

# Операционные системы реального времени (RTOS)

# Понятие операционной системы

Операционная система была создана для того, чтобы автоматизировать работу оператора и скрыть от пользователя сложности общения с аппаратурой, предоставив ему более удобную систему команд.

Таким образом, операционная система — это программный интерфейс между пользователем и аппаратурой вычислительной машины.

# Понятие операционной системы

С точки зрения пользователя ОС выполняет функцию расширенной машины или виртуальной машины, в которой легче программировать и легче работать, чем непосредственно с аппаратным обеспечением, составляющим реальный компьютер. Операционная система предоставляет простой, абстрагированный интерфейс, скрывая множество кропотливой работы с прерываниями, счетчиками времени, организацией памяти и другими компонентами низкого уровня. Это **взгляд сверху вниз**.

**Взгляд снизу вверх** дает представление об операционной системе как о менеджере, присутствующем в компьютере для управления всеми компонентами этой сложнейшей системы. Так, работа операционной системы заключается в обеспечении организованного и контролируемого распределения процессоров, памяти, дисков, принтеров, устройств ввода-вывода, датчиков времени и т.п. между различными программами, конкурирующими за право их использовать.

# Понятие операционной системы



**Расположение операционной системы в иерархической структуре программного и аппаратного обеспечения компьютера.**

Уточним первоначальное определение.

Операционная система – это набор программ, контролирующих работу прикладных программ и системных приложений и исполняющих роль интерфейса между пользователями, программистами, прикладными программами, системными приложениями и аппаратным обеспечением компьютера.

# ОС как расширенная виртуальная машина

Вместо реальной аппаратуры компьютера ОС представляет пользователю расширенную виртуальную машину, с которой удобнее работать и которую легче программировать. Вот список основных сервисов, предоставляемых типичными операционными системами.

**1. Разработка программ:** ОС представляет программисту разнообразные инструменты разработки приложений: редакторы, отладчики и т.п. Ему не обязательно знать, как функционируют различные электронные и электромеханические узлы и устройства компьютера. Часто пользователь не знает даже системы команд процессора, поскольку он может обойтись мощными высокоуровневыми функциями, которые представляет ОС.

**2. Исполнение программ.** Для запуска программы нужно выполнить ряд действий: загрузить в основную память программу и данные, инициализировать устройства ввода-вывода и файлы, подготовить другие ресурсы. ОС выполняет всю эту рутинную работу вместо пользователя.

# ОС как расширенная виртуальная машина

**3. Доступ к устройствам ввода-вывода.** Для управления каждым устройством используется свой набор команд. ОС предоставляет пользователю единообразный интерфейс, который скрывает все эти детали и обеспечивает программисту доступ к устройствам ввода-вывода с помощью простых команд чтения и записи. Если бы программист работал непосредственно с аппаратурой компьютера, то для организации, например, чтения блока данных с диска ему пришлось бы использовать более десятка команд с указанием множества параметров. После завершения обмена программист должен был бы предусмотреть еще более сложный анализ результата выполненной операции.

**4. Контролируемый доступ к файлам.** При работе с файлами управление со стороны ОС предполагает не только глубокий учет природы устройства ввода-вывода, но и знание структур данных, записанных в файлах. Многопользовательские ОС, кроме того, обеспечивают механизм защиты при обращении к файлам.

**5. Системный доступ.** ОС управляет доступом к совместно используемой или общедоступной вычислительной системе в целом, а также к отдельным системным ресурсам. Она обеспечивает защиту ресурсов и данных от несанкционированного использования и разрешает конфликтные ситуации.

# ОС как расширенная виртуальная машина

**6. Обнаружение ошибок и их обработка.** При работе компьютерной системы могут происходить разнообразные сбои за счет внутренних и внешних ошибок в аппаратном обеспечении, различного рода программных ошибок (переполнение, попытка обращения к ячейке памяти, доступ к которой запрещен и др.). В каждом случае ОС выполняет действия, минимизирующие влияние ошибки на работу приложения (от простого сообщения об ошибке до аварийной остановки программы).

**7. Учет использования ресурсов.** Хорошая ОС имеет средства учета использования различных ресурсов и отображения параметров производительности вычислительной системы. Эта информация важна для настройки (оптимизации) вычислительной системы с целью повышения ее производительности

Таким образом, назначение ОС состоит в предоставлении пользователю (программисту) некоторой расширенной виртуальной машины, которую легче программировать и с которой легче работать, чем непосредственно с аппаратурой.

