Лекция 1. Основные понятия и определения ОСРВ

1.1 Виды современных систем

В настоящее время получили широкое распространение так называемые "системы с непосредственной связью" (on line) и "системы, работающие в реальном времени" (real-time).

В этих системах данные от внешних объектов, с которыми работает ЭВМ, могут вводиться непосредственно в вычислительную машину, и информация из ЭВМ посылается обратно этим объектам. Множество разнообразных устройств которые посылают данные к ЭВМ и получают обработанную информацию, называются ***терминалами***.

Терминалы в такой системе могут быть спроектированы для работы с коммерческими данными или могут представлять собой разнообразное технической оборудование, например, термопары, тензометры и т. д. Очень широкий диапазон приборов может применяться для сбора данных из мест их возникновения и для передачи результатов вычислений на места, где они нужны. Сеть терминалов является характерной для системы с непосредственной связь.

Система с непосредственной связью может быть определена как система, в которой входные данные вводятся в ЭВМ непосредственно из точки их возникновения и/или, в которой выходные данные передаются непосредственно туда, где они используются.

Вычислительная система вместо того, чтобы делать работу, результаты которой использовались бы позднее, может теперь в любую минуту вступать в непосредственный контакт с внешними объектами, управляя ими. Она может планировать работу фабрики и перепланировать ее в случае появления новых требований или при изменении ситуации на рынке сбыта. Это и есть работа в реальном времени.

Вычислительная система, работающая в реальном времени, может быть определена как система, которая управляет внешними объектами, получая информацию, обрабатывая ее и возвращая результаты достаточно быстро для того, чтобы воздействовать на функционирование внешних объектов в почти тот же момент времени.

В своём развитии системы строились на основе следующих архитектур:

1 Монолитная архитектура. ОС определяется как набор модулей, взаимодействующих между собой внутри ядра системы и предоставляющих прикладному ПО входные интерфейсы для обращений к аппаратуре. Основной недостаток этого принципа построения ОС заключается в плохой предсказуемости её поведения, вызванной сложным взаимодействием модулей между собой.

2 Уровневая (слоевая) архитектура. Прикладное ПО имеет возможность получить доступ к аппаратуре не только через ядро системы и её сервисы, но и напрямую. По сравнению с монолитной такая архитектура обеспечивает значительно большую степень предсказуемости реакций системы, а также позволяет осуществлять быстрый доступ прикладных приложений к аппаратуре. Главным недостатком таких систем является отсутствие многозадачности.

3 Архитектура «клиент-сервер». Основной её принцип заключается в вынесении сервисов ОС в виде серверов на уровень пользователя и выполнении микроядром функций диспетчера сообщений между клиентскими пользовательскими программами и серверами — системными сервисами.

Преимущества такой архитектуры:

1. Повышенная надёжность, так как каждый сервис является, по сути, самостоятельным приложением и его легче отладить и отследить ошибки.

2. Улучшенная масштабируемость, поскольку ненужные сервисы могут быть исключены из системы без ущерба к её работоспособности.

3. Повышенная отказоустойчивость, так как «зависший» сервис может быть перезапущен без перезагрузки системы.

