

**Направления подготовки: «Информатика и вычислительная техника» и «Информационные системы и технологии»**

**Профили образовательных программ:**

**«Системотехника и автоматизация проектирования в строительстве»**

**«Системотехника и информационные технологии управления в строительстве»**

# **ОПЕРАЦИОННЫЕ СИСТЕМЫ**

## **Тема 3. Классификация операционных систем**

№	Наименование раздела дисциплины	Тема и содержание лекций
1	Основные понятия	<p>Тема 1. Основные сведения об операционных системах. История развития операционных систем. Доступ к ЭВМ: локальный непосредственный, через оператора, удаленный. Режимы решения задач в ЭВМ: пакетный, индивидуальный, разделение времени, реального времени. Операционная система как часть вычислительной системы. Принципы построения вычислительных систем.</p>
		<p>Тема 2. Назначение и функции операционной системы. Требования к современным операционным системам. Основные функциональные компоненты ОС. Требования, предъявляемые к современным ОС. Расширяемость. Переносимость. Совместимость и множественные прикладные среды. Безопасность.</p>
		<p>Тема 3. Классификация операционных систем. Особенности областей использования. Особенности алгоритмов управления ресурсами. Особенности аппаратных платформ. Особенности методов построения ОС. Сетевые операционные системы.</p>
		<p>Тема 4. Архитектура современных операционных систем. Модульная структура построения ОС. Ядро и модули расширения ядра. Режимы работы аппаратуры. Многослойная структура ядра операционной системы. Микроядерная архитектура.</p>

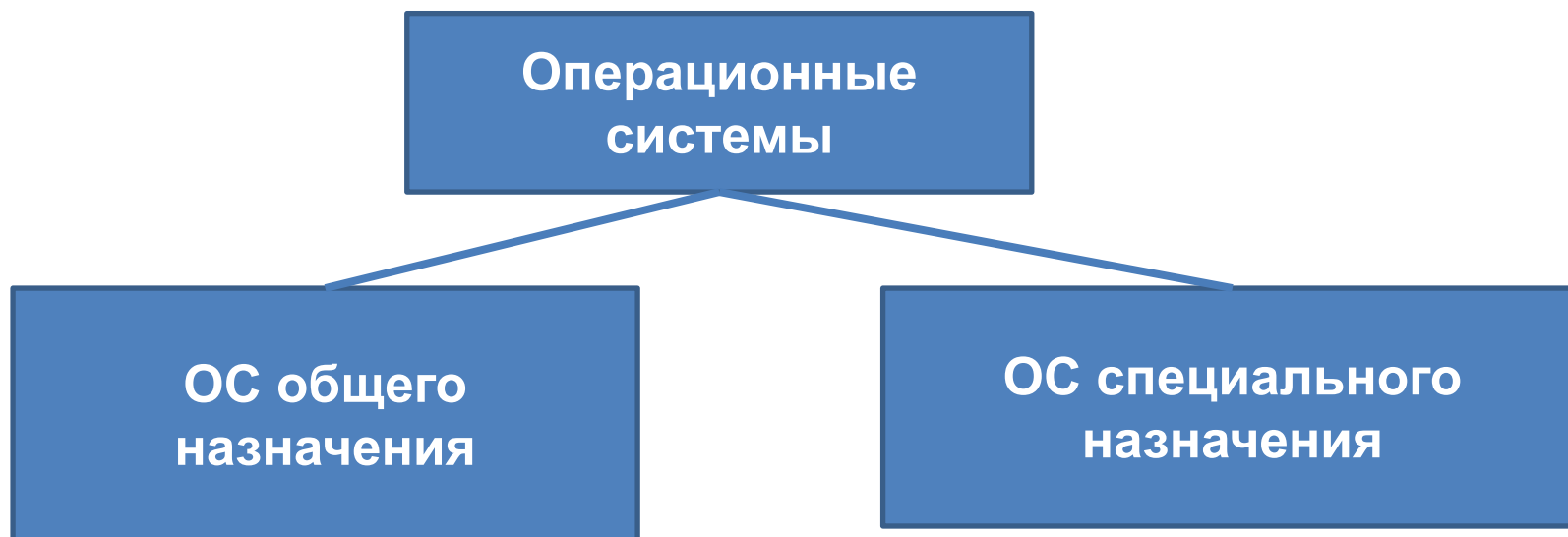
**Операционная система (ОС)** – комплекс управляющих и обрабатывающих программ, который, с одной стороны, выступает как **интерфейс между аппаратурой компьютера и пользователем с его задачами**, а с другой – предназначен для **наиболее эффективного использования ресурсов вычислительной системы и организации надежных вычислений**.