# ОС как менеджер ресурсов

ОС не только представляет пользователям и программистам удобный интерфейс к аппаратным средствам компьютера, но и является своеобразным диспетчером ресурсов компьютера. К числу основных ресурсов современных вычислительных систем относятся процессоры, основная память, таймеры, наборы данных, диски, накопители на магнитных лентах (МЛ), внешние накопители памяти (CD/DVD/Blu-Ray/USB), принтеры, сетевые устройства и др.

Управление ресурсами может происходить:

- во времени (например, выделение процессорного времени);
- в пространстве (например, запись файла на жёсткий диск).



# ОС как менеджер ресурсов

Управление ресурсами включает решение ряда общих, не зависящих от типа ресурса задач:

- планирование ресурса – определение, какому процессу, когда и в каком качестве (если ресурс может выделяться частями) следует выделить данный ресурс;
- удовлетворение запросов на ресурсы – выделение ресурса процессам;
- отслеживание состояния и учет использования ресурса – поддержание оперативной информации о занятости ресурса и распределенной его доли;
- разрешение конфликтов между процессами, претендующими на один и тот же ресурс.

В отличие от функций расширенной виртуальной машины большинство функций управления ресурсами выполняются операционной системой автоматически и прикладному программисту недоступны.

# ОС как менеджер ресурсов

ОС облегчает процессы эксплуатации аппаратных и программных средств вычислительной системы. Ряд операционных систем имеет в своем составе наборы служебных программ, обеспечивающие резервное копирование, архивацию данных, проверку, очистку и дефрагментацию дисковых устройств и др.

Кроме того, современные ОС имеют достаточно большой набор средств и способов диагностики и восстановления работоспособности системы. Сюда относятся:

- диагностические программы для выявления ошибок в конфигурации ОС;
- средства восстановления последней работоспособной конфигурации;
- средства восстановления поврежденных и пропавших системных файлов и др.

# ОС как менеджер ресурсов

Наиболее важными подсистемами управления ресурсами являются подсистемы управления процессами, памятью, файлами и внешними устройствами, а подсистемами, общими для всех ресурсов, являются подсистемы пользовательского интерфейса, защиты данных и администрирования.

**Управление процессами.** Подсистема управления процессами непосредственно влияет на функционирование вычислительной системы. Для каждой выполняемой программы ОС организует один или более процессов. Каждый такой процесс представляется в ОС информационной структурой (таблицей, дескриптором, контекстом процессора), содержащей данные о потребностях процесса в ресурсах, а также о фактически выделенных ему ресурсах (область оперативной памяти, количество процессорного времени, файлы, устройства ввода-вывода и др.). Кроме того, в этой информационной структуре хранятся данные, характеризующие историю пребывания процесса в системе: текущее состояние (активное или заблокированное), приоритет, состояние регистров, программного счетчика и др.

# ОС как менеджер ресурсов

**Управление памятью.** Подсистема управления памятью производит распределение физической памяти между всеми существующими в системе процессами, загрузку и удаление программных кодов и данных процессов в отведенные им области памяти, настройку адресно-зависимых частей кодов процесса на физические адреса выделенной области, а также защиту областей памяти каждого процесса. Стратегия управления памятью складывается из стратегий выборки, размещения и замещения блока программы или данных в основной памяти.

Одним из наиболее популярных способов управления памятью в современных ОС является виртуальная память.

Важная функция управления памятью – защита памяти. Нарушения защиты памяти связаны с обращениями процессов к участкам памяти, выделенной другим процессам прикладных программ или программ самой ОС. Средства защиты памяти должны пресекать такие попытки доступа.

# ОС как менеджер ресурсов

**Управление файлами.** Функции управления файлами сосредоточены в файловой системе ОС. Операционная система виртуализирует отдельный набор данных, хранящихся на внешнем накопителе, в виде файла – простой неструктурированной последовательности байтов, имеющих символьное имя. Для удобства работы с данными файлы группируются в каталоги, которые, в свою очередь, образуют группы – каталоги более высокого уровня. Файловая система преобразует символьные имена файлов, с которыми работает пользователь или программист, в физические адреса данных на дисках, организует совместный доступ к файлам, защищает их от несанкционированного доступа.

# ОС как менеджер ресурсов

**Управление внешними устройствами.** Функции управления внешними устройствами возлагаются на подсистему управления внешними устройствами, называемую также подсистемой ввода-вывода. Она является интерфейсом между ядром компьютера и всеми подключенными к нему устройствами. Спектр этих устройств очень обширен (принтеры, сканеры, мониторы, модемы, манипуляторы, сетевые адаптеры, АЦП разного рода и др.), сотни моделей этих устройств отличаются набором и последовательностью команд, используемых для обмена информацией с процессором и другими деталями.