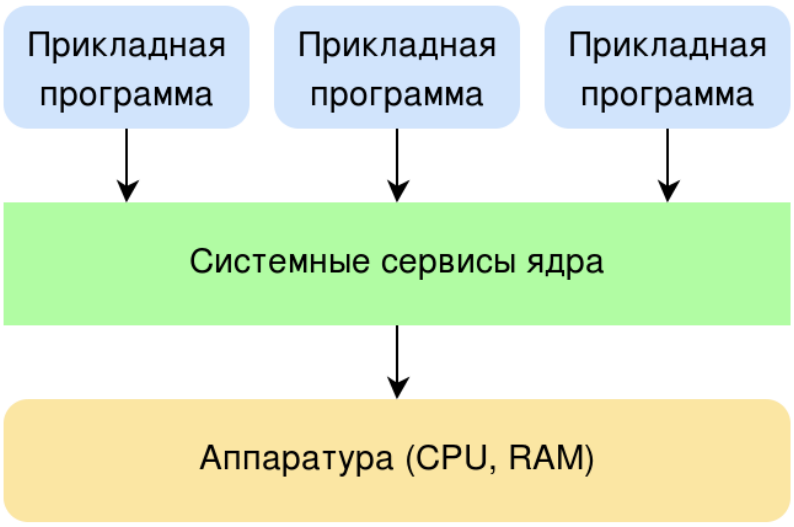

Рисунок 1 - Монолитная архитектура

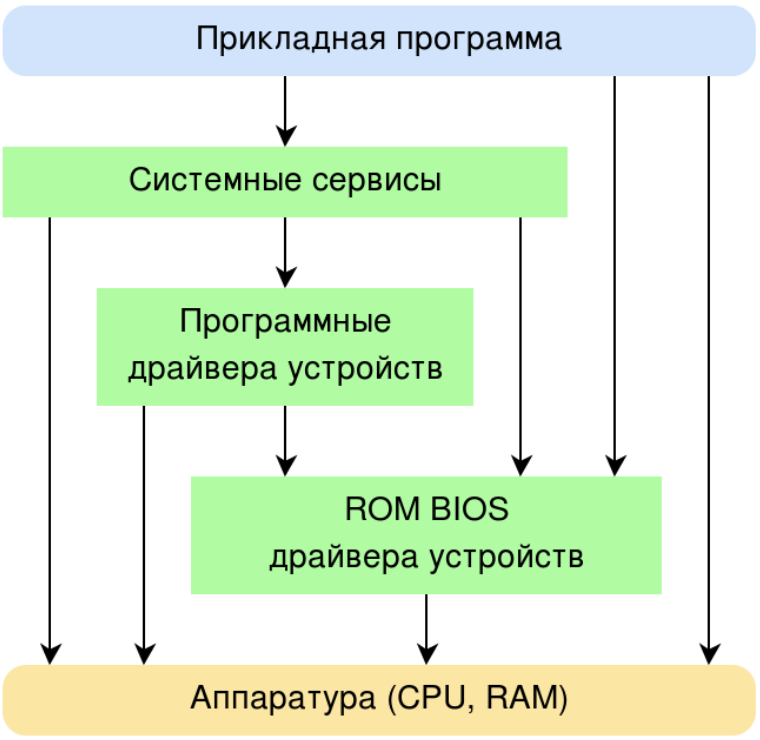

Рисунок 2 - Уровневая (слоевая) архитектура

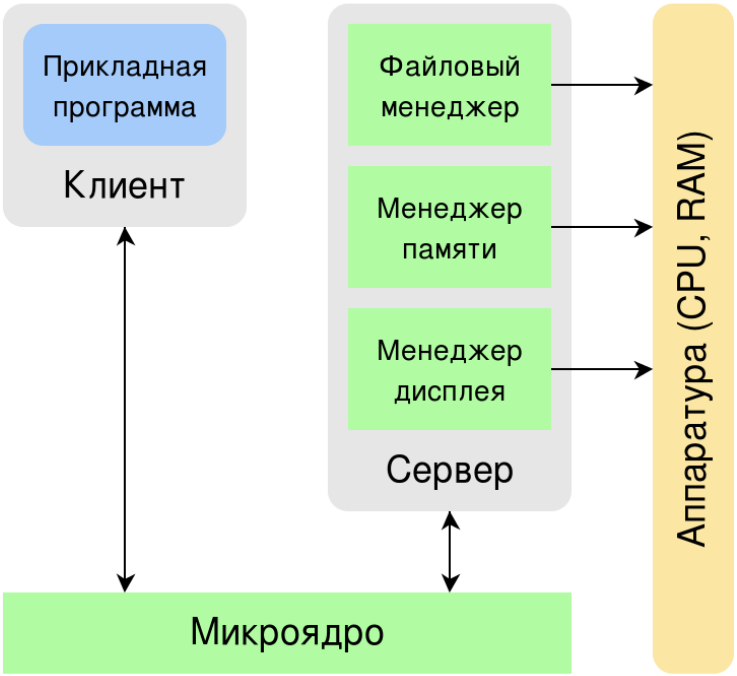

Рисунок 3 - Архитектура «клиент–сервер»

1.2 Использование реального масштаба времени

Области применения систем реального времени можно разделить на две категории.

1. ***К первой*** относятся те работы, которые раньше выполнялись вручную или с помощью машин и могли бы выполняться более эффективно или более экономично (что, в конечном счете, означает одно и тоже) системами с непосредственной, связью, работающими в реальном времени.
2. ***Ко второй***, и гораздо более интересной категории, относятся применения новых систем для таких работ, которые раньше выполнять было совершенно невозможно.

Под **реальным временем** понимается количественная характеристика, которая может быть измерена реальными физическими часами, в отличие от логического времени, определяющего лишь качественную характеристику, выражаемую относительным порядком следования событий.

Говорят, что система работает в режиме реального времени, если для описания работы этой системы требуются количественные временные характеристики.

Процессы (задачи) систем реального времени могут иметь следующие характеристики и связанные с ними ограничения:

- **дедлайн (deadline)** — критический срок обслуживания, предельный срок завершения какой-либо работы;
- **латентность (latency)** — время отклика (время задержки) системы на внешние события;
- **джиттер (jitter)** — разброс значений времени отклика. Можно различить **джиттер запуска (release jitter)** — период времени от готовности к исполнению до начала собственно исполнения задачи и **джиттер вывода (output jitter)** — задержка по окончании выполнения задачи.

На промышленном предприятии большинство входных данных для учета и расчетов поступают из цехов предприятия. Они содержат такую информацию как время наладки станков, время начала и окончания работ, количество отходов и т.д. Для получения этой информации могут использоваться системы сбора данных. При сборе информации в системе с непосредственной связью значительно снижается объем канцелярской работы, уменьшается количество ошибок, и в ЭВМ накапливается обновляемая информация о состоянии дел в цехе. Эта информация может использоваться для ***подсчета издержек***. В более совершенных системах она может использоваться для ***непрерывного контроля состояния производства***.