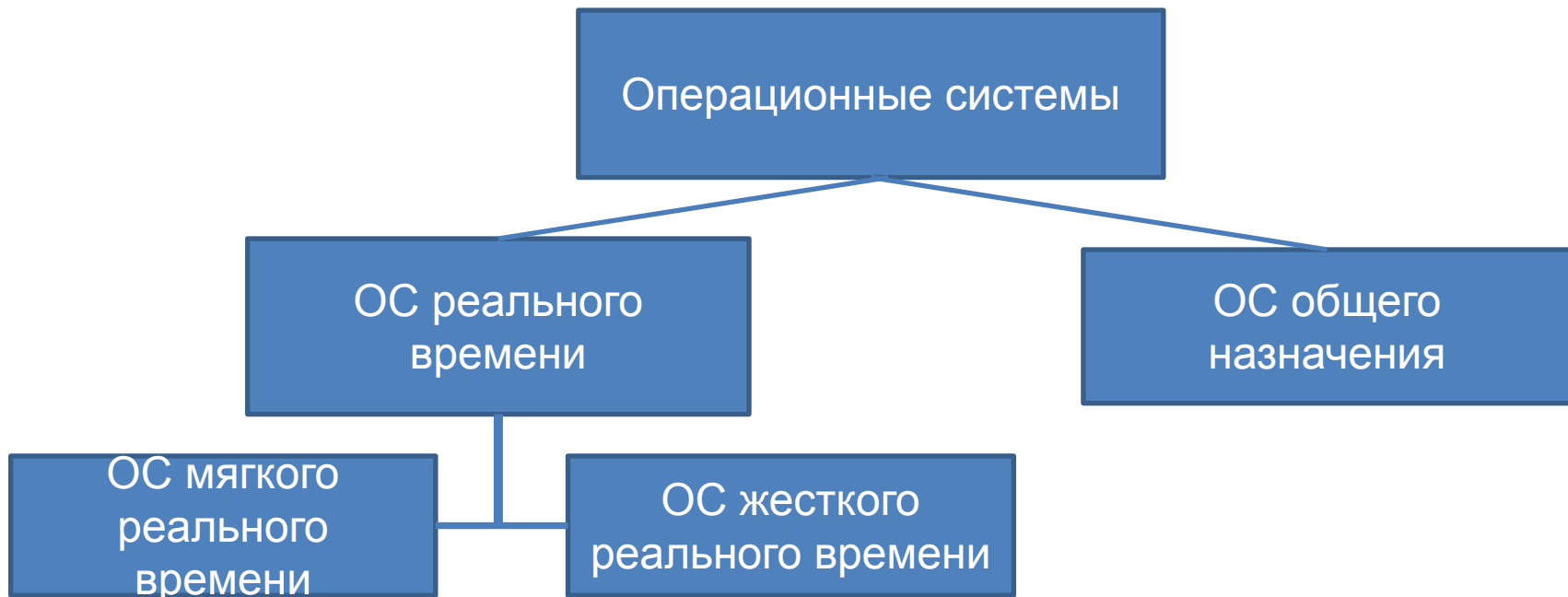
Применительно к операционным системам сформировалось относительно устойчивое число признаков для классификации:

- по назначению,
- по способности операционной системы обеспечить требуемый уровень сервиса в определённый промежуток времени,
- по режиму обработки задач,
- по способу взаимодействия пользователя с системой,
- по способам построения (архитектурным особенностям системы).



ОС для персональных компьютеров относятся к операционным системам общего назначения, поскольку их можно использовать для самых разнообразных целей.

ОС различных встроенных систем, например, таких как устройства бытовой электроники, контроллеры для управления технологическими процессами и устройства управления коммутационным оборудованием.



