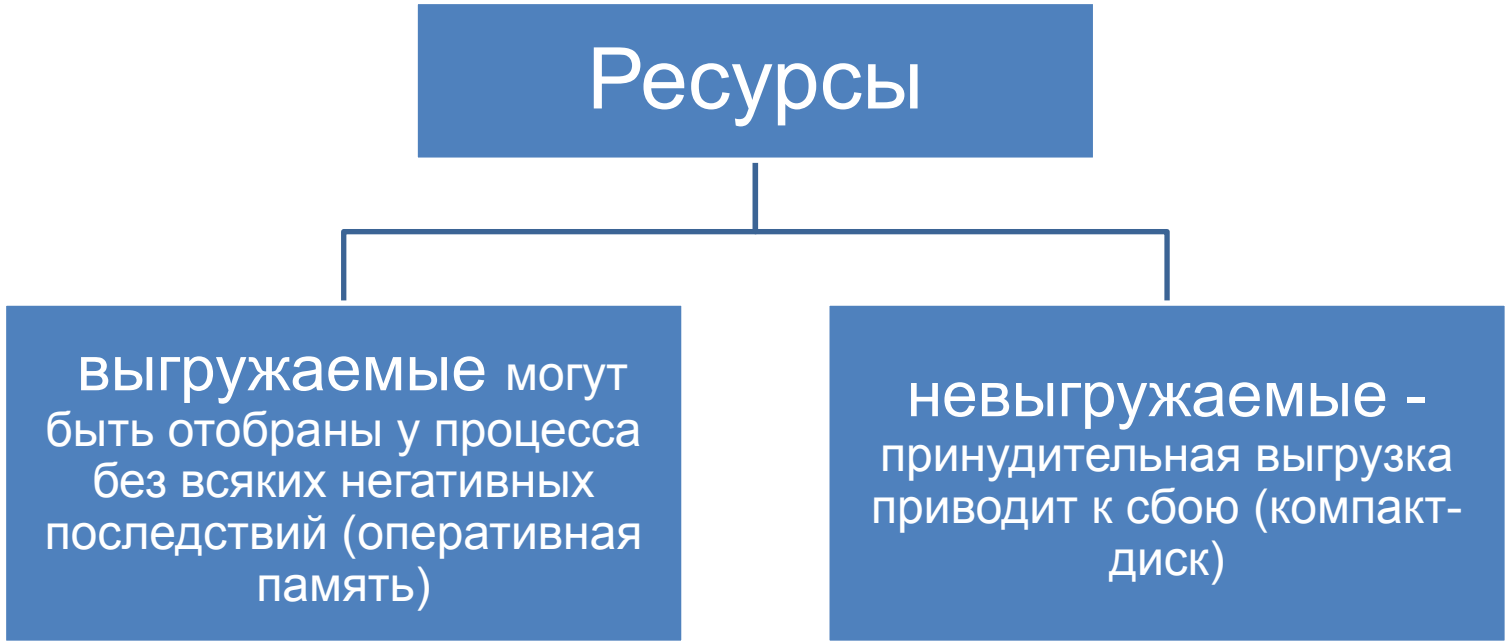
Физические:

- процессоры (процессорное время);
- память (оперативная, внешняя);
- периферийные устройства (диски, таймеры, наборы данных, принтеры, сетевые устройства и др.)

Логические

- существуют только в пределах самой ОС: таблицы выполняемых процессов, сетевых подключение и т.д.)

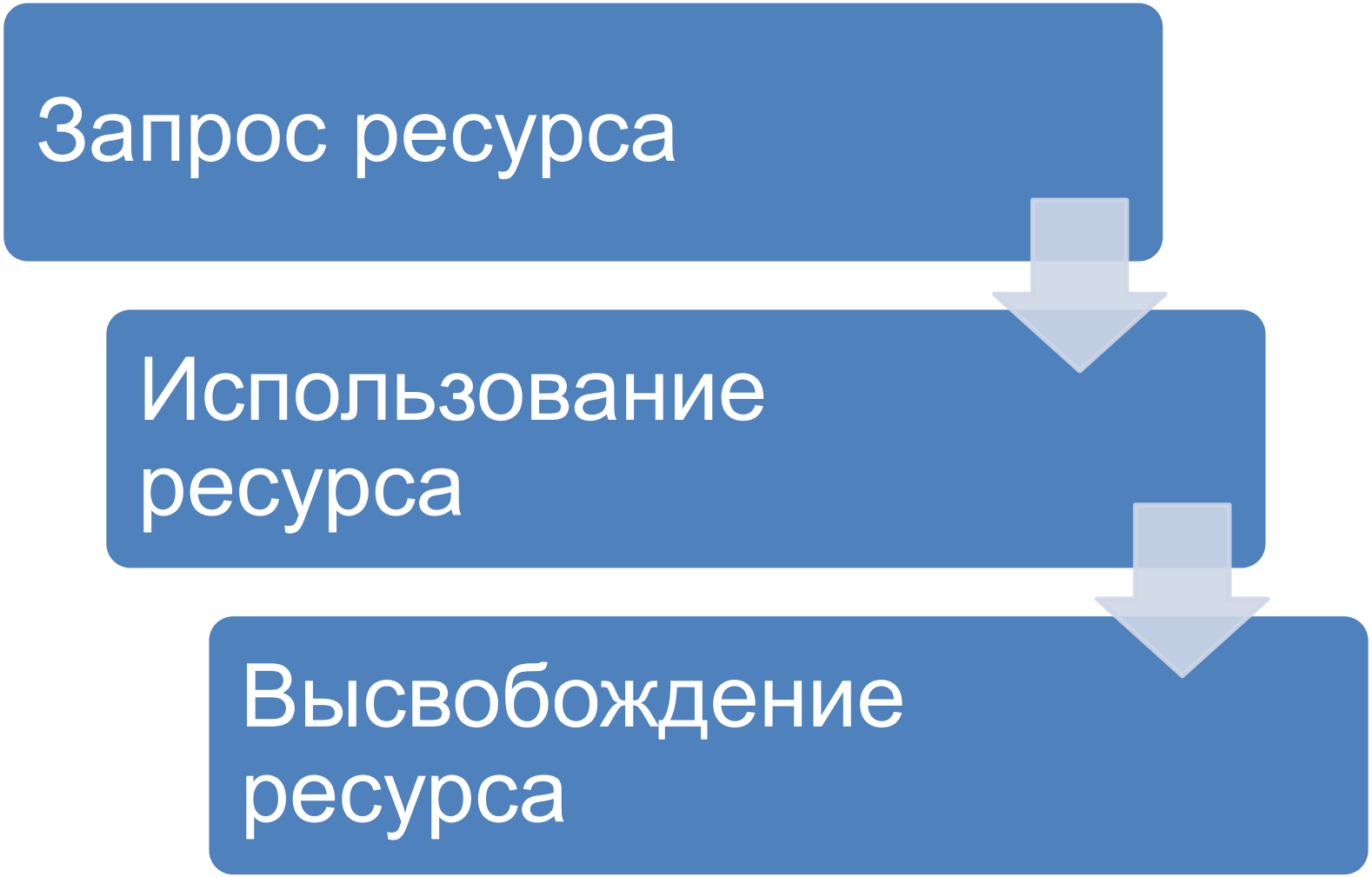

Ресурсы
неделимые
разделяемые (несколько процессов могут использовать их одновременно)
одновременного доступа
доступа с разделением времени


Ресурсы
выгружаемые могут быть отобраны у процесса без всяких негативных последствий (оперативная память)
невыгружаемые - принудительная выгрузка приводит к сбою (компакт-диск)

Управление ресурсами

- Планирование ресурса
- Диспетчеризация ресурса
- Отслеживание состояния и учет использования ресурса
- Разрешение конфликтов между процессами

Цикл использования ресурса:

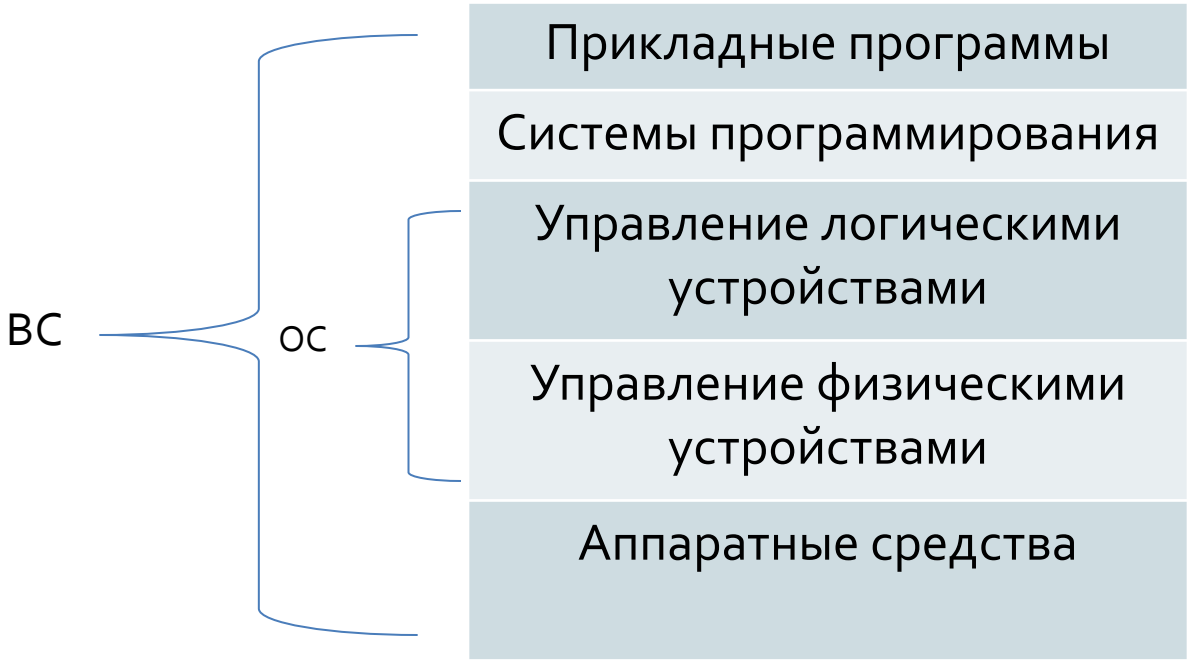


Место ОС в структуре вычислительной системы

Вычислительная система – программно-аппаратный комплекс

предоставляющий услуги пользователю.

Структура вычислительной системы



Аппаратный уровень

Физические ресурсы: процессор, оперативная память, внешние устройства.

Характеристики:

- правила программного использования, которые определяют возможность корректного использования в программе;
- производительность или емкость: тактовая частота, длина обрабатываемого машинного слова;
- степень занятости или используемости данного физического ресурса.

Средства программирования, доступные на аппаратном уровне:

- система команд компьютера;
- аппаратные интерфейсы программного взаимодействия с физическими ресурсами.

Уровень управления физическими устройствами

Первый уровень системного программного обеспечения. Назначение – систематизация и стандартизация правил программного использования физических ресурсов.

Для управления физическими ресурсами используются программы, которые называются драйверами физического ресурса (устройства).

Драйвер физического устройства – программа, основанная на использовании команд управления конкретного физического устройства и предназначенная для организации работы с данным устройством

Уровень управления логическими ресурсами

Логическое устройство (ресурс) – устройство, некоторые эксплуатационные характеристики которого (возможно, все) реализованы программным образом.

Этот уровень ориентирован на пользователя. Команды данного уровня не зависят от физических устройств, они обращены к предыдущему уровню.

Уровень систем программирования

Система программирования – это комплекс программ, обеспечивающий поддержание жизненного цикла программы в вычислительной системе.

Уровень прикладных систем

Прикладная система – программная система, ориентированная на решение или автоматизацию решения задач из конкретной предметной области.

ОС компьютера – это комплекс взаимосвязанных программ, обеспечивающий взаимодействие пользователя с вычислительной системой, а также управляющий ресурсами вычислительной системы.

Назначение и функции ОС

Операционная система - это набор программ, реализующих интерфейсы

1.Предоставление пользователю вместо реальной аппаратуры компьютера расширенной виртуальной машины

Операционная система превращает уродливое аппаратное обеспечение в красивые абстракции

ПРИКЛАДНЫЕ ПРОГРАММЫ

КРАСИВЫЙ ИНТЕРФЕЙС

ОПЕРАЦИОННАЯ СИСТЕМА

УРОДЛИВЫЙ ИНТЕРФЕЙС

АППАРАТНОЕ ОБЕСПЕЧЕНИЕ

2. Повышение эффективности использования компьютера путем рационального управления его ресурсами в соответствии с некоторым критерием

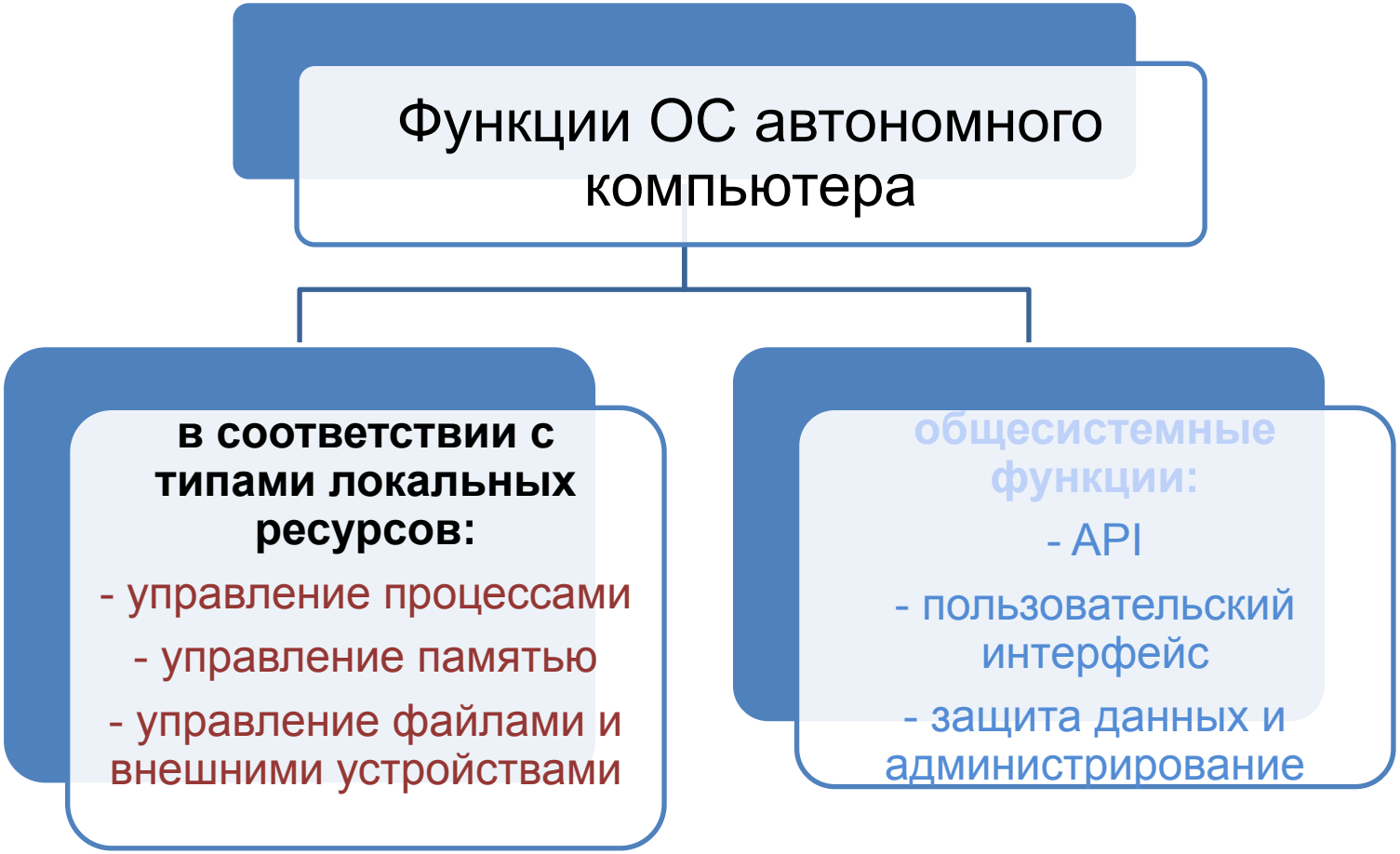
Критерии эффективности и классы операционных систем