Для достижения цели, заключающейся в максимизации прибыли, руководство производства должно устанавливать план, обеспечивать его выполнение и затем оценивать его. Чтобы контролировать и минимизировать затраты, нужна непрерывная их оценка с принятием срочных мер при их повышении. Также и планы производства должны немедленно корректироваться при поступлении новых заказов. Любое изменение в объеме работ или в имеющихся ресурсах должно вести к корректировке графиков работ. Чтобы быть конкурентоспособной, фирме надо принимать решения быстро, и эти решения должны основываться на последних данных. Система реального времени, которая содержит всю эту информацию, дает возможность руководству принимать необходимые решения быстро и безошибочно.

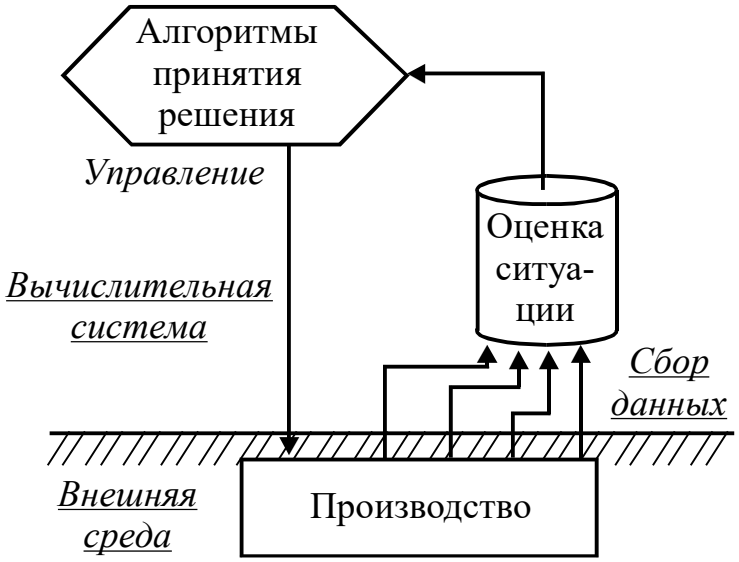

Рисунок 4 -Система управления производством в реальном времени

Работая таким образом, ЭВМ выполняет ***функции управления***. Вместо своей старой функции обработки некоторого объема данных или выполнения ряда вычислений, которые не зависят от времени, машина теперь управляет объектом, и время при этом является очень важным фактором. Она непрерывно получает информацию с производства, моделирует ситуацию, применяет набор правил решения к этой ситуации и посылает команды мастеру или оператору. Таким образом, ЭВМ оптимально управляет происходящим процессом (рисунок 1).

Практически система обработки данных, кроме диспетчеризации производства, могла бы выполнять многие другие функции, связанные с управлением работой предприятия. Это могут быть учет товаров на складах, контроль закупок, планирование запасов деталей, материалов и рабочей силы, оценка работ, расчеты стоимости, ведение счетов и т. д.

Основным фактором является потребность в данных о текущем состоянии дел на промышленном объекте. На основе полученных данных выполняется вычисления или вырабатываются решения, и ЭВМ может или непосредственно управлять производством, или давать команды операторам. Таких же результатов нельзя достичь посредством усилий человека, так как необходима высокая скорость вычислений и выработки логических решений.

Если система реального времени дает ежеминутную оценку ситуации, то возникает вопрос, является ли это достаточной платой за стоимость ЭВМ? Если система, дает руководству своевременную информацию или уменьшает время планирования работы предприятия, то сколько это может стоить? Если система реального времени ускоряет работу или улучшает обслуживание клиента, то будет ли это компенсировать стоимость ЭВМ?

Традиционная оценка систем обработки данных обычно базировалась на понятии оборота денежных средств, однако в настоящее время значительные улучшения экономических показателей основываются на так называемых неосязаемых прибылях руководства. Вычислительные системы не могут больше оцениваться старыми мерками. Система реального времени не означает только сокращения штата, она должна оцениваться по увеличению, эффективности производства. Это может быть информационная система, которая помогает

управлением предприятия в меняющейся обстановке.

Целью системы реального времени является объединение различных решение или действий таким образом, чтобы некоторый процесс протекал наиболее эффективно.

Система может разрабатываться для того, чтобы

- уменьшить цикли планирования производства,
- максимизировать оборот,
- добиться лучшего использования возможностей организации
- и своевременно поставить информацию туда, где она требуется.

При проектировании системы и планировании программ необходимо учитывать стоимости обеспечения различного времени реакции системы.

1.3 Определения систем реального времени (СРВ)

Существует несколько определений систем реального времени (ОСРВ) (real time operating systems (RTOS)). Приведем несколько из них, чтобы продемонстрировать различные взгляды на назначение и основные задачи ОСРВ.

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

Таким образом, временные ограничения должны быть гарантированно удовлетворены. Это требует от системы быть предсказуемой, т.е. вне зависимости от своего текущего состояния и загруженности выдавать нужный результат за требуемое время. При этом желательно, чтобы система обеспечивала как можно больший процент использования имеющихся ресурсов.