Основной особенностью операционных систем реального времени является *обеспечение обработки поступающих заданий в течение заданных интервалов времени, которые нельзя превышать.*



## Операционные системы реального времени

### ОС **жесткого** реального времени

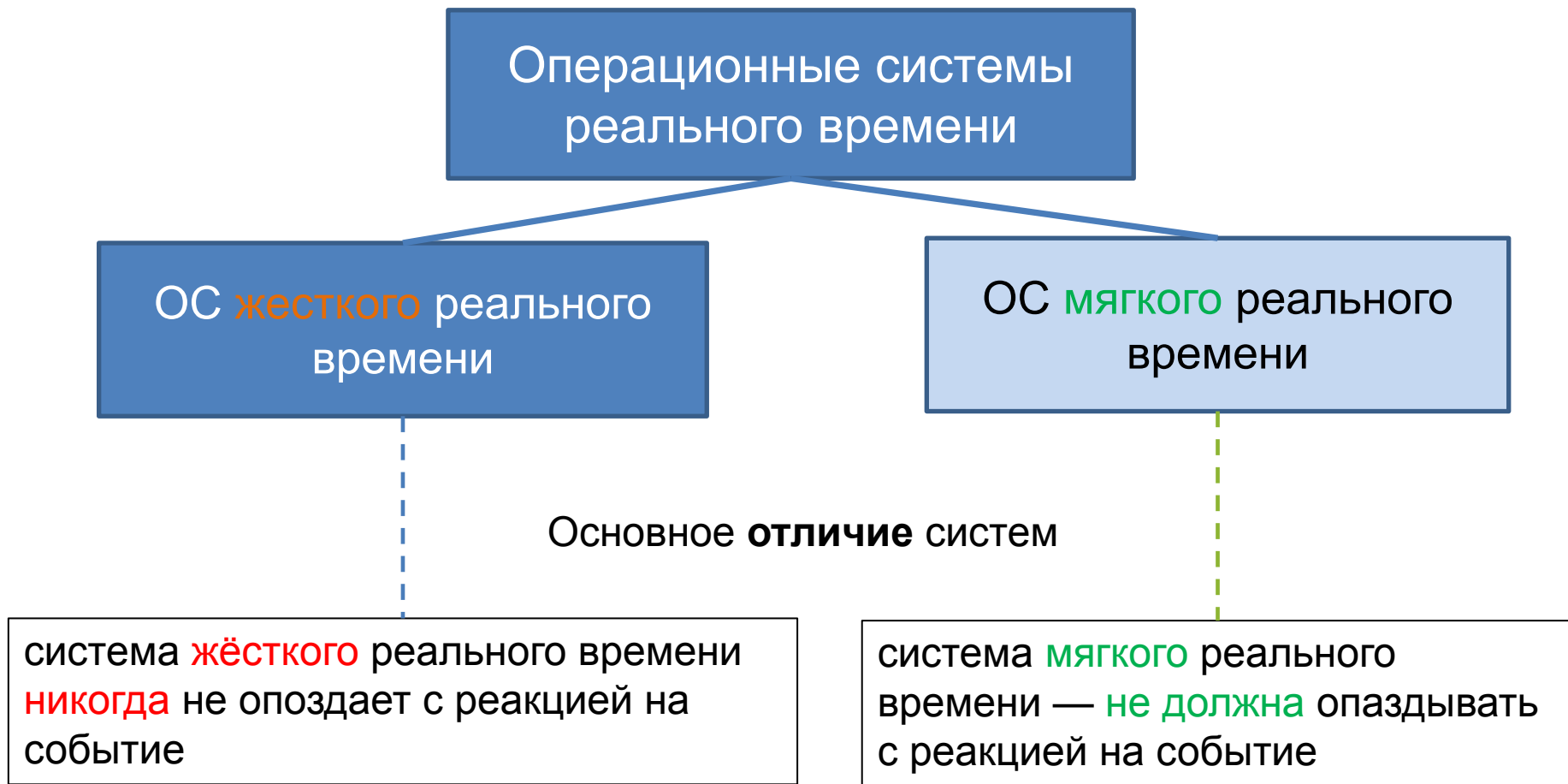
Системы жёсткого реального времени *не допускают задержек реакции системы, так как это может привести к потере актуальности результатов, большим финансовым потерям, авариям и катастрофам.*

К таким системам относятся системы управления бортового оборудования, системы аварийной защиты, регистраторы аварийных событий.