Класс ОС	Критерий эффективности
ОС пакетной обработки	Максимальная пропускная способность (максимальная загрузка процессора)
Интерактивные ОС (ОС разделения времени)	Удобство работы пользователя
ОС реального времени	Реактивность (гарантированное время реакции системы на то или иное событие)

Лекция 2

ФУНКЦИОНАЛЬНЫЕ КОМПОНЕНТЫ ОПЕРАЦИОННОЙ СИСТЕМЫ АВТОНОМНОГО КОМПЬЮТЕРА

Функции операционных систем



Функциональные компоненты операционной системы автономного компьютера

1. Подсистема управления процессами
2. Подсистема управления памятью
3. Подсистема управления файлами и внешними устройствами
4. Подсистема защиты данных и администрирования
5. Интерфейс прикладного программирования
6. Пользовательский интерфейс
7. Поддержка сетей

Подсистема управления процессами

1. • распределяет процессорное время между несколькими одновременно выполняющимися в системе процессами
2. • создает и уничтожает процессы
3. • обеспечивает процессы необходимыми ресурсами
4. • реализует обмен данными между процессами
5. • поддерживает синхронизацию процессов

Подсистема управления памятью

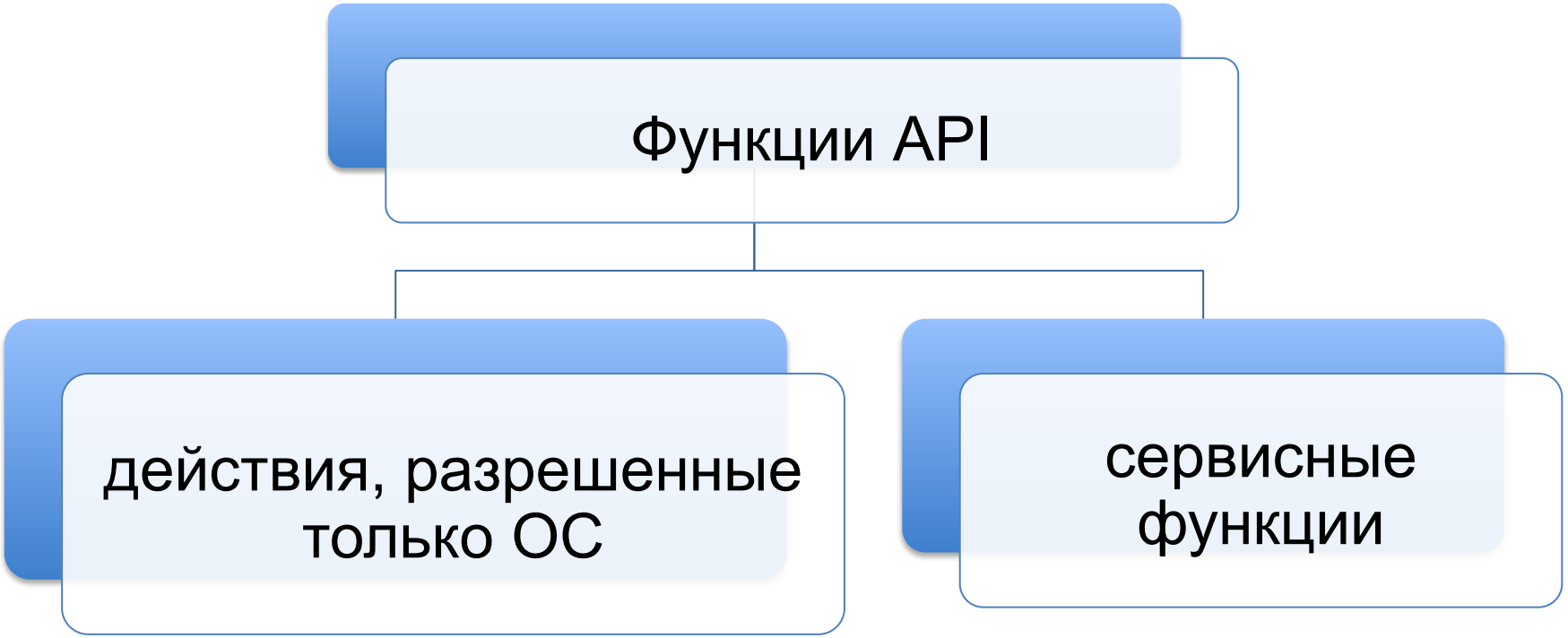
1. отслеживает свободную и занятую память
2. выделяет память процессам и освобождает память по завершении процессов
3. загружает коды и данные процессов в отведенную память
4. организует виртуальную память
5. настраивает адреса программы на конкретную область физической памяти
6. динамически распределяет память
7. выполняет дефрагментацию памяти
8. реализует защиту памяти

Подсистема управления файлами и внешними устройствами

1. организует параллельную работу устройств ввода-вывода и процессора
2. осуществляет согласование скоростей обмена и кэширование данных
3. разделяет устройства и данные между процессами
4. организует удобный интерфейс между устройствами и остальной частью системы
5. поддерживает широкий спектр драйверов с возможностью простого включения в систему нового драйвера
6. динамически загружает и выгружает драйверы
7. поддерживает несколько файловых систем
8. поддерживает синхронные и асинхронные операции ввода-вывода

Интерфейс прикладного программирования

Возможности ОС доступны прикладному программисту виде набора функций, называющегося интерфейсом прикладного программирования (Application Programming Interface, API).



- Для разработчика приложений все особенности конкретной ОС представлены особенностями ее API. Поэтому разные ОС с одинаковым набором API, кажутся им одной и той же ОС. Это упрощает стандартизацию ОС. Например, стандартом API UNIX является стандарт Posix.

- Приложения выполняют обращения к функциям API с помощью системных вызовов.

Системные вызовы

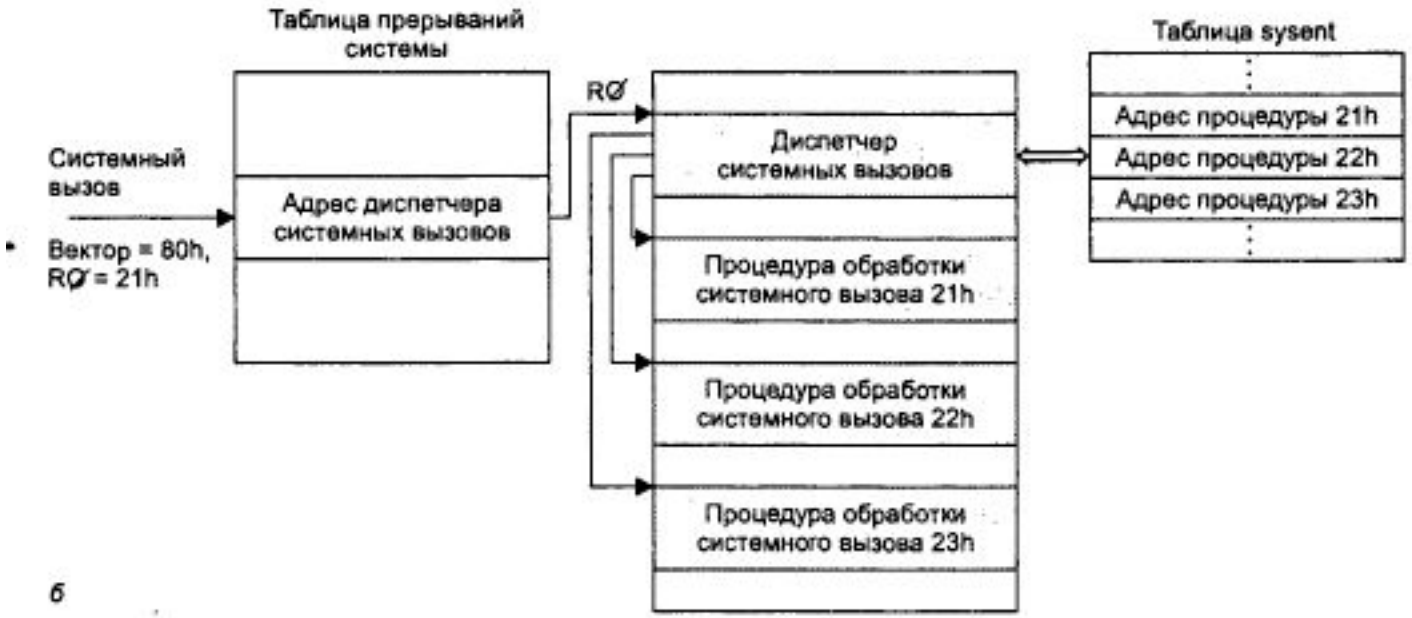
Реализация системных вызовов должна обеспечивать:

- ✔ переключение в привилегированный режим;
- ✔ высокую скорость вызова процедур ОС;
- ✔ по возможности единообразное обращение к системным вызовам для всех аппаратных платформ, для которых предназначена ОС;
- ✔ возможность расширения набора системных вызовов;
- ✔ контроль со стороны ОС использования системных вызовов

Реализуются с помощью механизма программных прерываний.