Хорошим примером задачи, где требуется ОСРВ, является управление роботом, берущим деталь с ленты конвейера. Деталь движется, и робот имеет лишь маленькое временное окно, когда он может ее взять. Если он опоздает, то деталь уже не будет на нужном участке конвейера, и, следовательно, работа не будет сделана, несмотря на то, что робот находится в правильном месте. Если он позиционируется раньше, то деталь еще не успеет подъехать, и он заблокирует ей путь.

Другим примером может быть самолет, находящийся на автопилоте. Сенсорные серводатчики должны постоянно передавать в управляющий компьютер результаты измерений. Если результат какого-либо измерения будет пропущен, то это может привести к недопустимому несоответствию между реальным состоянием систем самолета и информацией о нем в управляющей программе.

2. Стандарт POSIX 1003:1 определяет ОСРВ следующим образом: Реальное время в операционных системах это способность операционной системы *обеспечить требуемый уровень сервиса в заданный промежуток времени*.
3. Иногда системами реального времени называют *системы постоянной готовности (on-line системы), или интерактивные системы с достаточным временем реакции*. Обычно это делают по маркетинговым соображениям. Действительно, если интерактивную программу называют работающей в реальном времени, то это просто означает, что она успевает обрабатывать запросы от человека, для которого задержка в сотни миллисекунд даже незаметна.
4. Иногда понятие "система реального времени" отождествляют с понятием "быстрая система". Это не всегда правильно. Время задержки реакции ОСРВ на событие не так уж важно (оно может достигать нескольких секунд). Главное, чтобы это время было *достаточно для рассматриваемого приложения и гарантированно*. Очень часто алгоритм с гарантированным временем работы менее эффективен, чем алгоритм, таким свойством не обладающий.

Например, если при обработке аудио данных требуется 2:01 секунд для анализа 2:00 секунд

звука, то это не процесс реального времени. Если же требуется 1:99 секунд, то это процесс реального времени.

Попытаемся дать короткое определение системам реального времени:

1. ***Система называется системой реального времени (СРВ), если правильность ее функционирования зависит не только от логической корректности вычислений, но и от времени, за которое эти вычисления производятся.*** То есть для событий, происходящих в такой системе, то, КОГДА эти события происходят, так же важно, как логическая корректность самих событий.
2. ***Говорят, что система работает в реальном времени, если ее быстродействие адекватно скорости протекания физических процессов на объектах контроля или управления.*** Здесь имеются в виду процессы, непосредственно связанные с функциями, выполняемыми конкретной системой реального времени. Система управления должна собрать данные, произвести их обработку в соответствии с заданными алгоритмами и выдать управляющее воздействие за такой промежуток времени, который обеспечивает успешное выполнение поставленных перед системой задач.

Системы реального времени обычно используют специализированное оборудование и программное обеспечение (рис.5).

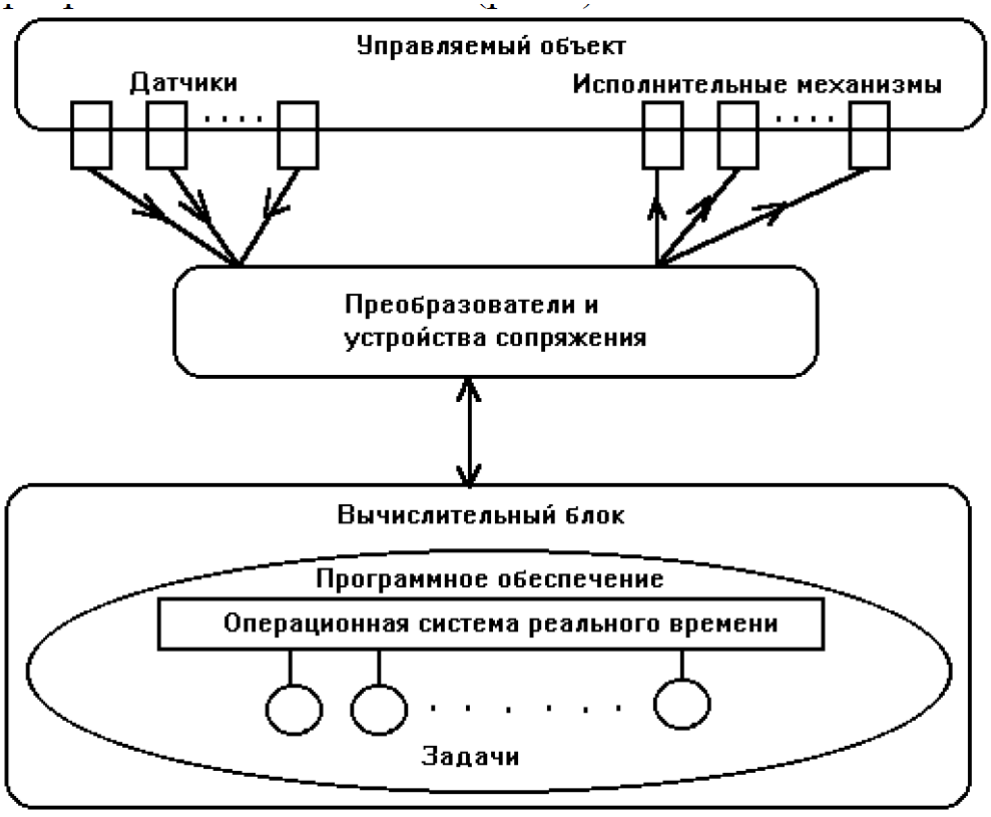

Рисунок 5- Система реального времени

Аппаратные средства СРВ условно можно разделить на две большие группы:

- средства вычислительной техники (ЭВМ с ее стандартными устройствами и интерфейсами);
- специализированные устройства для связи ЭВМ с объектом.