### ОС **мягкого** реального времени

Системы мягкого реального времени характеризуются **возможностью задержки реакции, что может привести к увеличению стоимости результатов и снижению производительности системы в целом.**

Примером может служить работа компьютерной сети.







Операционная система реального времени QNX  
Инструменты для создания надёжных встраиваемых систем и  
интеллектуальных устройств любой сложности



Компания Продукты Обучение Внедрения Материалы Поддержка Скачать



▶ Продукты QNX

- ▶ **Операционная система реального времени QNX Neutrino**
- ▶ Операционная система реального времени QNX4
- ▶ Комплект разработчика QNX Momentics
- ▶ Встраиваемые СУБД
- ▶ Аппаратное обеспечение

[Главная страница](#) > [Продукты](#) > [Продукты QNX](#) > Операционная система реального времени QNX Neutrino



## Операционная система реального времени QNX Neutrino

Выбирая QNX Neutrino как платформу для Ваших решений, Вы сможете:

- ▶ Проектировать чрезвычайно надежные системы с возможностью динамического обновления во время работы. В итоге Вы получаете выигрыш в продлении жизненного цикла системы.
- ▶ Создавать системы, способные к самостоятельному интеллектуальному восстановлению и устранению последствий сбоя. В результате увеличивается доступность предоставляемых сервисов.
- ▶ Создавать системы любого масштаба, от малых до больших. QNX Neutrino предоставляет встроенную поддержку симметричной мультипроцессорной обработки данных (SMP) и многоядерных процессоров.
- ▶ Упростить проектирование отказоустойчивых кластеров, а также добавлять вычислительные мощности путем простого включения дополнительного процессора в сеть
- ▶ Улучшить производительность благодаря быстрым и предсказуемым откликам жесткого реального времени
- ▶ Создавать переносимые приложения, полностью соответствующие стандарту POSIX. Вы также получаете возможность быстро переносить в QNX приложения Linux, Unix и другие приложения с открытыми исходными кодами.
- ▶ Оснастить Вашу систему развитым графическим интерфейсом, построенным с использованием Web-технологий, многослойности и 3D графики.
- ▶ Адресовать максимально возможное для Вашего процессора адресное пространство благодаря расширенной поддержке памяти, выходящей за границы 4-х Гб. Эта возможность доступна для процессоров MIPS, PowerPC и x86.
- ▶ Использовать наиболее подходящее для Ваших задач оборудование благодаря поддержке и наличию BSP для множества чипсетов, включая MIPS, PowerPC, SH-4, ARM, StrongArm, Intel XScale и x86.

### Тематические материалы

- [Микроядерная архитектура](#) ▶
- [Производительность жесткого реального времени](#) ▶
- [Обмен сообщениями](#) ▶
- [Совместимость с POSIX](#) ▶
- [Адаптивное квотирование системных ресурсов](#) ▶
- [Менеджер высокой готовности](#) ▶
- [Поддержка многоядерных процессоров](#) ▶
- [Мультипроцессорная обработка данных \(SMP\)](#) ▶
- [Технология минидрайверов](#) ▶
- [Прозрачные распределенные вычисления](#) ▶
- [Графические решения](#) ▶
- [Мультимедийные решения](#) ▶
- [Сетевые технологии](#) ▶
- [Файловые системы](#) ▶
- [Среда управления ресурсами](#) ▶
- [Пакеты поддержки плат \(BSP\)](#) ▶
- [Инструментальное ядро](#) ▶
- [GNU C/C++](#) ▶
- [Виртуальные машины M3K-61131-3](#) ▶
- [Средства поддержки кириллицы для QNX6](#) ▶