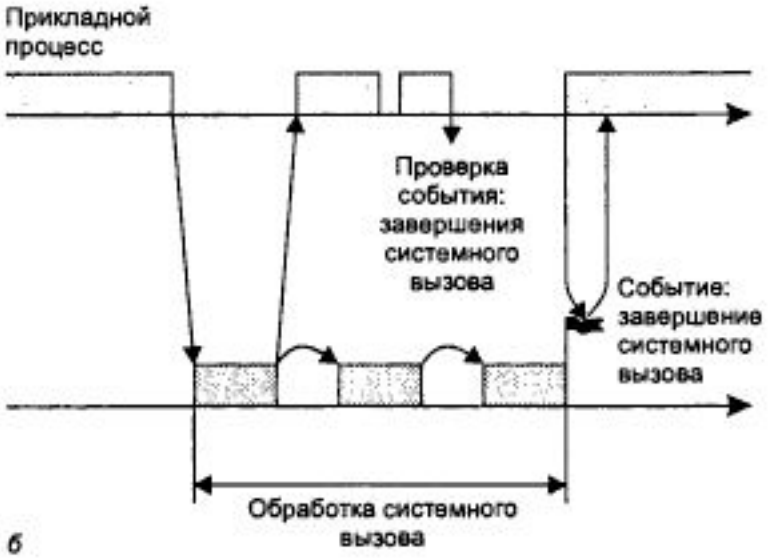
Схема обработки системных вызовов

- децентрализованная
- централизованная



Системные вызовы

- синхронные
- асинхронные

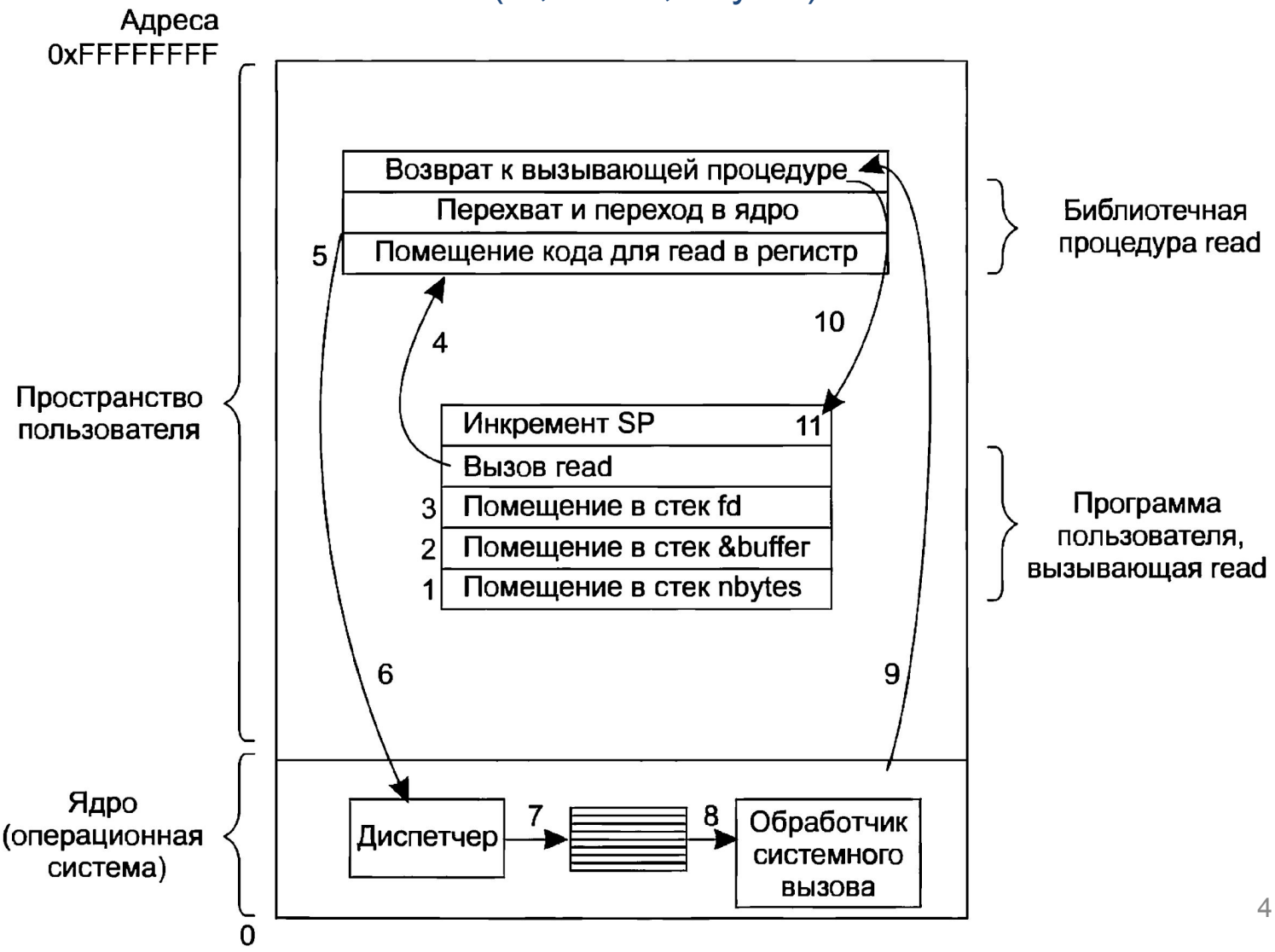


Приблизительное соответствие системных вызовов Win32 API вызовам Unix

UNIX	Win32	Описание
fork	CreateProcess	Создает новый процесс
waitpid	WaitForSingleObject	Ожидает завершения процесса
execve	(нет)	CreateProcess=fork+execve
exit	ExitProcess	Завершает выполнение процесса
open	CreateFile	Создает файл или открывает существующий файл
close	CloseHandle	Закрывает файл
read	ReadFile	Читает данные из файла
write	WriteFile	Записывает данные в файл
lseek	SetFilePointer	Перемещает указатель файла
stat	GetFileAttributesEx	Получает различные атрибуты файла
mkdir	CreateDirectory	Создает новый каталог
rmdir	RemoveDirectory	Удаляет пустой каталог
link	(нет)	Win32 не поддерживает связи
unlink	DeleteFile	Удаляет существующий файл
mount	(нет)	Win32 не поддерживает подключение к файловой системе
umount	(нет)	Win32 не поддерживает подключение к файловой системе
chdir	SetCurrentDirectory	Изменяет рабочий каталог

11 этапов выполнения системного вызова read(fd, buffer, nbytes)

count=read(fd, buffer, nbytes)



Защита данных и администрирование

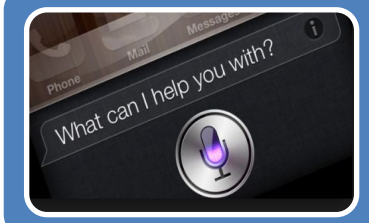
- защита от сбоев и отказов аппаратуры (реализуется путем резервирования) и ошибок программного обеспечения (например, самовосстанавливающиеся файловые системы)

- защита от несанкционированного доступа (процедура логического входа, аутентификация, авторизация, администрирование, аудит)

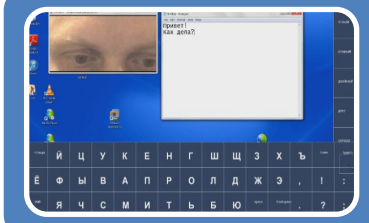
Пользовательский интерфейс

алфавитно-цифровой (командный) интерфейс

графический (WIMP: Window, Image, Menu, Pointer) интерфейс



голосовой (SILK: Speech, Image, Language, Knowledge) интерфейс



биометрическая технология

Лекция 3

АРХИТЕКТУРА ОПЕРАЦИОННЫХ СИСТЕМ

Требования, предъявляемые к современным ОС

Главное требование - выполнение основных функций эффективного управления ресурсами и обеспечение удобного интерфейса для пользователя и прикладных программ.

Традиционные требования к ОС:

1. **прозрачность (незаметность) работы служебных программ**
2. **гарантированная надежность**
3. **максимальная скорость выполнения**
4. **минимальный машинный код**
5. **использование стандартных средств для связи с проблемными программами**

Сравнение количества строк кода в некоторых модулях ядра Linux и Windows

Функции ядра	Linux	Vista
Планировщик CPU	50 000	75 000
Инфраструктура ввода-вывода	45 000	60 000
Виртуальная память	25 000	175 000

Год	Версия	Строк кода
1994	Windows NT 3.5	4 000 000
1996	Windows NT 4	16 500 000
2000	Windows 2000	20 000 000
2002	Windows XP	45 000 000

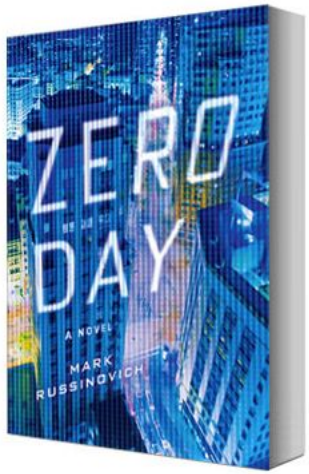
Год	Версия	Строк кода
1991	Ядро Linux 0.1	10 239
1994	Ядро Linux 1.0.0	176 250
1995	Ядро Linux 1.2.0	310 950
1996	Ядро Linux 2.0.0	777 956
1999	Ядро Linux 2.2.0	1 800 847
2001	Ядро Linux 2.4.0	3 377 902
2003	Ядро Linux 2.6.0	5 929 913
2009	Ядро Linux 2.6.32	12 606 910[5]
2017	Ядро Linux 4.11.7	18 373 471[6]

Распухание ОС так же неотвратимо, как смерть и налоги

Уязвимость нулевого дня

Zero-day эксплойт — киберугроза, использующая ошибку или уязвимость в приложении или операционной системе и появившаяся сразу после обнаружения данной уязвимости, пока разработчики ПО еще не успели создать патч, а IT-администраторы — принять другие меры безопасности.

Происхождение термина связано с тем обстоятельством, что уязвимость или атака становится публично известна до момента выпуска производителем ПО исправлений ошибки (то есть потенциально уязвимость может эксплуатироваться на работающих копиях приложения без возможности защититься от неё).



Расширяемость

Код должен быть написан так, чтобы систему можно было легко наращивать и модифицировать по мере изменения потребностей рынка.

Переносимость

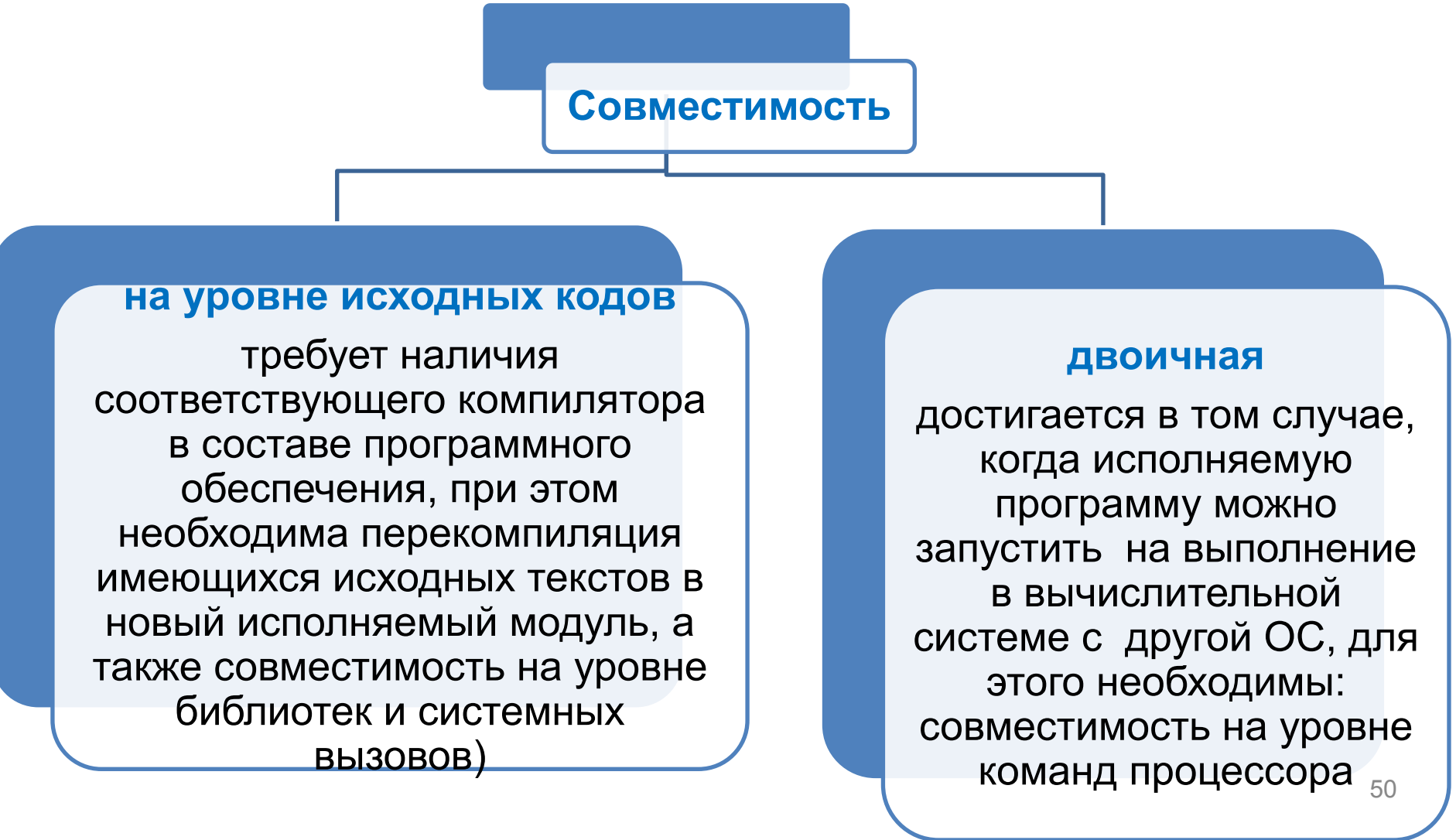
Переносимость (многоплатформенность) дает возможность перемещать всю систему на машину, базирующуюся на другом процессоре или аппаратной платформе, делая при этом по возможности минимальные изменения в коде.

Для написания переносимой ОС необходимо:

1. использовать язык высокого уровня
2. изолировать процессор
3. изолировать платформу

Совместимость

- способность ОС выполнять программы, написанные для других ОС или для более ранних версий данной операционной системы, а также для другой аппаратной платформы.



При совпадении архитектур процессоров (набора команд и диапазона адресов) двоичная совместимость достигается при:

1. поддержке вызовов API-функций новой ОС
2. соответствии внутренней структуры исполняемого файла правилам новой ОС

Для достижения двоичной совместимости в случае различных архитектур, кроме этих мер, необходимы :

1. эмуляция
2. использование множественных прикладных программных сред
3. система виртуальных машин (СВМ)

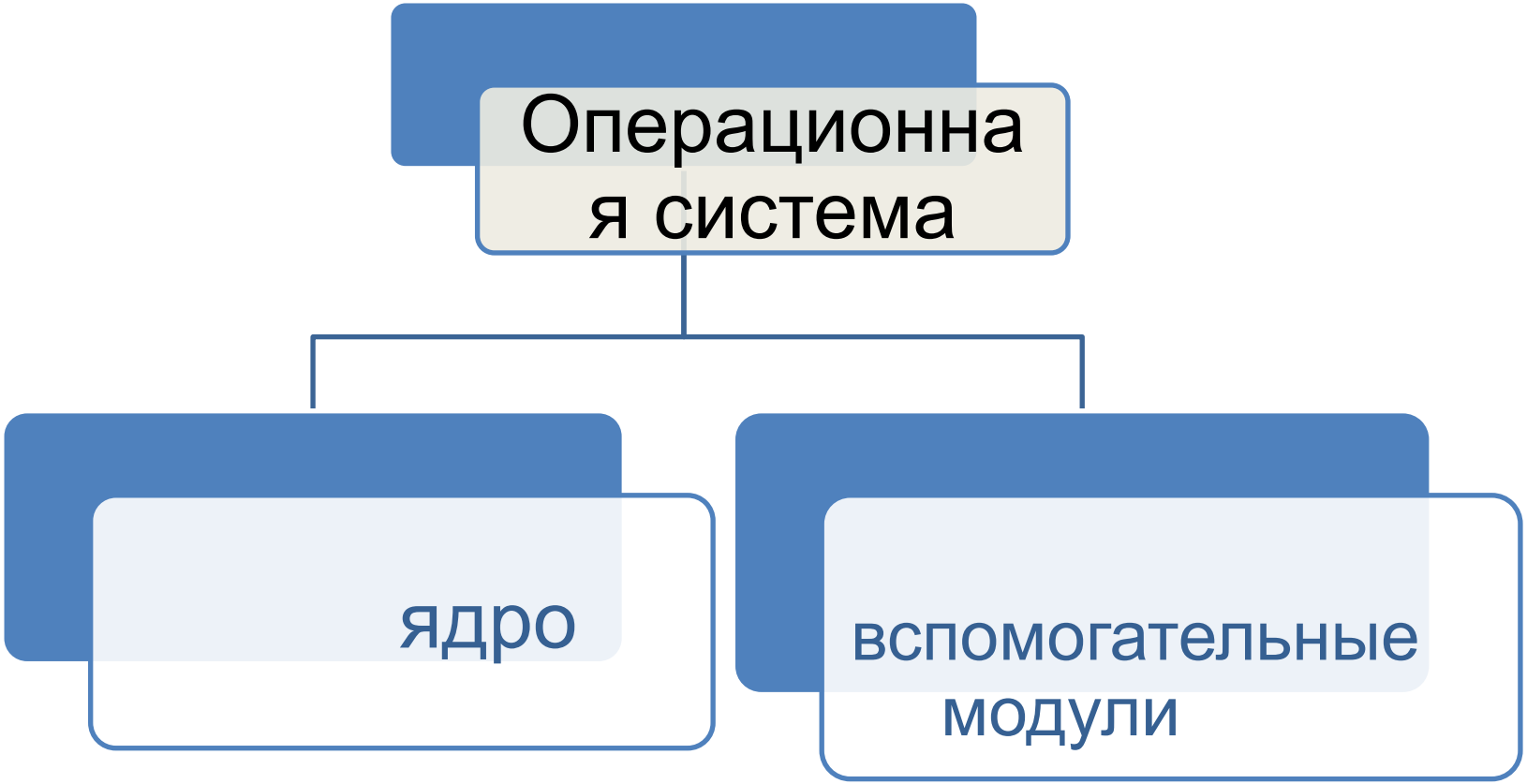
Система виртуальных машин (СВМ)

– такой вариант организации вычислительного процесса, при котором на одном компьютере одновременно выполняются несколько копий одной и той же или нескольких разных ОС. Каждая из этих ОС функционирует так же, как если бы она выполнялась на отдельном компьютере.