Программа, управляющая конкретной моделью внешнего устройства и учитывающая все его особенности, называется драйвером. Созданием драйверов занимаются как разработчики ОС, так и компании, выпускающие внешние устройства. ОС должна поддерживать четко определенный интерфейс между драйверами и остальными частями ОС.

# ОС как менеджер ресурсов

**Защита данных и администрирование.** Безопасность данных вычислительной системы обеспечивается средствами отказоустойчивости ОС, направленными на защиту от сбоев и отказов аппаратуры и ошибок программного обеспечения, а также средствами защиты от несанкционированного доступа. Для каждого пользователя системы обязательна процедура логического входа, в процессе которой ОС убеждается, что в систему входит пользователь, разрешенный административной службой. Администратор вычислительной системы определяет и ограничивает возможности пользователей в выполнении тех или иных действий.

Важным средством защиты являются функции аудита ОС, заключающиеся в фиксации всех событий, от которых зависит безопасность системы.

Поддержка отказоустойчивости вычислительной системы реализуется на основе резервирования (дисковые RAID-массивы, резервное копирование и пр.)

# ОС как менеджер ресурсов

**Интерфейс прикладного программирования.** Прикладные программисты используют в своих приложениях обращения к операционной системе, когда для выполнения тех или иных действий им требуется особый статус, которым обладает только ОС. Возможности операционной системы доступны программисту в виде набора функций, который называется интерфейсом прикладного программирования (Application Programming Interface, API). Приложения обращаются к функциям API с помощью системных вызовов. Способ, которым приложение получает услуги операционной системы, очень похож на вызов подпрограмм.

Способ реализации системных вызовов зависит от структурной организации ОС, особенностей аппаратной платформы и языка программирования.



# ОС как менеджер ресурсов

**Пользовательский интерфейс.** ОС обеспечивает удобный интерфейс не только для прикладных программ, но и для пользователя (программиста, администратора).

Современные ОС поддерживают развитые функции пользовательского интерфейса для интерактивной работы за терминалами двух типов: алфавитно-цифрового и графического.

При работе за алфавитно-цифровым терминалом пользователь имеет в своем распоряжении систему команд, развитость которой отражает функциональные возможности данной ОС. Обычно командный язык ОС позволяет запускать и останавливать приложения, выполнять различные операции с каталогами и файлами, получать информацию о состоянии ОС, администрировать систему. Команды могут вводиться не только в интерактивном режиме с терминала, но и считываться из так называемого командного файла, содержащего некоторую последовательность команд.

Программный модуль ОС, ответственный за чтение отдельных команд или же последовательности команд из командного файла, называют командным интерпретатором.

# Особенности ОС реального времени

Существует несколько определений систем реального времени (СРВ) (real time operating systems (RTOS)), большинство из которых противоречит друг другу.

Например, системой реального времени называется система, в которой успешность работы любой программы зависит не только от ее логической правильности, но и от времени, за которое она получила результат. Если временные ограничения не удовлетворены, то фиксируется сбой в работе систем.

Примером задачи, где требуется СРВ, является управление роботом, берущим деталь с ленты конвейера.

Различают сильное (hard) и слабое (soft) требование реального времени. Если запаздывание программы приводит к полному нарушению работы управляемой системы, то говорят о сильном реальном времени (жесткие СРВ). Если же запаздывание ведет только к потере производительности, то говорят о слабом реальном времени (мягкие СРВ).

В СРВ необходимо введение некоторого директивного срока (в англоязычной литературе – deadline), до истечения которого задача должна обязательно (для систем мягкого реального времени – желательно) выполняться.

# Особенности ОС реального времени

## Признаки систем жесткого реального времени:

- . недопустимость никаких задержек, ни при каких условиях;
- . бесполезность результатов при опоздании;
- . катастрофа при задержке реакции;
- . цена опоздания бесконечно велика.

Пример системы жесткого реального времени — бортовая система управления самолетом.

## Признаки систем мягкого реального времени:

- . за опоздание результатов приходится платить;
- . снижение производительности системы, вызванное запаздыванием реакции на происходящие события.

Пример — автомат розничной торговли или подсистема сетевого интерфейса.

# Особенности ОС реального времени

Не существует операционных систем жесткого или мягкого реального времени. Понятия системы реального времени и операционной системы реального времени (ОСРВ) часто смешиваются.