УСО (устройства связи с объектом) – это комплекс устройств, обеспечивающих взаимодействие объектов внешнего мира и ЭВМ. К ним относятся:

1) ***Датчики (или первичные измерительные преобразователи)*** - это устройства, выполняющие преобразование значения физической величины (температуры, давления,

перемещения и т.п.) в электрический сигнал. При этом информация может быть заключена в величине напряжения, тока или частоты изменения сигнала.

2) ***Промежуточные измерительные преобразователи*** – устройства, сохраняющие вид представления сигнала (например, напряжение остается напряжением) и его форму, но изменяющие его величину. Они необходимы в тех случаях, когда электрический сигнал, поступающий с датчика, слишком слаб по величине, либо слишком силен, либо засорен помехами и т.п. К устройствам этого типа относятся разнообразные усилители, нормализаторы, фильтры и пр. Нередко они конструктивно входят в состав датчика (например, мостовая схема подключения термосопротивлений).

3) ***Аналогово-цифровые преобразователи (АЦП)*** – устройства, предназначенные для преобразования значения электрического сигнала в число.

4) ***Цифро-аналоговые преобразователи (ЦАП)*** – устройства, предназначенные для преобразования числовой величины в электрический сигнал (напряжение или ток). Свойства и технические характеристики ЦАП аналогичны АЦП.

5) ***Коммутаторы аналоговых сигналов*** – устройства, осуществляющие физическое переключение (коммутацию) связей между различными устройствами. Различают мультиплексоры (устройства, способные подключать несколько входов на один выход), демультиплексоры (устройства, способные подключать один вход к нескольким выходам) и комбинированные коммутаторы. Существуют также коммутаторы цифровых сигналов – «свитчи».

6) ***Исполнительные устройства*** – устройства, предназначенные для организации непосредственного воздействия на объект. Примером ИУ могут служить шаговый двигатель, электрический нагреватель и т.п.

Средства вычислительной техники, к которым относятся:

1) ***«Промышленные» ПЭВМ***. Назначение – сбор данных и управление многими объектами с большим числом точек доступа в масштабах цеха или предприятия; сбор, обработка, визуализация и хранение потоков данных, поступающих из локальных узлов, в качестве которых выступают промышленные контроллеры и микроконтроллеры. Пример: промышленная ЭВМ ROBO-2000

2) ***Промышленные контроллеры и программируемые логические контроллеры (ПЛК)*** – полноценные микроЭВМ, которые выполняются в виде функциональных модулей магистрально-модульных систем. Основное назначение – решение несложных задач сбора и обработки данных и локального управления в масштабе станка, промышленной установки, автономного агрегата и т.п. Примеры: ADVANTECH I-7188 , PEP/Kontron SMART I/O

3) ***Цифровые сигнальные процессоры (DSP) и цифровые микроконтроллеры (МК)*** – компактные устройства, предназначенные для решения задач автоматизации управления во встроенных и бортовых системах. DSP и МК часто выполняются в виде одной микросхемы (или в виде «чипсета» – комплекта микросхем), интегрирующей в себе ряд устройств. Примеры: микросхемы фирм ESS и YAMAHA в звуковых картах ПЭВМ; чипсеты Rockwell в модемах; микроконтроллер AT43USB351M с ПЗУ 24 Кб, ОЗУ 1 Кб, 10-разрядным АЦП, 8 и 16-битовыми таймерами-счетчиками и программируемым USB-интерфейсом.

1.4 Классификация ОСРВ

1 По времени реакции

По времени реакции различают:

- системы ***жесткого*** (***hard***) реального времени
- системы ***мягкого*** (***soft***) реального времени.

Системы жесткого реального времени не допускают никаких задержек реакции системы ни при каких условиях, т.к.:

1. в случае опоздания результаты окажутся бесполезными
2. в случае задержки реакции может произойти катастрофа
3. стоимость опоздания может оказаться бесконечно велика

Примеры систем жесткого реального времени - бортовые системы управления, системы аварийной защиты, регистраторы аварийных событий.

Системы мягкого реального времени характеризуются тем, что задержка реакции не критична, хотя и может привести к увеличению стоимости результатов и снижению производительности системы в целом. Большинство программного обеспечения ориентировано на слабое реальное время.

Примером может служить работа сети. Если система не успеет обработать очередной принятый пакет, это приведет к вынужденному перерыву на передающей стороне и, например, повторной посылке. Данные при этом не теряются, но производительность сети снижается.

Основное отличие между системами жесткого и мягкого реального времени можно выразить так: система жесткого реального времени никогда не опоздает с реакцией на событие, система мягкого реального времени - не должна опаздывать с реакцией на событие.