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		

Архитектура ОС



Ядро выполняет:

базовые функции ОС:

- управление процессами, памятью, устройствами ввода/вывода;

функции, решающие внутрисистемные задачи организации вычислительного процесса:

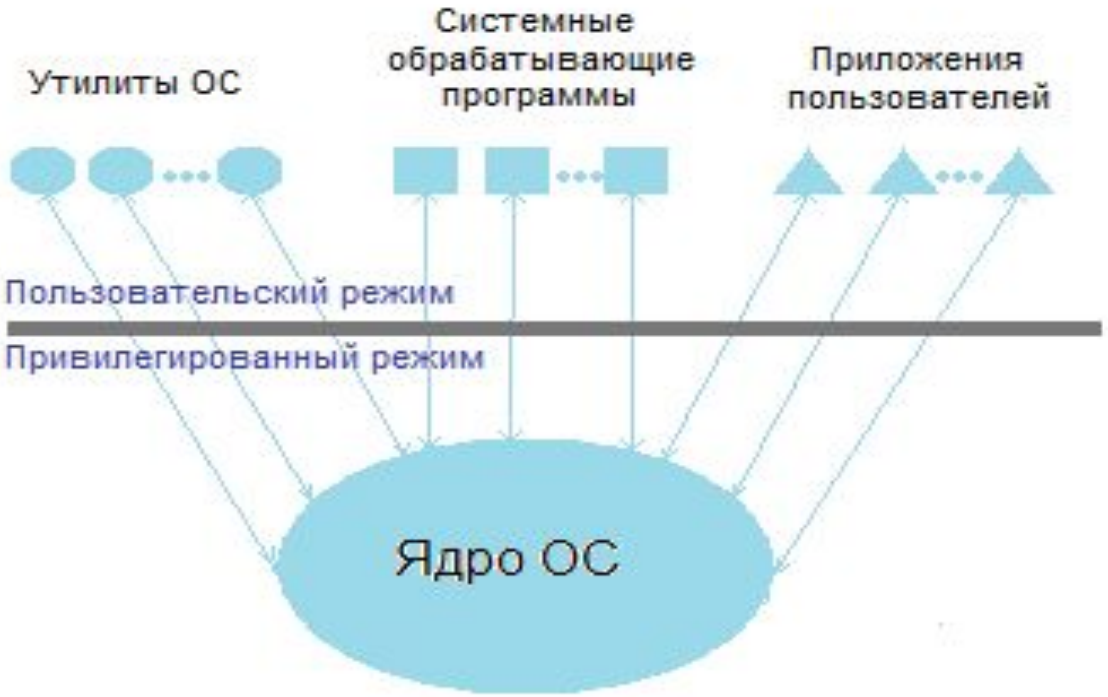
- переключение контекстов, загрузка/выгрузка страниц, обработка прерываний).

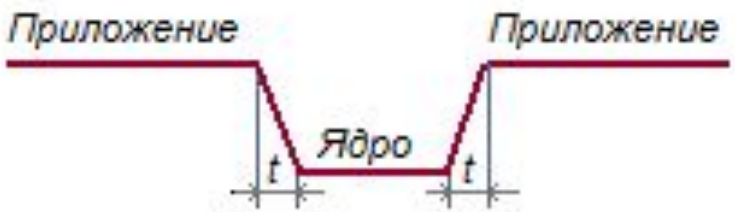
функции, создающие прикладную программную среду

Вспомогательные модули:

утилиты	• программы, решающие отдельные задачи управления и сопровождения компьютерной системы (дисковые утилиты: дефрагментаторы, очистка диска, разметка диска, программы сжатия, резервное копирование; утилиты работы с реестром; утилиты мониторинга оборудования и т.д.)
системные обрабатывающие программы	• текстовые и графические редакторы, компиляторы, компоновщики, отладчики
программы дополнительных услуг	• специальный вариант пользовательского интерфейса калькулятор, игры;
библиотеки процедур	• библиотеки математических функций, функций ввода/вывода и т.д.

Классическая архитектура

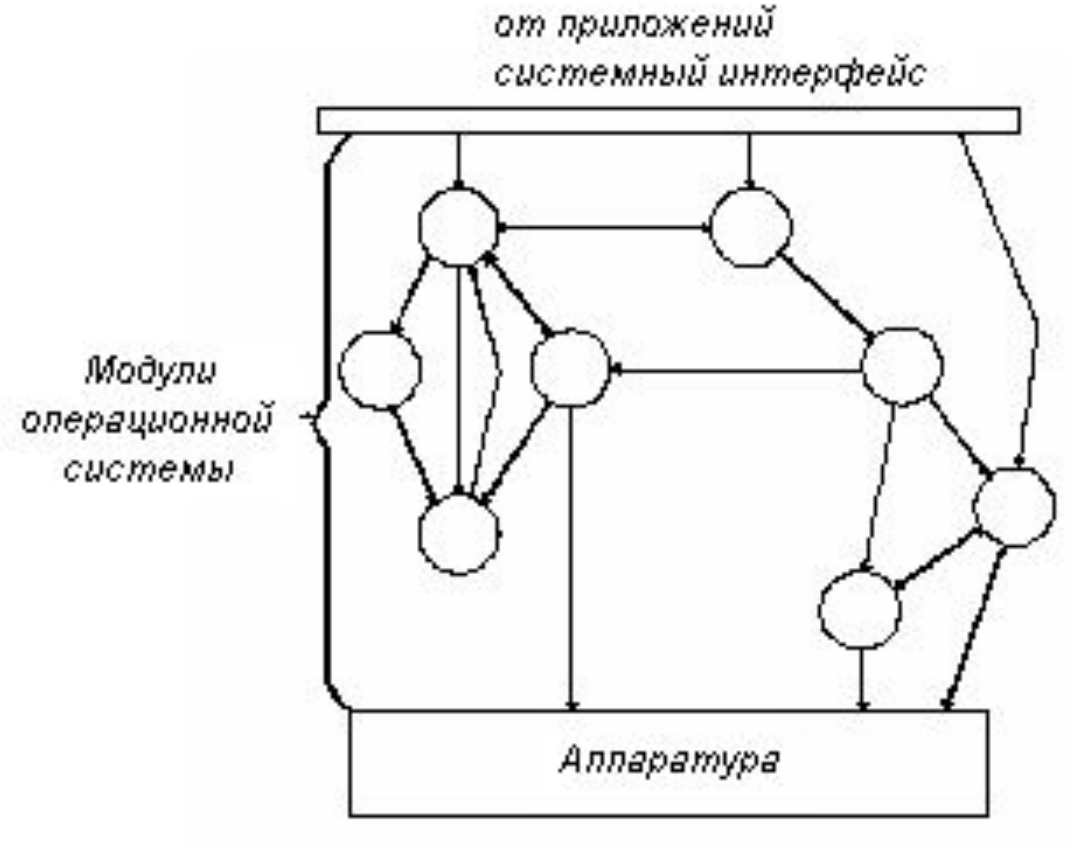




$$t_{з\Sigma} = 2t$$

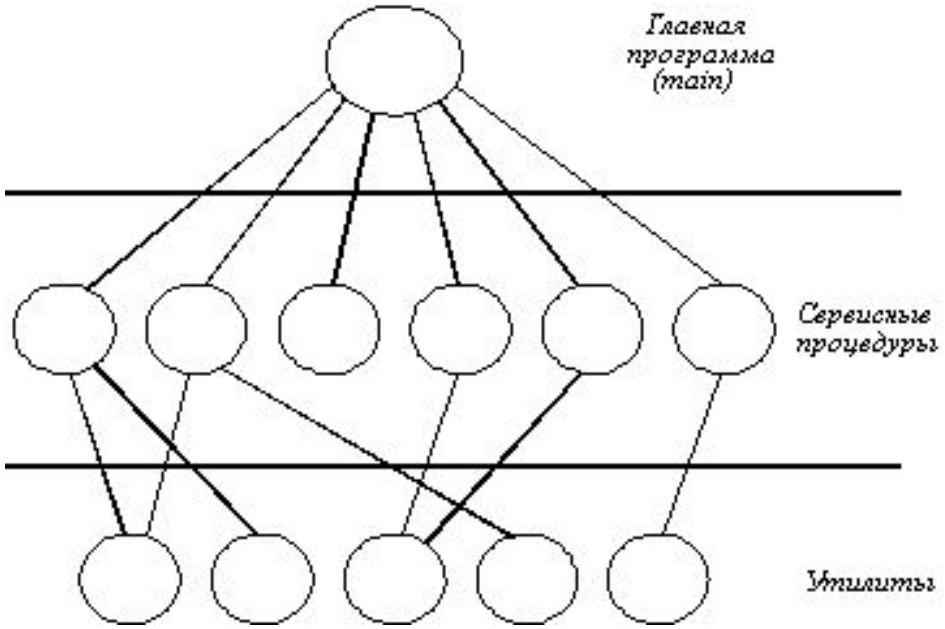
Монолитная структура ОС

Вся ОС работает как единая программа в режиме ядра. Монолитная ОС написана как набор процедур, связанных в одну большую исполняемую программу. Каждая процедура имеет возможность при необходимости вызвать другую.



Монолитные системы могут быть структурированными:

1. Главная программа, которая вызывает требуемые сервисные процедуры.
2. Набор сервисных процедур, реализующих системные вызовы.
3. Набор утилит, обслуживающих сервисные процедуры.



Многоуровневые системы

Обобщением предыдущего подхода является организация ОС как иерархии уровней с хорошо определенными связями между ними, так чтобы объекты уровня N могли вызывать только объекты из уровня N-1. Нижним уровнем в таких системах обычно является аппаратура, верхним уровнем интерфейс пользователя. Прикладные программы или модули самой операционной системы передают запросы вверх и вниз по этим уровням.

Многоуровневые системы хорошо реализуются.

Слоеные системы хорошо модифицируются.

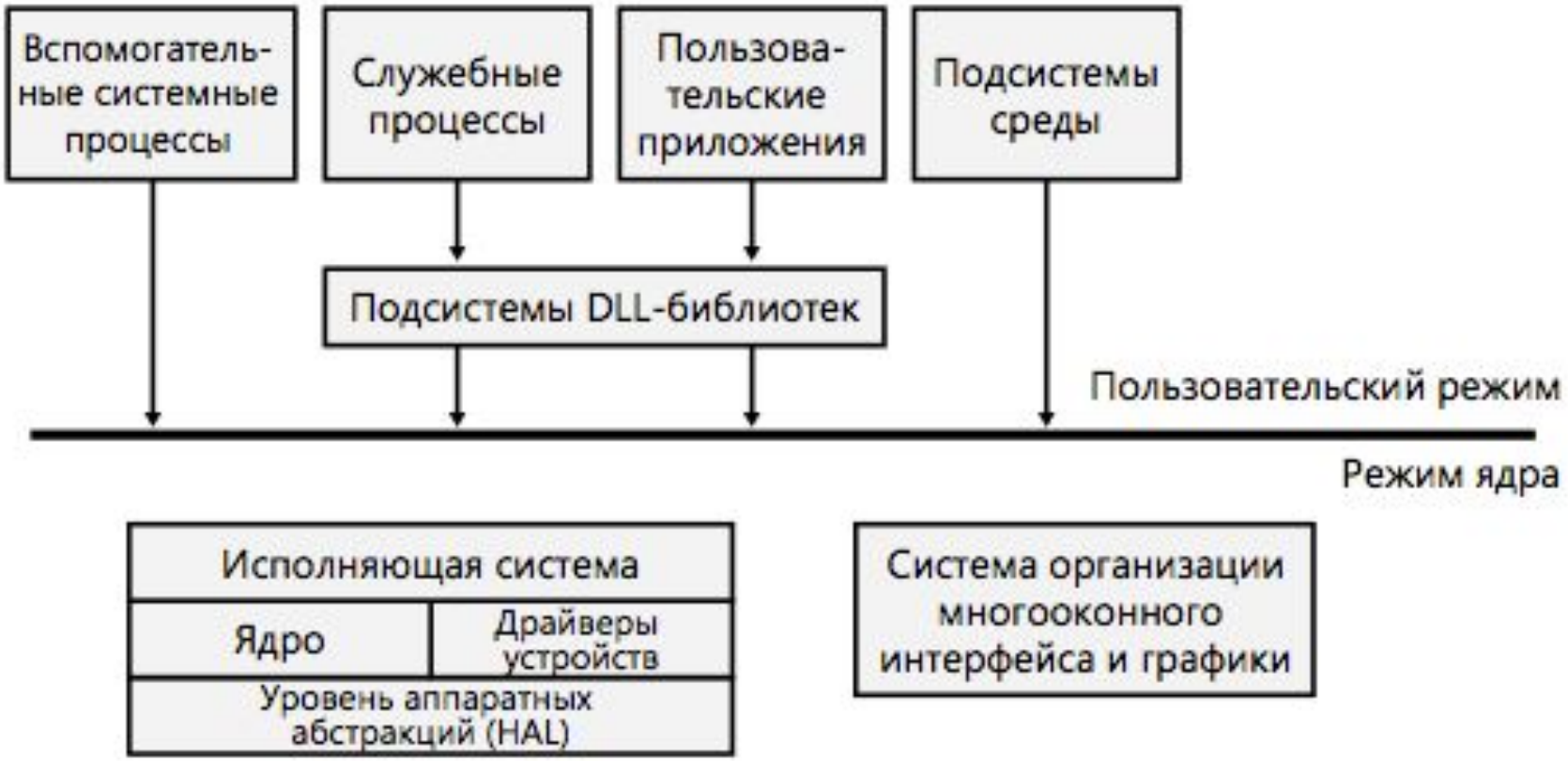
Многоуровневые системы сложны для разработки.

Менее эффективны, чем монолитные.