**Система реального времени** - это конкретная система, связанная с реальным объектом. Она включает в себя необходимые аппаратные средства, операционную систему и прикладное программное обеспечение.

**Операционная система реального времени** – это только инструмент, помогающий построить конкретную систему реального времени. Поэтому бессмысленно говорить об операционных системах жесткого или мягкого реального времени.

Для создания системы жесткого реального времени необходимо сочетание подходящих аппаратных средств, адекватной операционной системы и правильного проектирования прикладного программного обеспечения.

# Особенности ОС реального времени

Если, например, принято решение построить систему реального времени, обслуживающую TCP/IP-соединение через Ethernet, то система никогда не будет системой жесткого реального времени, поскольку сам Ethernet непредсказуем. В данном случае, основное ограничение на создание СРВ оказывает метод случайного доступа CSMA/CD.

Если, с другой стороны, вы создаете приложение над такой ОС, как "Windows 3.11", то ваша система никогда не будет системой жесткого реального времени, поскольку непредсказуемо поведение операционной системы.

# Особенности ОС реального времени

Стандарт POSIX 1003.1 определяет СРВ следующим образом: «Реальное время в операционных системах — это способность операционной системы обеспечить требуемый уровень сервиса в заданный промежуток времени».

Иногда системами реального времени называют системы постоянной готовности (on-line системы), или «интерактивные системы с достаточным временем реакции». Обычно это делают фирмы-производители по маркетинговым соображениям. Если интерактивную программу называют работающей в реальном времени, то это означает, что она успевает обрабатывать запросы от человека, для которого задержка в сотни миллисекунд незаметна.

Часто понятие «система реального времени» отождествляют с понятием «быстрая система». Это не всегда правильно.

# Особенности ОС реального времени

Согласно определению, СРВ должна «обеспечить требуемый уровень сервиса в заданный промежуток времени». Этот промежуток времени обычно задается периодичностью и скоростью процессов, которыми управляет система.

Типичные времена реакции на внешние события в процессах, управляемых СРВ:

- радиолокация - несколько миллисекунд;
- складской учет - несколько секунд;
- торговые операции - несколько минут;
- управление производством - несколько минут.

Интервал между поступлениями сообщений в ЭВМ может быть случайным и определяться внешними факторами, такими, как нажатие клавиши оператором, или он может быть циклическим и управляться от часов или от сканирующего механизма в ЭВМ.

# Особенности ОС реального времени

**Определение.** Система реального времени реагирует в предсказуемое время на непредсказуемое появление внешних событий.

Это определение предъявляет к системе вполне определенные базовые требования.

Мартин Тиммерман сформулировал следующие необходимые требования для ОСРВ:

- ОС должна быть многозадачной и допускающей вытеснение (preemptable),
- ОС должна обладать понятием приоритета для потоков,
- ОС должна поддерживать предсказуемые механизмы синхронизации,
- ОС должна обеспечивать механизм наследования приоритетов,
- поведение ОС должно быть известным и предсказуемым (задержки обработки прерываний, задержки переключения задач, задержки драйверов и т.д.); это значит, что во всех сценариях рабочей нагрузки системы должно быть определено максимальное время отклика.



# Особенности ОС реального времени

**Требование 1:** ОСРВ должна быть многопоточной или многозадачной и поддерживать диспетчеризацию с вытеснением.

Поведение ОСРВ должно быть предсказуемым. Это не означает, что ОСРВ должна быть быстрой, но означает, что максимальный промежуток времени для выполнения любой операции должен быть известен заранее и должен быть согласован с требованиями приложения.

Планировщик должен иметь возможность вытеснять любую нить (задачу) и передавать управление той нити (задаче), которая больше всего в этом нуждается.

# Особенности ОС реального времени

**Требование 2:** Должно существовать понятие приоритета нити (задачи).

При построении конкретной системы реального времени разработчик должен выстроить приоритеты задач таким образом, чтобы каждая из них успела с реакцией к своему критическому сроку, то есть он должен трансформировать базовое требование реального времени "успеть с реакцией к нужному моменту" в комбинацию приоритетов и в сценарий их динамического изменения.

Очевидно, что при этой трансформации возможны ошибки, приводящие к неправильной работе системы.

Как бы то ни было, во всех современных ОСРВ приходится использовать механизм приоритетов как один из инструментов предсказуемости поведения системы. На сегодня не имеется другого решения.

# Особенности ОС реального времени

**Требование 3:** ОС должна поддерживать предсказуемые механизмы синхронизации нитей (задач). Все нити (задачи) разделяют данные (ресурсы) и должны обмениваться между собой информацией, поэтому необходимы механизмы межзадачного (межнитиевого) взаимодействия.