Назовем операционной системой реального времени такую систему, которая может быть использована для построения систем жесткого реального времени. Это определение выражает отношение к операционным системам реального времени как к объекту, содержащему необходимые инструменты, но также означает, что этими инструментами еще необходимо правильно воспользоваться.

2 По способу разработки программного обеспечения

По способу разработки программного обеспечения их разделяют на следующие категории:

1. ***Self-Hosted*** ОСРВ - это системы, в которых пользователи могут разрабатывать приложения, работая в самой ОСРВ. Обычно это предполагает, что ОСРВ поддерживает файловую систему, средства ввода-вывода, пользовательский интерфейс, имеются компиляторы, отладчик, средства анализа программ, текстовые редакторы, работающие под управлением ОСРВ.

Достоинством таких систем является более простой и наглядный механизм создания и запуска приложений, которые работают на той же машине, что и пользователь.

Недостатком является то, что промышленному компьютеру во время его реальной эксплуатации часто вообще не требуется пользовательский интерфейс и возможность запуска тяжеловесных программ вроде компилятора. Следовательно, большинство из описанных выше возможностей ОСРВ просто не используются и только зря занимают память и другие ресурсы компьютера.

2. ***Host/Target*** ОСРВ - это системы, в которых операционная система и(или) компьютер, на котором разрабатываются приложения (host), и операционная система и(или) компьютер, на котором запускаются приложения (target), различны. Приложение реального времени разрабатывается на host- компьютере (компьютере системы разработки), затем компонуется с ядром и загружается в целевую систему для исполнения. Как правило, приложение реального времени - это одна задача и параллелизм здесь достигается с помощью нитей (threads). Связь между компьютерами осуществляется с помощью последовательного соединения (COM порта), Ethernet, общей шины VME или compact PCI. В качестве host системы обычно выступают компьютер под управлением UNIX или Windows NT, в качестве target системы, промышленный или встраиваемый компьютер под управлением ОСРВ. Бывают системы, в которых на одном компьютере работают две операционных системы: "обычная" и реального времени.

Системы этого типа обладают рядом *достоинств*, среди которых главное - скорость и реактивность системы. Главная причина высокой реактивности систем этого типа - наличие только нитей (потоков) и, следовательно, маленькое время переключения контекста между ними (в отличие от процессов). Так же *достоинством* таких систем является использование всех ресурсов "обычной" системы (таких, как графический интерфейс, файловая система, быстрый

процессор и большой объем оперативной памяти) для создания приложений и уменьшение размеров ОСРВ за счет включения только нужных приложению компонент.

С этим главным достоинством связан и ряд *недостатков*: зависание всей системы при зависании нити, проблемы с динамической подгрузкой новых приложений. *Недостатком* является и относительная сложность программных компонент: кросс-компилятора, удаленного загрузчика и отладчика, и т.д. Кроме того, системы разработки для продуктов этого класса традиционно дороги (порядка $20000). Хотя, надо отметить, что качество и функциональность систем разработки в этом классе традиционно хороши, так как они были изначально кроссовыми. Наиболее ярким представителем систем этого класса является операционная система VxWorks. Область применения - компактные системы реального времени с хорошими временами реакций.

Отметим, что, с одной стороны, рост мощности промышленных компьютеров позволяет использовать self-hosted системы на большем числе вычислительных систем. С другой стороны, увеличивающееся распространение встраиваемых систем (в разнообразном промышленном и бытовом оборудовании), расширяет сферу применения host/target систем (поскольку при больших объемах выпуска цена системы является определяющим фактором).

3 В зависимости от происхождения

В зависимости от происхождения, ОСРВ разделяют на следующие группы:

1. ***Обычные*** ОС, используемые в качестве ОСРВ. Часто к обычным ОС добавляют дополнительные модули, реализующие поддержку специфического оборудования (например, шины VME), а также планирование задач и обработку прерываний в соответствие с требованиями к ОСРВ и сглаживающие невозможность прервать ядро системы. Все такие системы относятся к разряду self-hosted.
2. ***Собственно*** ОСРВ. Специализированные операционные системы для применения в задачах реального времени. Бывают как self-hosted, так и host/target (большинство), некоторые ОСРВ поддерживают обе модели.
3. ***Специализированные*** (частные) ОСРВ. Это ОСРВ, разработанные для конкретного микроконтроллера его производителем. Часто не являются полноценными ОС, а представляют единый модуль с приложением и обеспечивают только необходимый минимум функциональности. Все такие системы относятся к разряду host/target.

4 По системной программной среде

Становится очевидным то, что задачи реального времени необходимо реализовывать в рамках специфической системной программной среды. Системы реального времени можно разделить на 4 класса.

1-й класс: программирование на уровне микропроцессоров. При этом программы для программируемых микропроцессоров, встраиваемых в различные устройства, очень небольшие и обычно написаны на языке низкого уровня типа ассемблера или PLM. Внутрисхемные эмуляторы пригодны для отладки, но высокоуровневые средства разработки и отладки программ не применимы. Операционная среда обычно недоступна.