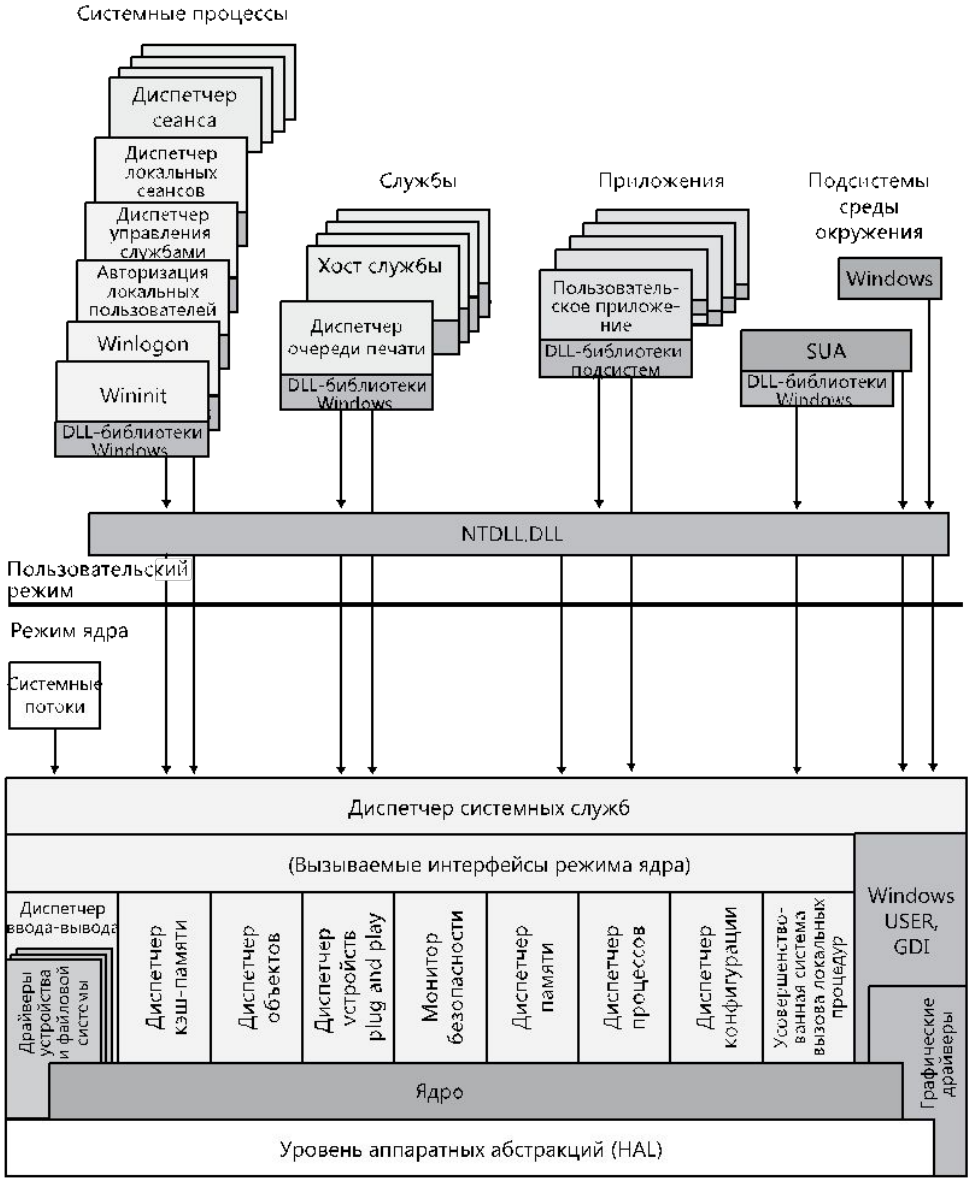
Примеры многоуровневых ОС

Структура ОС UNIX

Упрощенное представление архитектуры Windows

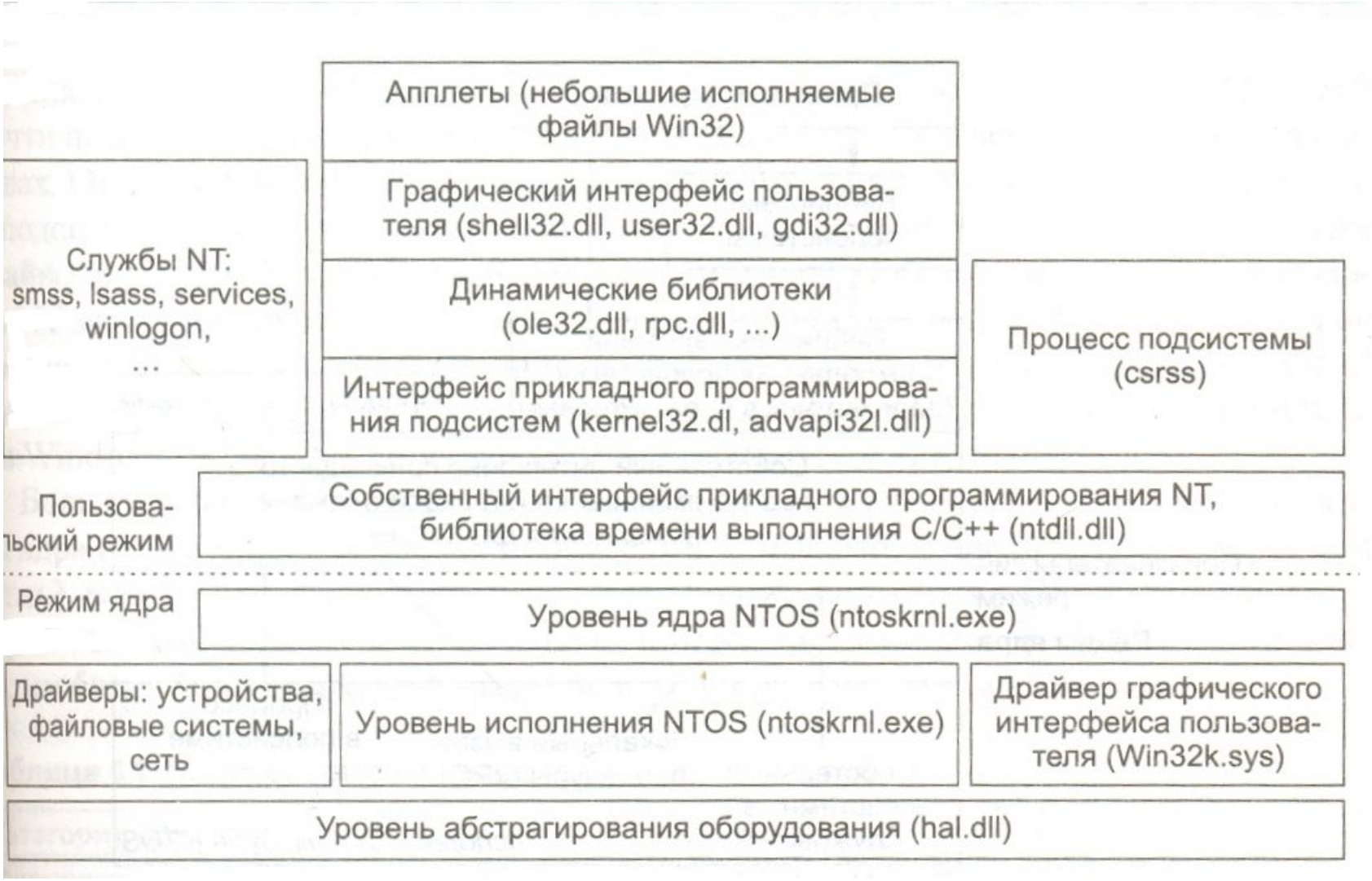


Архитектура Windows

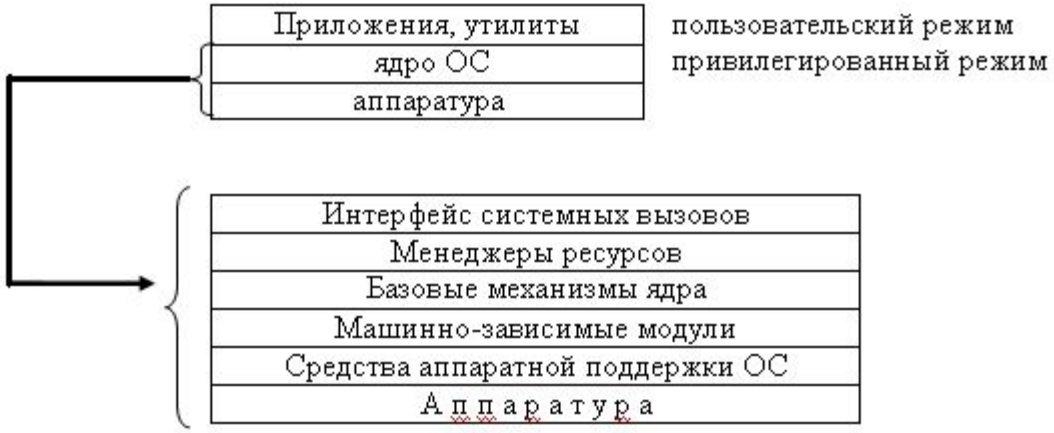


Аппаратные интерфейсы (шины, устройства ввода-вывода, прерывания, интервальные таймеры, DMA, управление кэш-памятью и т. д.)

Уровни программирования в Windows



Многослойная модель ядра



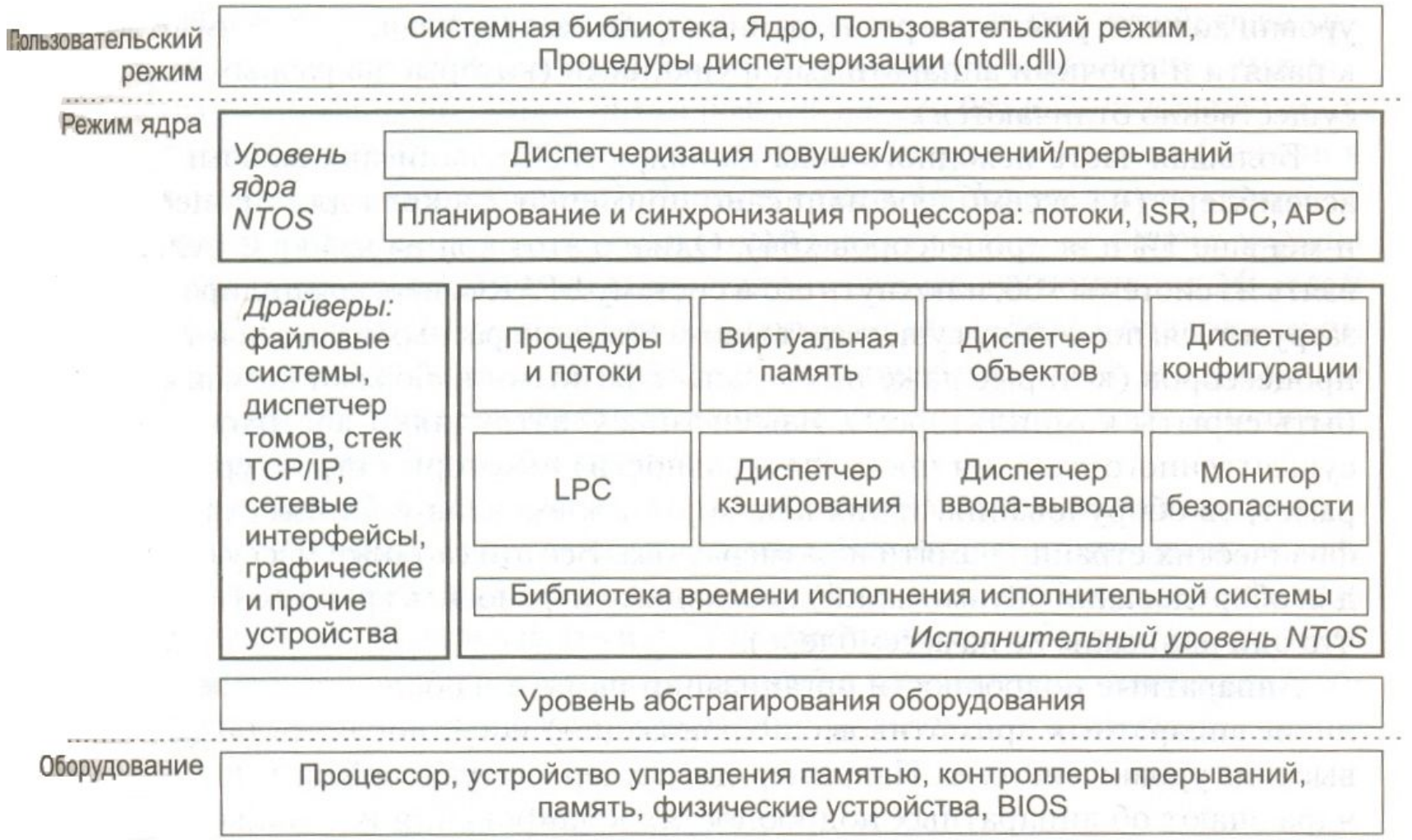
Ядро может состоять из следующих слоев:

- средства аппаратной поддержки (система прерываний, средства переключения контекстов процессов, средства поддержки привилегированного режима, средства защиты областей памяти и т. д.).
- машинно-зависимые компоненты ОС; в идеале этот слой полностью экранирует вышележащие слои ядра от особенностей аппаратуры (пример – слой HAL ОС Windows NT);
- базовые механизмы ядра, этот слой выполняет наиболее примитивные операции ядра, реализует решения о распределении ресурсов, принятые на более высоком уровне;
- менеджеры ресурсов; слой состоит из мощных функциональных модулей, реализующих стратегические задачи по управлению основными ресурсами ОС;
- интерфейс системных вызовов взаимодействует непосредственно с приложениями и системными утилитами, образуя прикладной программный интерфейс ОС.