Синхронизационные механизмы подразделяют на следующие основные категории:

- универсальные для низкого уровня — можно использовать различными способами (семафоры);
- простые для низкого уровня — каждый из которых приспособлен к решению только одной задачи (мьютексы и условные переменные);
- универсальные для высокого уровня, выраженные через простые — к этой группе относится концепция монитора, которая может быть выражена через мьютексы и условные переменные;
- простые для высокого уровня — приспособленные к решению конкретной синхронизационной задачи (блокировка чтения-записи и барьеры).

# Особенности ОС реального времени

Синхронизационные механизмы обеспечивают разделение **критических ресурсов**.

Те места в программах, в которых происходит обращение к критическим ресурсам, называются **критическими секциями**. Решение этой проблемы заключается в организации такого доступа к критическому ресурсу, когда только одному процессу разрешается входить в критическую секцию.

Ресурсы, которые не допускают одновременного использования несколькими процессами, называются **критическими**. Если нескольким вычислительным ресурсам необходимо пользоваться критическим ресурсом в режиме разделения, им следует синхронизировать свои действия таким образом, чтобы ресурс всегда находился в распоряжении не более чем одного из процессов.

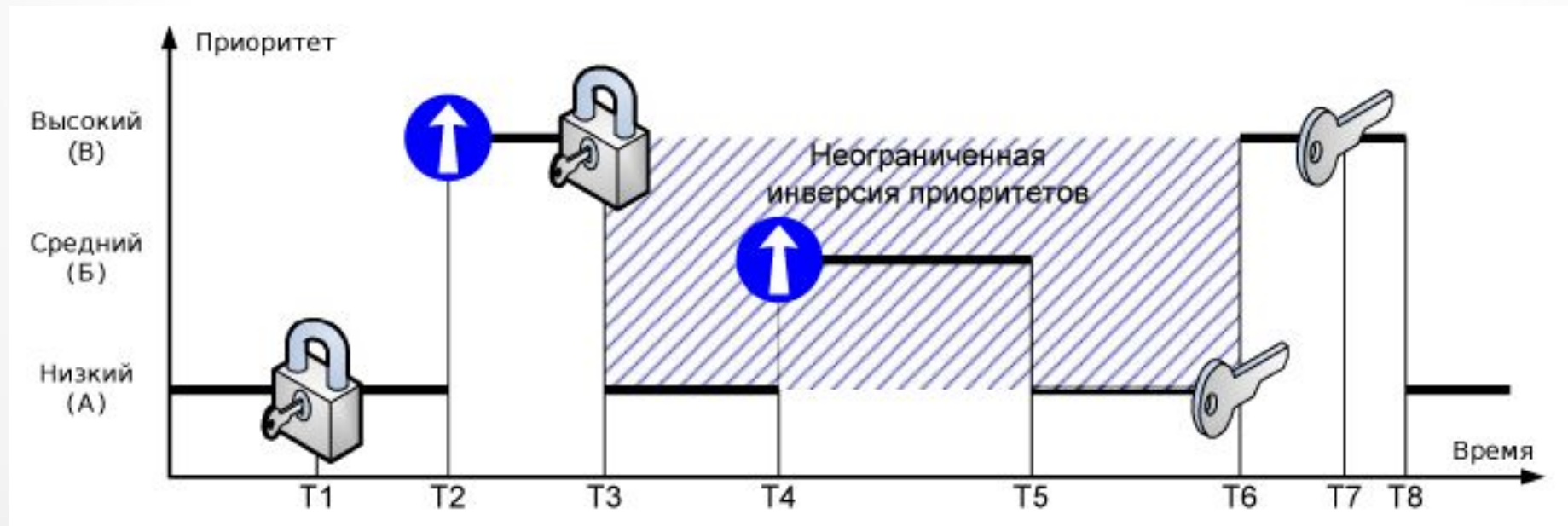
# Особенности ОС реального времени

**Требование 4:** Должен существовать механизм наследования приоритетов (система должна быть защищена от инверсии приоритетов).

Инверсия приоритета может произойти, когда многие потоки с различным приоритетом имеют общий доступ к одним и тем же блокировкам и ресурсам, и поток с низким приоритетом практически до бесконечности задерживает работу потока с более высоким приоритетом.