2-й класс: минимальное ядро системы реального времени. На более высоком уровне находятся системы реального времени, обеспечивающие минимальную среду исполнения. Предусмотрены лишь основные функции, а управление памятью и диспетчер часто недоступны. Ядро представляет собой набор программ, выполняющих типичные, необходимые для встроенных систем низкого уровня функции, такие, как операции с плавающей запятой и минимальный сервис ввода/вывода. Прикладная программа разрабатывается в инструментальной среде, а выполняется, как правило, на встроенных системах.

3-й класс: ядро системы реального времени и инструментальная среда. Этот класс систем обладает многими чертами ОС с полным сервисом. Разработка ведется в инструментальной среде, а исполнение - на целевых системах. Этот тип систем обеспечивает гораздо более высокий уровень сервиса для разработчика прикладной программы. Сюда включены такие средства, как

дистанционный символьный отладчик, протокол ошибок и другие средства CASE. Часто доступно параллельное выполнение программ.

4-й класс: ОС с полным сервисом. Такие ОС могут быть применены для любых приложений реального времени. Разработка и исполнение прикладных программ ведутся в рамках одной и той же системы.

Системы 2 и 3 классов принято называть системами "жесткого" реального времени, а 4 класса - "мягкого". Очевидно, это можно объяснить тем, что в первом случае к системе предъявляются более жесткие требования по времени реакции и необходимому объему памяти, чем во втором.

5 По внутреннему строению

Различают:

1. классические (QNX, pSOS, VxWorks)
2. объектно-ориентированные системы (SoftKernel, RT-Linux). К их числу мы будем относить системы, не только предоставляющие средства разработки и наборы библиотек для объектно-ориентированных языков, но и сами написанные на таких языках.

1.5 Особенности и требования, предъявляемые к ОСРВ . Области применения

В системах реального времени (СРВ), в которых главным критерием эффективности является обеспечение временных характеристик вычислительного процесса, планирование имеет особое значение. Специфика процесса планирования, определяется требованием своевременного выполнения прикладных задач. По мере расширения практики применения СРВ, расширялся и совершенствовался состав методов организации вычислений. В частности, расширялся состав, и повышалась эффективность используемых методов планирования выполнения задач.

Основная задача системы реального времени - получение правильных результатов за определенный крайний срок. Следовательно, вычислительная правильность системы зависит от двух составляющих: логической правильности результатов, и правильности выбора времени[2].

Система реального времени должна давать отклик на любые непредсказуемые внешние воздействия в течение предсказуемого интервала времени.

Для этого должны выполняться следующие требования.

• *Ограничение времени отклика.* После наступления события реакция на него гарантированно должна последовать до предустановленного крайнего срока. Отсутствие такого ограничения рассматривается как серьезный недостаток программного обеспечения.

• *Одновременность обработки.* Даже если наступает более одного события одновременно, все временные ограничения для всех событий должны быть выдержаны.

Это означает, что системе реального времени должен быть присущ параллелизм, что достигается использованием нескольких процессоров или многозадачного подхода.

Основные требования к операционным системам реального времени.

Мультипрограммность и мультизадачность.

Требование 1. Операционная система должна быть мультипрограммной и мультизадачной *(многопоточной* — multi-threaded), а также активно использовать прерывания для диспетчеризации.

Максимальное время выполнения того или иного действия в ОСРВ должно быть известно заранее и соответствовать требованиям приложения.

Операционная система так же должна быть многопоточной на принципе абсолютного приоритета (прерываемой). То есть планировщик должен иметь возможность прервать любой поток выполнения и предоставить ресурс тому потоку, которому он более необходим. Операционная система (и аппаратура) должна также обеспечивать прерывания на уровне обработки прерываний.

Приоритеты задач.

Требование 2. В системе реального времени должны существовать гарантии того, что событие с высоким приоритетом будет обработано перед событием более низкого приоритета.

ОСРВ должна обладать развитой системой приоритетов. Во-первых, это требуется потому, что система сама может рассматриваться как набор приложений, подразделяющихся на потоки, и несколько высоких уровней приоритетов должны быть выделены системным процессам и потокам. Во-вторых, в сложных приложениях необходимо все потоки реального времени помещать на разные приоритетные уровни, а потоки не реального времени помещать на один уровень (ниже, чем любые потоки реального времени). При этом потоки не реального времени можно обрабатывать в режиме циклического планирования при котором каждому процессу предоставляется квант времени процессора, а когда квант заканчивается, контекст процесса сохраняется, и он ставится в конец очереди. Во многих ОСРВ для планирования задач на одном уровне используется режим циклического планирования.

Наследование приоритетов.

Требование 3. Должна существовать система наследования приоритетов.

Комбинация приоритетов потоков и разделение ресурсов между ними приводит к проблеме инверсии приоритетов. Это можно проиллюстрировать на примере, где есть как минимум три потока. Когда поток низшего приоритета захватил ресурс, разделяемый с потоком высшего приоритета, и начал выполняться поток среднего приоритета, выполнение потока высшего приоритета будет приостановлено, пока не освободится ресурс и не отработает поток среднего приоритета. В этой ситуации время, необходимое для завершения потока высшего приоритета, зависит от нижних уровней приоритетов — это и есть инверсия приоритетов. В такой ситуации трудно выдержать ограничение на время исполнения.