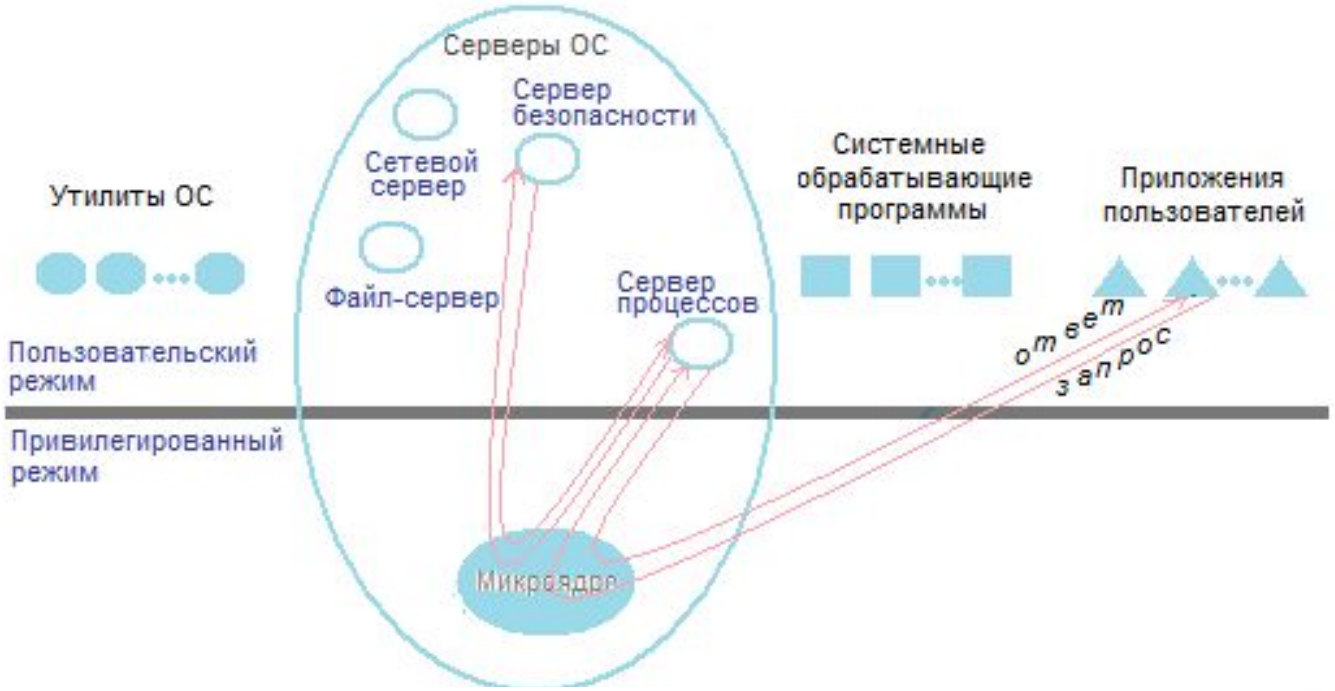
Ядро ОС UNIX


Системные вызовы
Аппаратные и эмулированные прерывания
Управление терминалом
Сокеты
Именование файла
Отображение адресов
Страничные прерывания
Обработка сигналов
Создание и завершение процессов
Необработанный телетайп
Обработанный телетайп
Сетевые протоколы
Файловые системы
Виртуальная память
Дисциплины линии связи
Маршрутизация
Буферный кэш
Драйверы сетевых устройств
Планирование процесса
Символьные устройства
Драйверы сетевых устройств
Драйверы дисковых устройств
Диспетчеризация процессов
Аппаратура