# Особенности ОС реального времени

Если ресурс заблокирован задачей (А), а он требуется задаче (В), то высокоприоритетная задача блокируется. Но допустим, что задача (Б) вытеснила (А), после того как (В) ушла в ожидание ресурса. Задача (Б) ничего не знает о конфликте, поэтому может выполняться сколь угодно долго на промежутке времени (Т5-Т4). Кроме того, кроме (Б) в системе могут быть и другие задачи, с приоритетами больше (А), но меньше (В). Поэтому длительность периода (Т6-Т3) в общем случае не определена. Такую ситуацию называют **неограниченной инверсией приоритетов**.

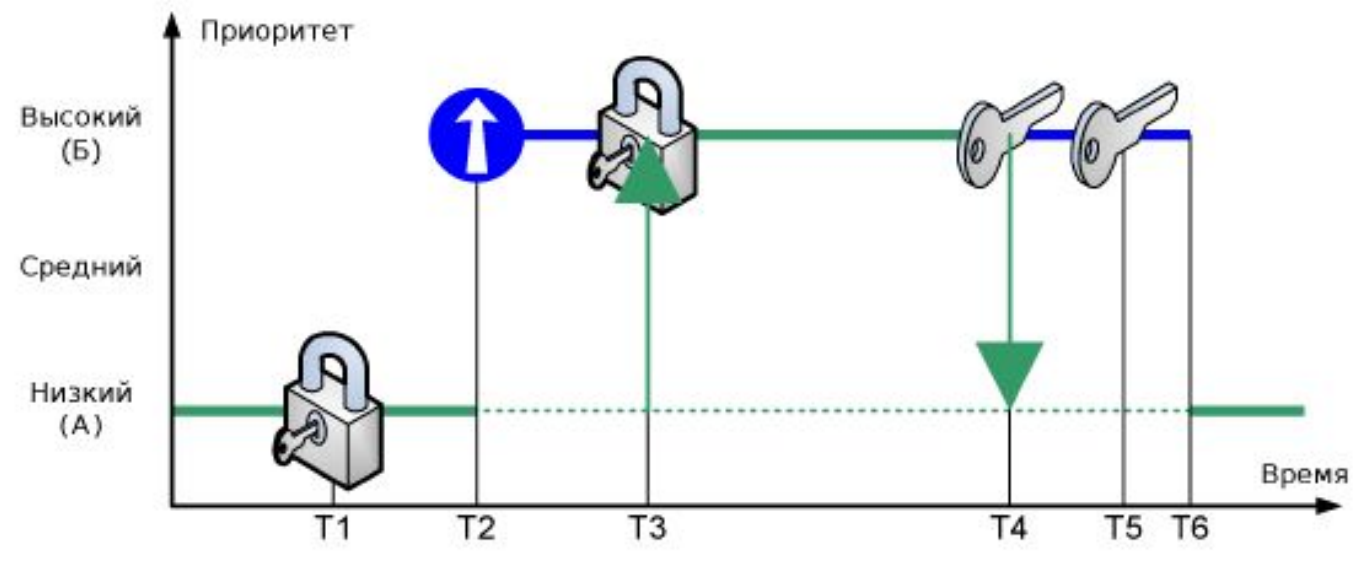


# Особенности ОС реального времени

ОСРВ должна быть снабжена механизмом **наследования приоритетов**, то есть блокирующая нить должна наследовать приоритет нити, которую она блокирует (конечно, только, в том случае, если заблокированная нить имеет более высокий приоритет). То есть поведение ОС должно быть предсказуемо.

Допустим в системе существуют две задачи с низким (А) и высоким (Б) приоритетом:

В момент T2 задача (Б) вытесняет низкоприоритетную задачу (А) и затем в момент времени T3 пытается захватить заблокированный (А) ресурс. Протокол наследования приоритета состоит в том, что приоритет задачи (А) повышается до приоритета задачи (Б) в момент времени T3, то есть когда (Б) пытается захватить заблокированный ресурс.



# Особенности ОС реального времени

В качестве основного требования к ОСРВ выдвигается требование обеспечения **предсказуемости** или **детерминированности поведения** системы в наихудших внешних условиях, что резко отличается от требований к производительности и быстродействию универсальных ОС. Хорошая ОСРВ имеет предсказуемое поведение при всех сценариях системной загрузки (одновременные прерывания и выполнение потоков).

Существует различие между **системами реального времени** и встроенными системами. От встроенной системы не всегда требуется, чтобы она имела предсказуемое поведение, и в таком случае она не является системой реального времени.

Однако большинство встроенных систем нуждается в предсказуемом поведении, по крайней мере, для некоторой функциональности, и таким образом, эти системы можно отнести к системам реального времени.