<http://www.swd.ru/index.php3?pid=377>



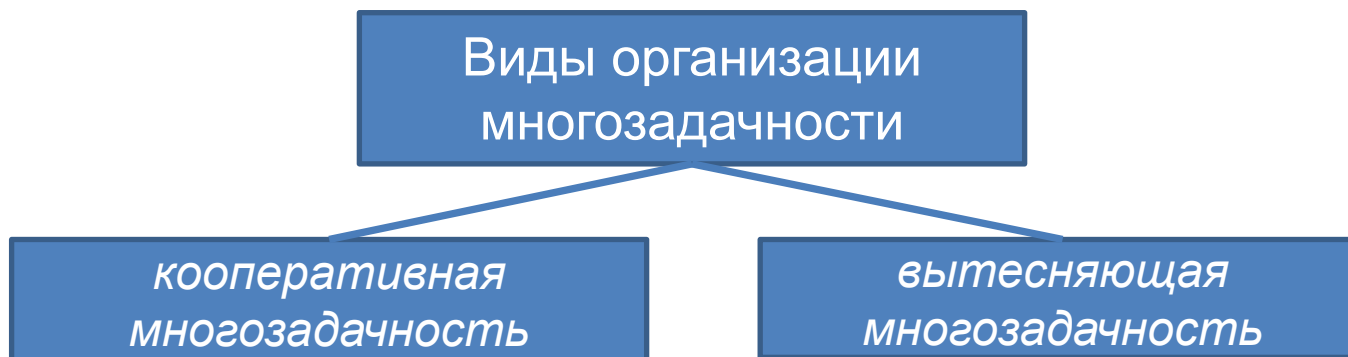
## Операционные системы

### ОС однозадачные

В однозадачных операционных системах после запуска управление передаётся прикладной программе, которая получает в своё распоряжение **все ресурсы компьютера** и может осуществлять ввод/вывод посредством как функций предоставляемых операционной системой, так и функций базовой системы ввода/вывода (BIOS), а также работать с устройствами напрямую.

### ОС многозадачные

Многозадачность — свойство операционной системы обеспечивать возможность *параллельной (псевдопараллельной) обработки* нескольких процессов.  
*Истинная* многозадачность операционной системы возможна только в *многопроцессорных (распределенных)* вычислительных системах. Многозадачные системы проводят **распределение ресурсов динамически**, когда задача стартует в памяти или покидает память в зависимости от её приоритета и от стратегии системы.



Кооперативная многозадачность — тип многозадачности, при котором следующая задача выполняется только после того, как *текущая задача явно объявит себя готовой отдать процессорное время операционной системы*. Примером могут служить Windows версий до 3.x включительно, а также Windows 95/98/Me при использовании 16-битных приложений. Недостатки: неспособность всех приложений работать в случае ошибки в одном из них, приводящей к отсутствию вызова операции «отдать процессорное время».

Вытесняющая многозадачность — это вид многозадачности, при которой *операционная система по окончании положенного кванта времени переводит задачу* из состояния выполнения в состояние готовности, отдавая ресурсы другим задачам. Благодаря этому, зависшие приложения, как правило, не приводят к зависанию операционной системы.



- каждая задача имеет свой приоритет, в соответствии с которым получает процессорное время и память;
- система организует очереди задач так, чтобы все задачи получили ресурсы, в зависимости от приоритетов и стратегии системы;
- система организует обработку прерываний, по которым задачи могут активироваться, деактивироваться и удаляться;
- при нехватке памяти образы или их части невыполняющихся задач могут быть вытеснены на диск (*свопинг*), а потом через определённое системой время, восстанавливаться в памяти;
- система обеспечивает защиту адресного пространства задачи от несанкционированного вмешательства других задач;
- система обеспечивает защиту адресного пространства своего ядра от несанкционированного вмешательства задач;
- система распознаёт сбои и зависания отдельных задач и прекращает их;
- система решает конфликты доступа к ресурсам и устройствам, не допуская тупиковых ситуаций общего зависания при ожидании заблокированных ресурсов;
- система гарантирует каждой задаче, что рано или поздно она будет активирована;
- система обеспечивает коммуникацию между задачами.



## Способы взаимодействия пользователя с ОС

### Диалоговые системы

При организации работы с операционной системой в **диалоговом режиме** можно говорить об **однопользовательских** (однотерминальных) и о **многотерминальных** ОС. В многотерминальных ОС с одной вычислительной системой одновременно могут работать несколько пользователей, каждый со своего терминала. При этом у пользователей возникает иллюзия, что у каждого из них имеется собственная вычислительная система.

### Системы пакетной обработки

Необходимость оптимального использования дорогостоящих вычислительных ресурсов привела к появлению концепции «пакетного режима» исполнения программ. **Пакетный режим** предполагает наличие очереди программ на исполнение, причём ОС может обеспечивать загрузку программы с внешних носителей данных в оперативную память, не дожидаясь завершения исполнения предыдущей программы, что позволяет избежать проста процессора.



# Классификация по типу архитектуры ядра ОС