Организация режима ядра Windows

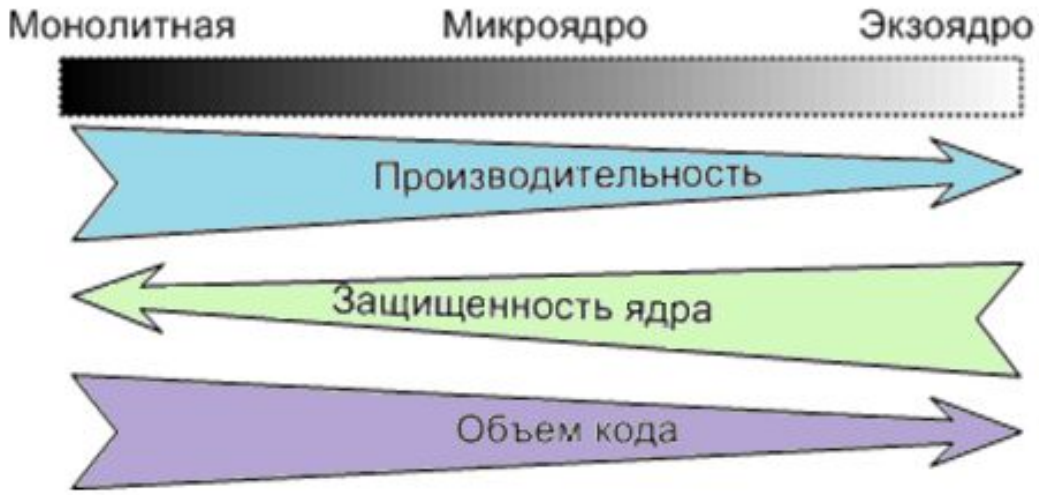


Микроядерная архитектура

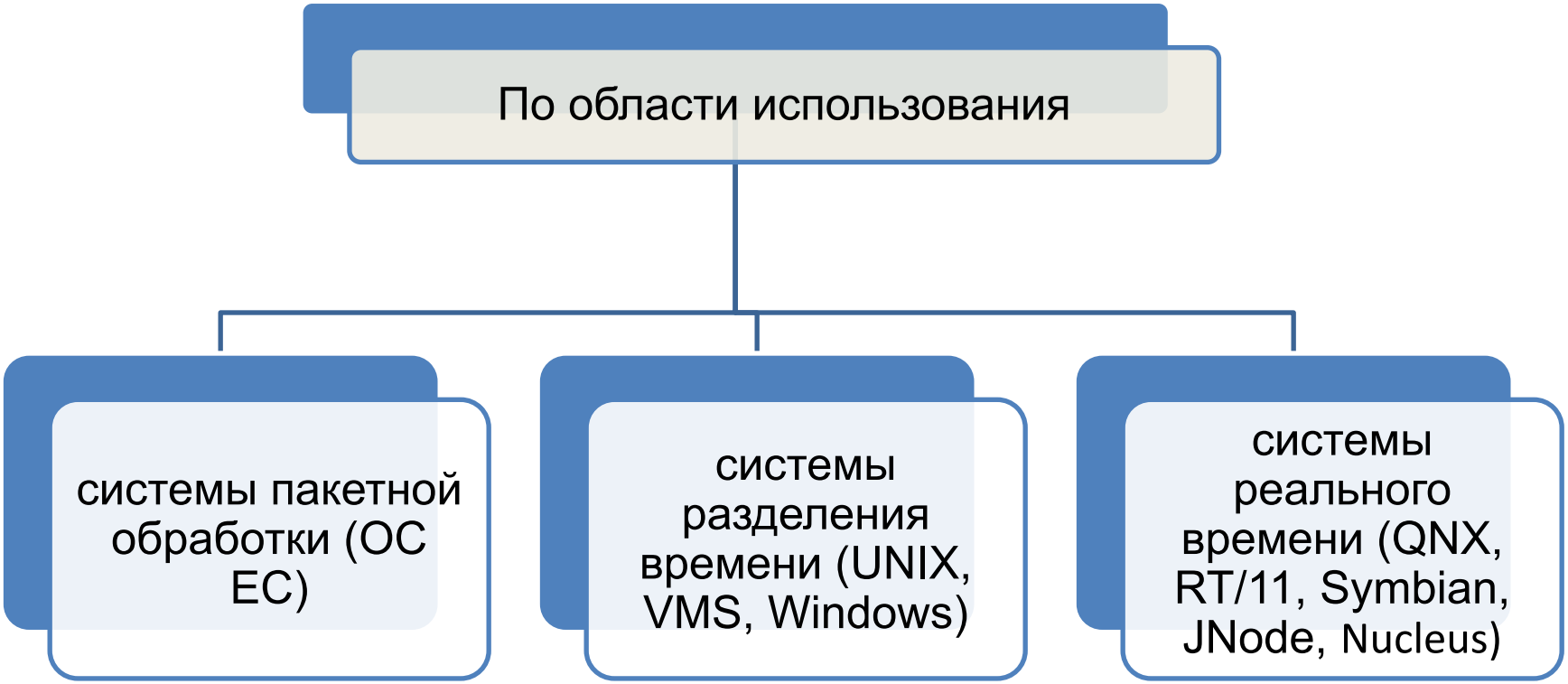


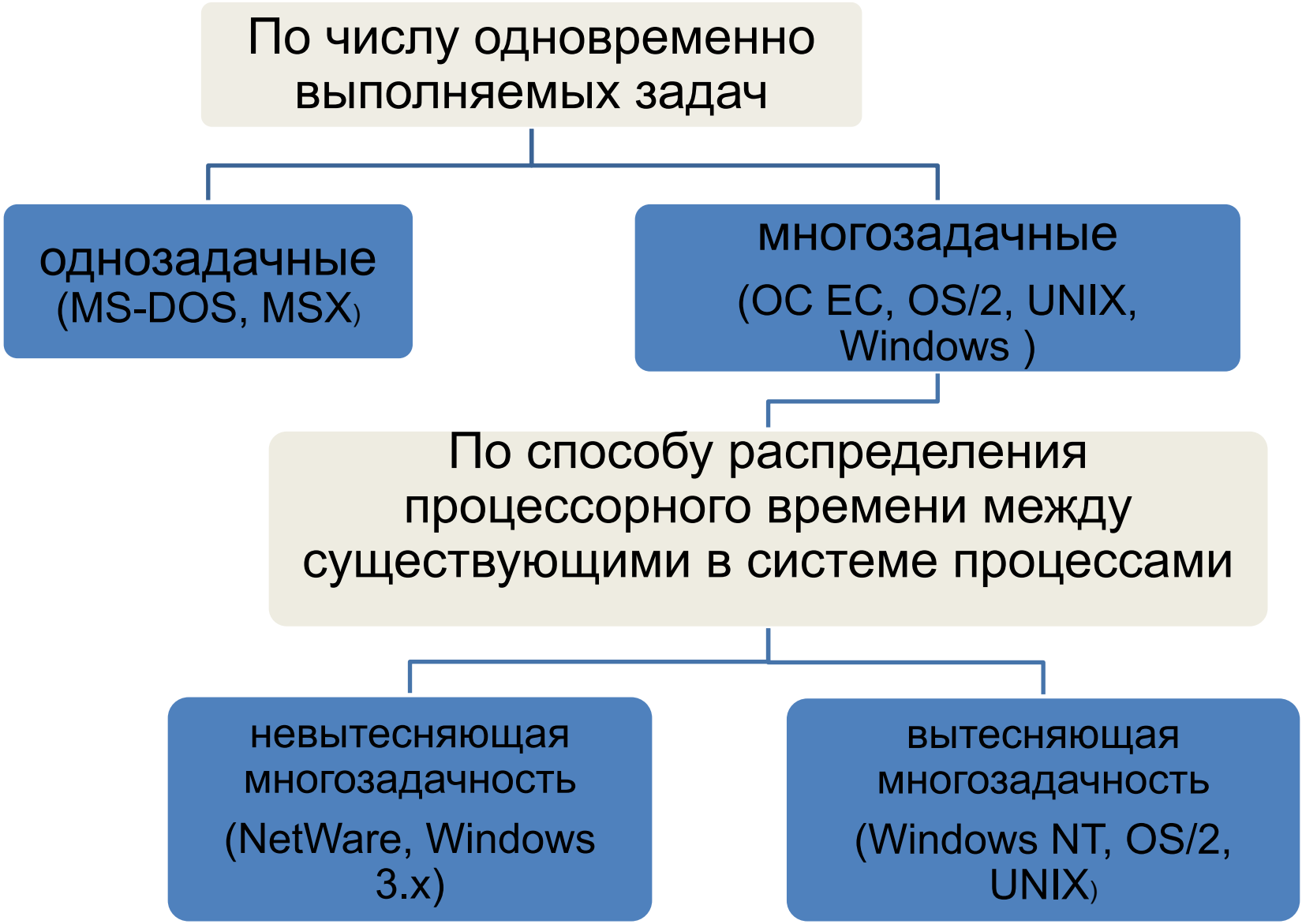
$$t_{з\Sigma} = 4t$$

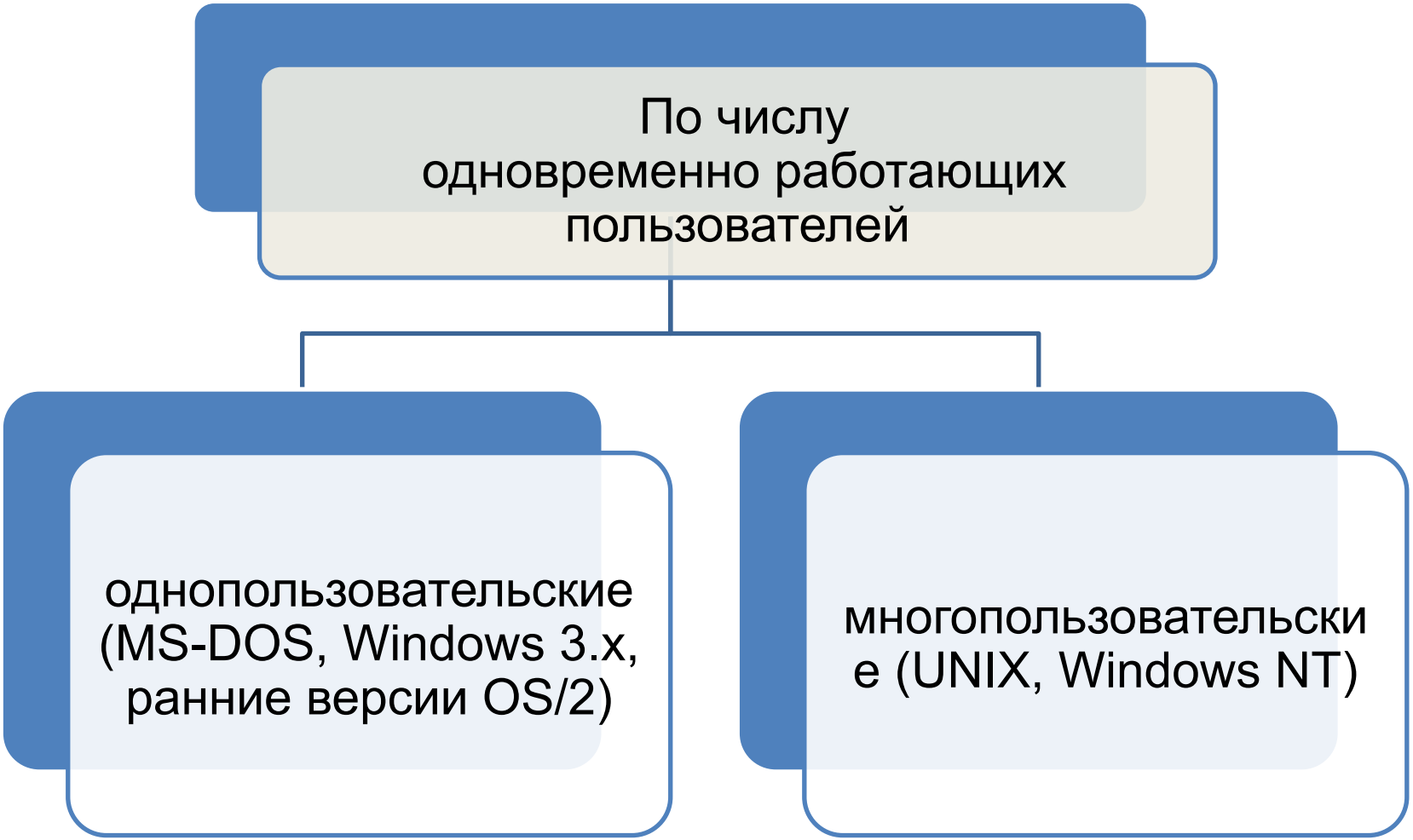
Сравнение моделей архитектур ОС



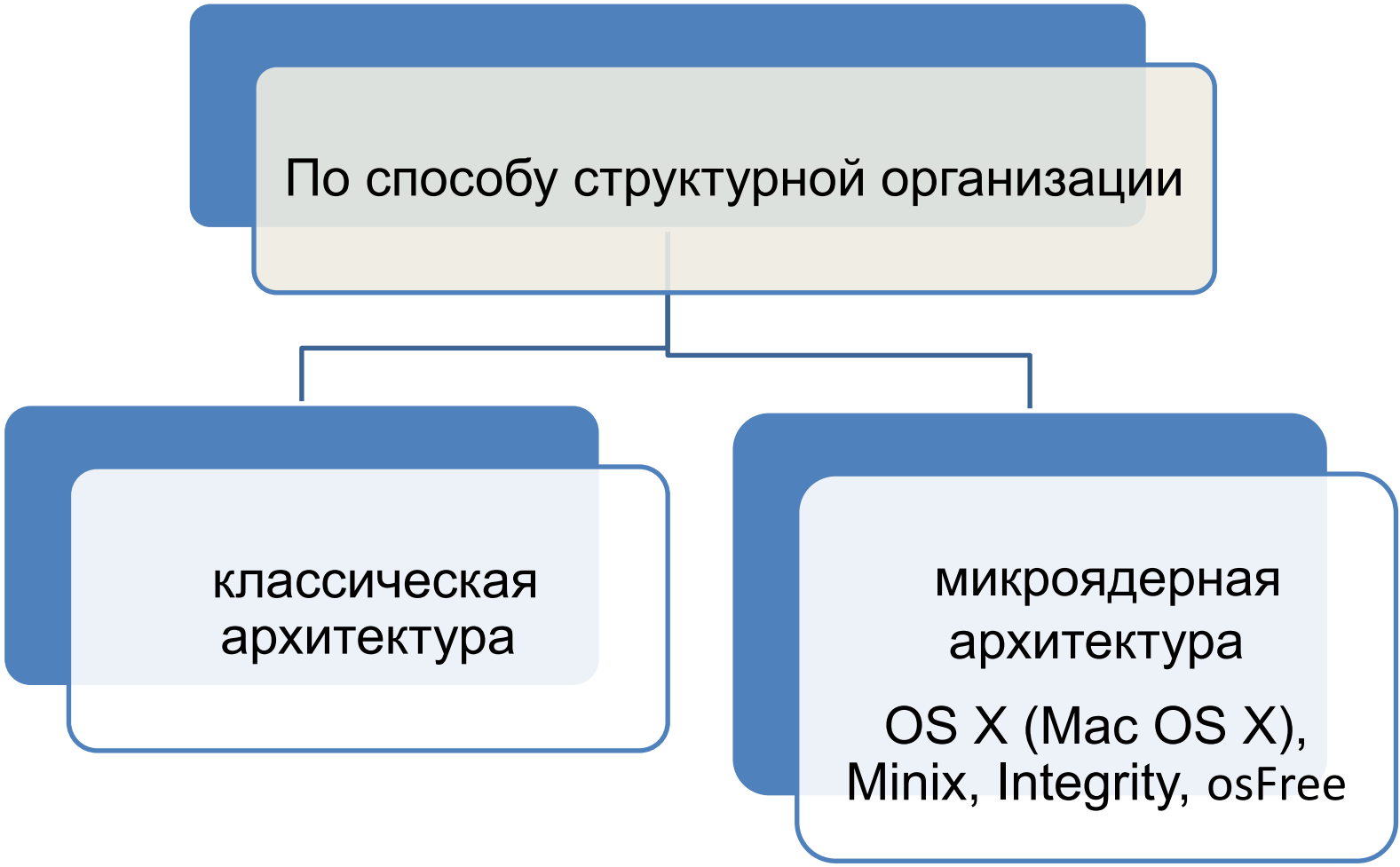
Классификации ОС




По числу одновременно выполняемых задач
однозадачные (MS-DOS, MSX)
многозадачные (ОС ЕС, OS/2, UNIX, Windows)
По способу распределения процессорного времени между существующими в системе процессами
невытесняющая многозадачность (NetWare, Windows 3.x)
вытесняющая многозадачность (Windows NT, OS/2, UNIX)

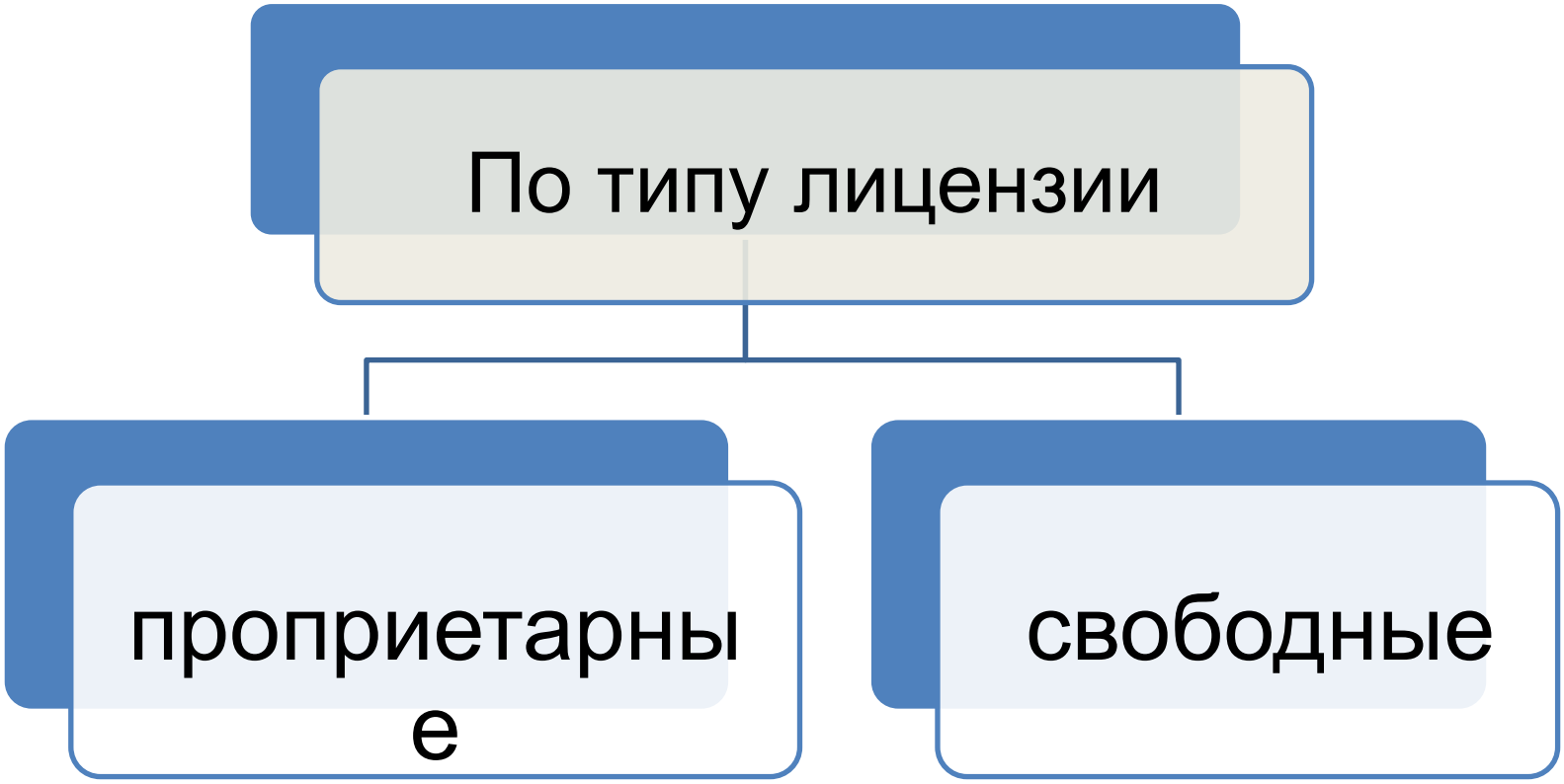

По числу
одновременно работающих
пользователей
однопользовательские
(MS-DOS, Windows 3.x,
ранние версии OS/2)
многопользовательски
е (UNIX, Windows NT)


Многопроцессорная обработка – такой способ организации вычислительного процесса в системах с несколькими процессорами, при котором несколько задач могут одновременно выполняться на разных процессорах системы
нет
поддерживается
симметричное мультипроцессирование
асимметричное мультипроцессирование


По способу структурной организации
классическая архитектура
микроядерная архитектура
OS X (Mac OS X), Minix, Integrity, osFree

По типу аппаратных платформ

- ОС мейнфреймов
- Серверные ОС
- ОС персональных компьютеров
- ОС смартфонов
- Встроенные ОС
- ОС сенсорных узлов
- ОС смарт-карт


По типу лицензии
проприетарные
свободные

Вымышленные операционные системы

ALTIMIT OS — из вселенной .hack
Hyper OS — из Patlabor
Wheatonix — первоапрельская шутка
Digitronix — из The Hacker Files
Luna/X — первоапрельская шутка Google в 2004 году
SEXLinux — первоапрельская шутка linuxcenter.ru (система на основе Gentoo)
Finux — из книги Нила Стивенсона «Криптономикон» («написанная финнами»)
Windows Хоум — из фантастической трилогии Сергея Лукьяненко «Лабиринт отражений»
Macrohard Nondows Twista — из компьютерной игры «Космические рейнджеры 2: Доминаторы. Перезагрузка»
Okna 96 — из мультсериала «Смешарики». Аллюзия на Windows 9x (Windows → «Окна» → Okna)
GLaDOS — из игры Portal
Copland OS — операционная система из аниме «Эксперименты Лэйн»

Эволюция ОС

- **Первый период (1945 -1955)**

Элементная база – электронные лампы и коммуникационные панели. Программирование осуществлялось исключительно на машинном языке. Операционных систем нет, все задачи организации вычислительного процесса решались вручную программистом с пульта управления. Системное программное обеспечение - библиотеки математических и служебных подпрограмм.

Программы-мониторы (середина 50-х) – прообраз современных ОС.

- **Второй период (1955 - 1965)**

Элементная база - полупроводниковые элементы (транзисторы).

Первые алгоритмические языки и, следовательно, первые системные программы - компиляторы.

Появились первые системы пакетной обработки, увеличивающие коэффициент загрузки процессора.

- **Третий период (1965 - 1975)**

Элементная база - интегральные микросхемы.

Создание семейств программно-совместимых машин (серия машин IBM System/360, советский аналог - машины серии ЕС).

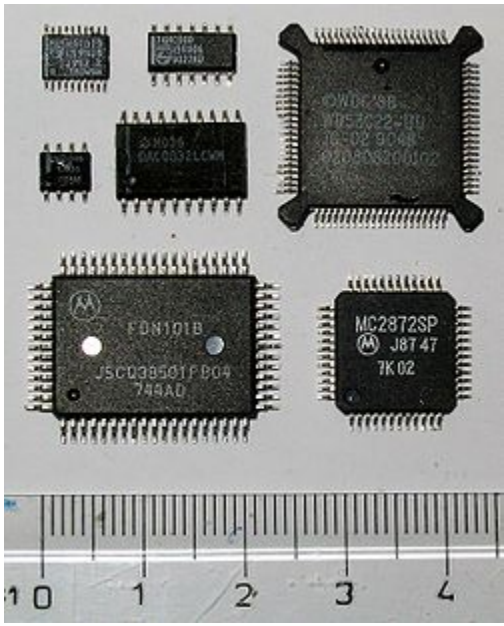
Реализованы практически все основные концепции, присущие современным ОС: мультипрограммирование, мультипроцессирование, многотерминальный режим, виртуальная память, файловая система, разграничение доступа и сетевая работа.

В процессорах появился привилегированный и пользовательский режим работы, специальные регистры для переключения контекстов, средства защиты областей памяти и система прерываний.

Появился новый тип ОС - системы разделения времени.

В конце 60-х годов начаты работы по созданию глобальной сети ARPANET.

К середине 70-х годов широкое распространение получили мини-компьютеры, создание первых локальных сетей. С середины 70-х годов началось массовое использование ОС UNIX. В конце 70-х был создан рабочий вариант протокола TCP/IP, в 1983 году он был стандартизирован.



Четвертый период (1980 - настоящее время)

Элементная база – большие и сверхбольшие интегральные схемы (БИС и СБИС). Эра персональных компьютеров.

- С середины 80-х бурное развитие сетей персональных компьютеров, развитие сетевых и распределенных операционных систем.
- В 1987г. была выпущена операционная система **MINIX** (прототип LINUX), она была построена по принципу микроядерной архитектуры.
- В 80-е годы были приняты основные стандарты на коммуникационное оборудование для локальных сетей: в 1980 году –Ethernet, в 1985 – Token Ring, в конце 80-х – FDDI. Это позволило обеспечить совместимость сетевых ОС на нижних уровнях, а также стандартизировать интерфейс ОС с драйверами сетевых адаптеров.
- В 90-е годы практически все ОС стали сетевыми. Появились специализированные ОС, предназначенные исключительно для решения коммуникационных задач (IOS компании Cisco Systems). Появление службы World Wide Web (WWW) в 1991 году придало мощный импульс развитию популярности Интернета. Развитие корпоративных сетевых операционных систем выходит на первый план. Возобновляется развитие ОС мейнфреймов. В 1991г. была выпущена **LINUX**. Чуть позже вышла **FreeBSD** (основой для нее послужила BSD UNIX).

Современный этап развития операционных систем

Эволюционное развитие свойств, механизмов и функций ОС, которые появились в 60-е и 90-е годы.

Тенденции развития – повышение удобства работы человека с компьютером, повышение надежности за счет использования микроядерной архитектуры, простота обслуживания, дружественный пользовательский интерфейс, эффективные средства поиска и хранения информации.

ОС суперкомпьютеров будут наделяться функциями поддержки виртуальных кластеров, способных разделять вычислительную мощность через Интернет.

.