# Особенности ОС реального времени

## Требования, накладываемые на вычислительную установку реального времени.

1. В зависимости от сложности программы управления, требование «реального времени» накладывает различные условия на **вычислительную мощность** процессора для СРВ.
2. Часто для ОСРВ более важна не мощность процессора, а характеристики компьютера, связанные с **подсистемой прерываний** (interrupt requests, IRQ). Желательными являются:
  - наличие как можно большего количества уровней прерываний (IRQ levels) (т.е. аппаратного или/и программного декодирования источника запроса);
  - как можно меньшее время реакции на прерывание (т.е. как можно меньшее время между поступлением запроса на обслуживание и началом выполнения обслуживающей программы).

# Особенности ОС реального времени

3. СРВ часто сама является инициатором периодических процессов, которыми управляет (например, движением космического аппарата или луча радара).

Поэтому необходимо иметь в наличии один или несколько **таймеров** (аппаратных устройств, выдающих прерывание через заданные промежутки времени), которые могут работать в периодическом или ждущем режиме.

4. Ввиду того, что СРВ часто управляет ответственными промышленными процессами, данное обстоятельство выдвигает **очень жесткие требования к надежности** используемого оборудования.

# Особенности ОС реального времени

Понятие «системы реального времени» является новым и в полной мере не устоявшимся. Однако основной мыслью существующих определений является наложение жестких или мягких ограничений на **время отклика системы при поступлении на нее внешнего воздействия**.

Отсюда критически важным параметром для СРВ является время ее реакции на прерывания (которое складывается из аппаратного времени задержки и программных задержек), а также предсказуемость этого времени.

Нельзя считать системами реального времени информационно-управляющие системы, не предусматривающие в алгоритмах своего функционирования возможность выдачи откликов в условиях лимита времени. Поэтому организация работы информационно-управляющих систем, функционирующих в режиме реального времени, существенно отличается от работы традиционных систем управления.

# Области применения систем реального времени

В течение длительного времени основными потребителями СРВ были военная и космическая области. Сейчас ситуация кардинально изменилась и СРВ можно встретить даже в товарах широкого потребления. Рассмотрим основные области применения СРВ.

## **Военная и космическая области:**

- бортовое и встраиваемое оборудование;
- системы измерения и управления, радары;
- цифровые видеосистемы, симуляторы;
- ракеты, системы определения положения и привязки к местности.

# Области применения систем реального времени

## **Промышленность и финансы:**

- автоматические системы управления производством (АСУП), автоматические системы управления технологическим процессом (АСУТП);
- автомобилестроение: симуляторы, системы управления двигателем, автоматическое сцепление, системы антиблокировки колес и т.д.;
- энергетика: сбор информации, управление данными и оборудованием;
- телекоммуникации: коммуникационное оборудование, сетевые коммутаторы, телефонные станции и т.д.;
- банковское оборудование (например, во многих банкоматах работает CPB QNX).

# Области применения систем реального времени

## Товары широкого потребления:

- мобильные телефоны (например, в телефонах стандарта GSM работает СРВ рSOS);
- цифровые телевизионные декодеры;
- цифровое телевидение (мультимедиа, видеосерверы);
- компьютерное и офисное оборудование (принтеры, копиры), например, в факсах применяется СРВ VxWorks, в устройствах чтения компакт-дисков – СРВ VRTX32.

# Аппаратурная среда систем реального времени

Систему реального времени можно разделить как бы на три слоя:

1. **Ядро** — содержит только строгий минимум, необходимый для работы системы: управление задачами, их синхронизация и взаимодействие, управление памятью и устройствами ввода/вывода. Размер ядра очень ограничен: часто несколько килобайт.

2. **Система управления** — содержит ряд дополнительных сервисов, расширяющих возможности ядра: расширенное управление памятью, вводом/выводом, задачами, файлами и т.д., обеспечивает также взаимодействие системы и управляющего/управляемого оборудования.

3. **Система реального времени** — содержит систему управления и набор утилит: средства разработки (компиляторы, отладчики и т.д.), средства визуализации (взаимодействия человека и операционной системы).

# Аппаратурная среда систем реального времени

Вычислительные установки, на которых применяются СРВ, можно условно разделить на три группы.

1. **«Обычные» компьютеры.** По логическому устройству совпадают с настольными системами. Аппаратное устройство несколько отличается. Для обеспечения минимального времени простоя в случае технической неполадки процессор, память и т.д. размещены на съемной плате.

Сам компьютер помещается в специальный защищённый корпус.

Среди процессоров этих компьютеров доминирует семейство Intel 80x86.