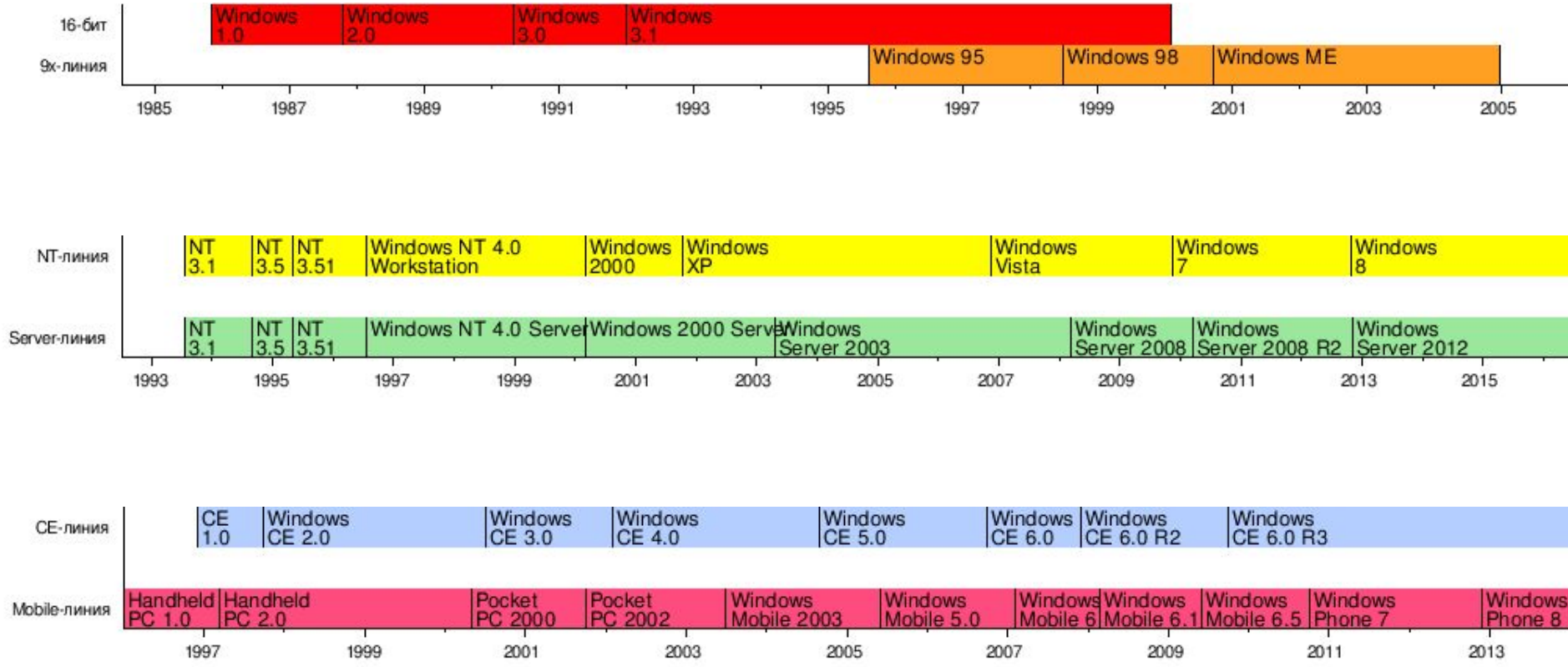
«Облачная обработка данных — это парадигма, в рамках которой информация постоянно хранится на серверах в интернете и временно кэшируется на клиентской стороне, например, на персональных компьютерах, игровых приставках, ноутбуках, смартфонах и т. д.»

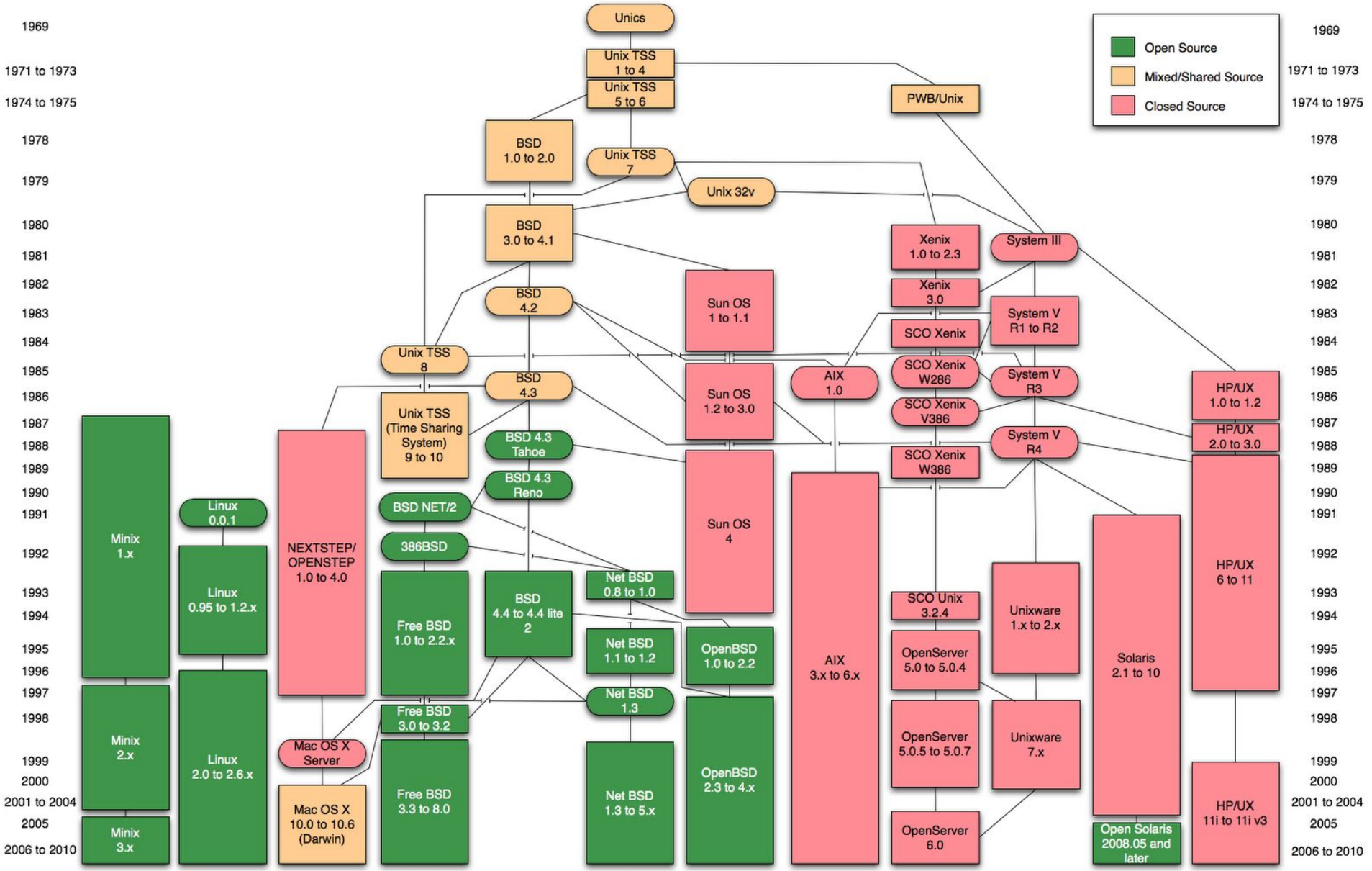
Облачная обработка данных включает в себя понятия:
«Всё как услуга»,
«Инфраструктура как услуга»,
«Платформа как услуга»,
«Программное обеспечение как услуга»,
«Данные как услуга»,
«Рабочее место как услуга»
другие технологические тенденции, общим в которых является уверенность, что сеть Интернет в состоянии удовлетворить потребности пользователей в обработке данных.

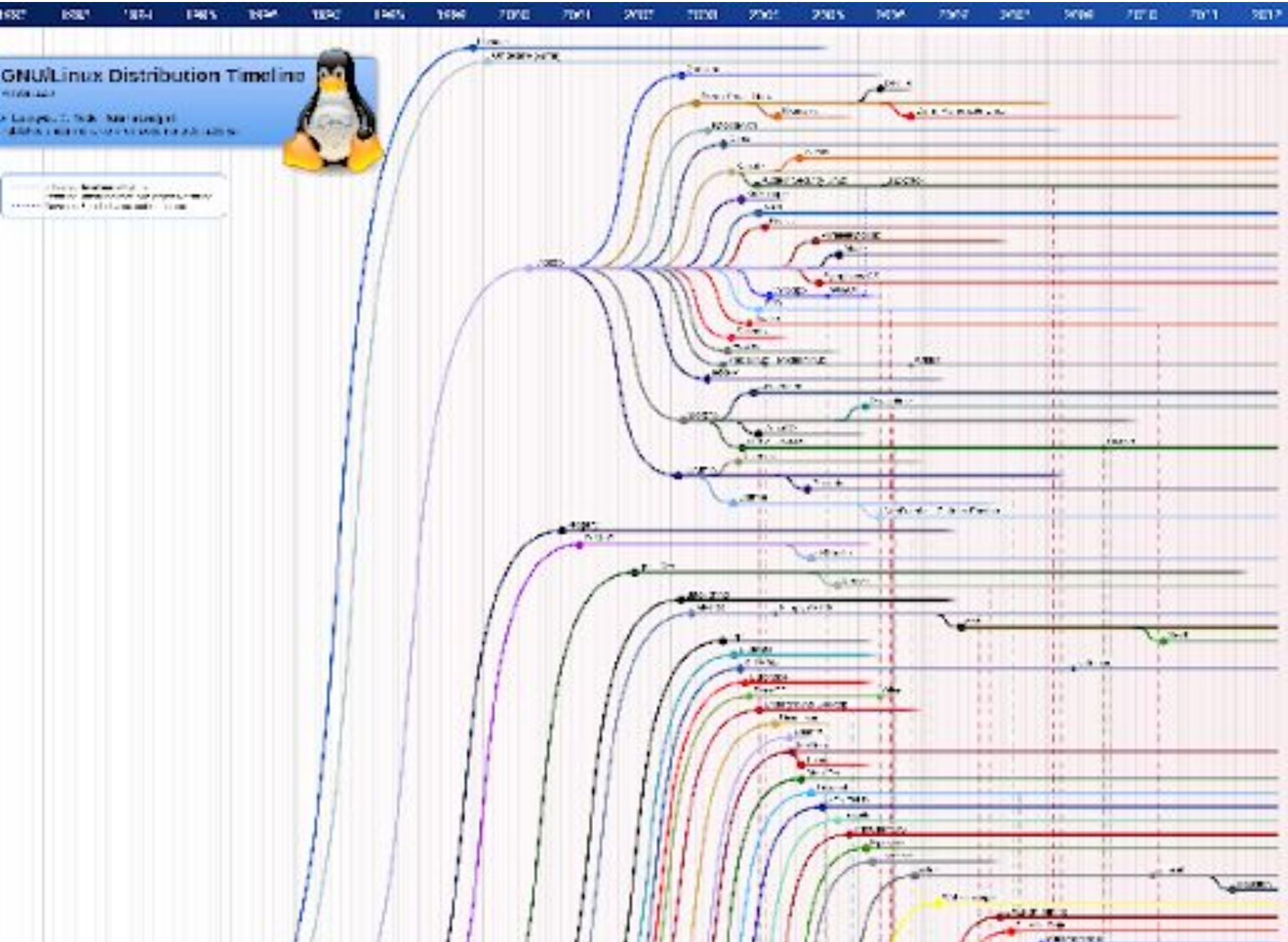
Cloud OS (от англ. ***Cloud Operating System — дословно: "Облачная" Операционная Система или операционная система на базе "облака") — клиент-серверное гибридное программное обеспечение, базирующееся на парадигме Cloud computing и использующее развитую систему многооконного интерфейса пользователя, функционирующего, обычно, в окне современного веб-браузера. Под понятием "облака", в данном случае, скрывается Интернет, точнее, совокупность объединённых глобальной сетью аппаратных ресурсов, использующихся для хранения и обработки данных в контексте конкретной реализации Cloud OS.Термин Cloud OS является синонимом Internet OS (в большей степени) и WebOS.[править]***

2006	ОС Windows Vista Старт второй очереди проекта виртуальной распределенной вычислительной системы (GRID)
2000	Windows 2000
1995	ОС для мэйнфреймов OS/390
1993	Windows NT 3.1; NetWare 4.0
1991	Первая ОС семейства LINUX
1987	OS/2 - первая мультипрограммная ОС для персонального компьютера
1985	Первая ОС семейства Windows
1984	Первая ОС семейства Mac
1963	Первая ОС семейства NetWare
1981	Операционная система MS-DOS для персонального компьютера
1980	Принят стандарт технологии Ethernet локальных сетей
1979	Рабочий вариант стека TCP/IP
1976	Первое сетевое приложение UUCP для Unix
1971	Сетевые технологии SNA и X.25
1970	Глобальная вычислительная сеть ARPANET
1969	Первая ОС семейства Unix
1968	Многомашинная система разделения времени АИСТ
1965	Первые мультипрограммные ОС: MULTICS. OS/360
1964	Первое семейство програмно совместимых компьютеров IBM Svsten/ 360
1962	Программный монитор для БЭСМ-6
1961	Первая реализация виртуальной памяти компанией Burroughs для ее компьютера B5000

Операционные системы Windows
Хронологическая схема




1969
1971 to 1973
1974 to 1975
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 to 2004
2005
2006 to 2010
Open Source
Mixed/Shared Source
Closed Source
Unics
Unix TSS 1 to 4
Unix TSS 5 to 6
PWB/Unix
BSD 1.0 to 2.0
Unix TSS 7
Unix 32v
BSD 3.0 to 4.1
Xenix 1.0 to 2.3
System III
Xenix 3.0
BSD 4.2
Sun OS 1 to 1.1
System V R1 to R2
SCO Xenix
Unix TSS 8
SCO Xenix W286
AIX 1.0
System V R3
BSD 4.3
Sun OS 1.2 to 3.0
HP/UX 1.0 to 1.2
SCO Xenix V386
HP/UX 2.0 to 3.0
Unix TSS (Time Sharing System) 9 to 10
BSD 4.3 Tahoe
System V R4
SCO Xenix W386
BSD 4.3 Reno
Linux 0.0.1
BSD NET/2
Sun OS 4
386BSD
Minix 1.x
NEXTSTEP/ OPENSTEP 1.0 to 4.0
Net BSD 0.8 to 1.0
HP/UX 6 to 11
Linux 0.95 to 1.2.x
Free BSD 1.0 to 2.2.x
BSD 4.4 to 4.4 lite 2
SCO Unix 3.2.4
Unixware 1.x to 2.x
Net BSD 1.1 to 1.2
OpenBSD 1.0 to 2.2
AIX 3.x to 6.x
OpenServer 5.0 to 5.0.4
Solaris 2.1 to 10
Net BSD 1.3
Free BSD 3.0 to 3.2
Minix 2.x
Linux 2.0 to 2.6.x
Mac OS X Server
OpenServer 5.0.5 to 5.0.7
Unixware 7.x
OpenBSD 2.3 to 4.x
Free BSD 3.3 to 8.0
Net BSD 1.3 to 5.x
Mac OS X 10.0 to 10.6 (Darwin)
OpenServer 6.0
HP/UX 11i to 11i v3
Minix 3.x
Open Solaris 2008.05 and later


GNU/Linux Distribution Timeline