Подобные вычислительные системы обычно служат как терминалы для взаимодействия с промышленными компьютерами и встроенными контроллерами. В качестве операционных систем часто используются «обычные» операционные системы с дополнительными программными комплексами, адаптирующими их к требованиям «реального времени».



# Аппаратурная среда систем реального времени

**2. Промышленные компьютеры.** Состоят из одной платы, на которой размещены: процессор, контроллер памяти, память 4-х видов:

- ПЗУ, постоянное запоминающее устройство (ROM, read-only memory), где обычно размещена сама операционная система реального времени; типичная емкость – 0,5-1 Мб;
- ОЗУ, оперативное запоминающее устройство (RAM, random access memory), куда загружается код и данные ОСРВ; обычно организована на базе динамической памяти (dynamic RAM, DRAM); типичная емкость – 16-128 Мб;
- статическое ОЗУ (static RAM, SRAM) (то же, что и ОЗУ, но питается от имеющейся на плате батарейки), где размещаются критически важные данные, которые не должны пропадать при выключении питания; типичная емкость - 2Мб; типичное время сохранения данных - 5 лет;
- флеш-память (flash RAM) (электрически программируемое ПЗУ), которая играет роль диска для ОСРВ; типичная емкость - 4Мб.

# Аппаратурная среда систем реального времени

Контроллеры периферийных устройств: SCSI (Small Computer System Interface), Ethernet, COM портов, параллельного порта, несколько программируемых таймеров. На плате находится также контроллер и разъем шины, через которую компьютер управляет внешними устройствами. В качестве шины в подавляющем большинстве случаев используется шина VME, которую в последнее время стала теснить шина Compact PCI.

Несмотря на наличие контроллера SCSI, обычно ОСПВ работает без дисковых накопителей, поскольку они не удовлетворяют предъявляемым к системам реального времени требованиям по надежности, устойчивости к вибрации, габаритам и времени готовности после включения питания. В специальный корпус (крейт) вставляются платы аналого-цифровых и/или цифро-аналоговых преобразователей (АЦП и/или ЦАП), через которые осуществляется ввод/вывод управляющей информации, платы управления электромоторами.

# Аппаратурная среда систем реального времени

Среди процессоров промышленных компьютеров доминируют процессоры семейств Power PC (Motorola IBM) и Motorola 68xxx (Motorola). Также присутствуют процессоры семейств SPARC (SUN), Intel 80x86 (Intel), ARM (ARM), Intel 80960x (Intel).

При выборе процессора определяющими факторами являются получение требуемой производительности при наименьшей тактовой частоте, а, значит, и наименьшей рассеиваемой мощности, а также наименьшее время переключения задач и реакции на прерывания.

Требование малой рассеиваемой мощности процессора важно с точки зрения получения высокой отказоустойчивости системы в целом, поскольку слабый нагрев процессора позволяет обойтись без охлаждающего вентилятора.

Промышленные компьютеры используются для непосредственного управления промышленным или иным оборудованием. Они часто не имеют монитора и клавиатуры, и для взаимодействия с ними служат «обычные» компьютеры, соединенные с ними через последовательный порт (СОМ порт) или Ethernet.

# Аппаратурная среда систем реального времени

**3. Встраиваемые системы.** Устанавливаются внутрь оборудования, которым они управляют. Для крупного оборудования (например, ракета или космический аппарат) могут по исполнению совпадать с промышленными компьютерами.

Для оборудования поменьше (например, принтер) могут представлять собой процессор с сопутствующими элементами, размещенный на одной плате с другими электронными компонентами этого оборудования.

Для миниатюрного оборудования (например, мобильный телефон) процессор с сопутствующими элементами может быть частью одной из больших интегральных схем этого оборудования.

# Аппаратурная среда систем реального времени

Система ОСРВ часто должна работать **на бездисковом компьютере** и осуществлять начальную загрузку из ПЗУ. В силу этого:

- критически важным является размер системы;
- для экономии места в ПЗУ часть системы может храниться в сжатом виде и загружаться в ОЗУ по мере необходимости;
- система часто позволяет исполнять код как в ОЗУ, так и в ПЗУ;
- при наличии свободного места в ОЗУ система часто копирует себя из медленного ПЗУ в более быстрое ОЗУ;
- сама система компилируется, линкуется и превращается в загрузочный модуль на другом, «обычном» компьютере, связанном с промышленным компьютером через последовательный порт или Ethernet. Это требует специального кроссплатформенного инструментария разработчика, поскольку типы процессоров и/или операционных систем на этих двух компьютерах не совпадают.