Чтобы устранить такие инверсии, ОСРВ должна допускать наследование приоритета, то есть повышение уровня приоритета потока до уровня потока, который его вызывает. Наследование означает, что блокирующий ресурс поток наследует приоритет потока, который он блокирует, но это справедливо только в случае, когда блокируемый поток имеет более высокий приоритет.

Синхронизация процессов и задач.

Требование 4. Операционная система должна обеспечивать мощные, надежные и удобные механизмы синхронизации задач. Так как задачи разделяют данные (ресурсы) и должны сообщаться друг с другом, представляется логичным существование механизмов блокирования и коммуникации. То есть необходимы механизмы, гарантированно предоставляющие возможность оперативно обменяться сообщениями и синхросигналами между параллельно выполняющимися задачами и процессами. Эти системные механизмы должны быть всегда доступны процессам, требующим реального времени. Следовательно, системные ресурсы для их функционирования должны быть распределены заранее. Пренебрежение вопросами синхронизации процессов, выполняющихся в режиме мультипрограммирования, может привести к их неправильной работе или даже к краху системы.

Предсказуемость.

Требование 5. Поведение операционной системы должно быть известно и достаточно точно прогнозируемо. Время выполнения системных вызовов и временные характеристики поведения системы в различных обстоятельствах должны быть известны разработчику. Поэтому создатель ОСРВ должен приводить следующие характеристики:

• задержку прерывания, то есть время от момента прерывания до момента запуска задачи: она должна быть предсказуема и согласована с требованиями приложения;

• максимальное время выполнения каждого системного вызова (оно должно быть предсказуемо и не должно зависеть от числа объектов в системе)[1].

Общие характеристики ОСРВ это:

- большие и сложные системы,
- распределенные системы,

- жесткое взаимодействие с аппаратурой,
- выполнение задач зависит от времени,
- сложность в тестировании.

Области применения ОСРВ

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

Основные области применения ОСРВ:

- Военная и космическая области: бортовое и встраиваемое оборудование:
  - системы измерения и управления, радары;
  - цифровые видеосистемы, симуляторы;
  - ракеты, системы определения положения и привязки к местности.
- Промышленность:
  - автоматические системы управления производством (АСУП) (computer-aided manufacturing (CAM)), автоматические системы управления технологическим процессом (АСУТП);
  - автомобилестроение: симуляторы, системы управления мотором, автоматическое сцепление, системы антиблокировки колес.
  - энергетика: сбор информации, управление данными и оборудованием.
  - телекоммуникации: коммуникационное оборудование, сетевые коммутаторы, телефонные станции.
  - банковское оборудование (например, во многих банкоматах работает ОСРВ QNX).
- Товары широкого потребления:
  - мобильные телефоны, например, в телефонах стандарта GSM работает ОСРВ pSOS;
  - цифровые телевизионные декодеры;
  - цифровое телевидение (мультимедиа, видеосерверы...);
  - компьютерное и офисное оборудование (принтеры, копиры), например, в факсах применяется ОСРВ VxWorks, в устройствах чтения компакт-дисков - ОСРВ VRTX32.

Отметим, что часто ОСРВ существуют в нескольких вариантах: полном и сокращенном, когда объем системы составляет несколько килобайтов.

1.6 Отличия ОСРВ от ОС общего назначения

Все ОСРВ сегодня являются многозадачными системами. Задачи делят между собой ресурсы вычислительной системы, в том числе и процессорное время.

Количество иллюзий и мифов в мире реального времени велико. Часто путают такие понятия, как ***реальное время*** и ***скорость***. Иногда полагают, что применение операционной системы реального времени (ОСРВ) автоматически разрешит все проблемы надежности предсказуемости. Порой, наоборот, считают, что системы реального времени - занятие для теоретиков, а практически любую задачу реального времени можно решить с помощью популярных ОС общего назначения: достаточно быть просто хорошим программистом и знать архитектуру компьютера. Так ли это?

Чем принципиально отличаются операционные системы реального времени от операционных систем общего назначения?

1. ОС общего назначения, особенно многопользовательские вроде UNIX, ориентированы на оптимальное распределение ресурсов компьютера между пользователями и выполняемыми процессами (системы разделения времени). В ОСРВ подобная задача отходит на второй план, все отступает перед главной целью - успеть среагировать на события, происходящие на объекте.

2. Другое отличие состоит в том, что применение ОСРВ всегда связано с аппаратурой, объектом и событиями, происходящими на объекте. Система реального времени как аппаратно-программный комплекс включает в себя:

- ***датчики***, регистрирующие события на объекте,
- ***модули ввода-вывода***, которые преобразуют показания датчиков в цифровой вид, пригодный для их обработки на компьютере,
- и, наконец, ***компьютер*** с программой, реагирующей на события в объекте.

ОСРВ ориентирована на обработку внешних событий. Именно это обуславливает коренные отличия от ОС общего назначения:

- по структуре,
- по функциям ядра,
- по построению системы ввода-вывода.

Т.о. ОСРВ может быть похожа на ОС общего назначения по пользовательскому интерфейсу (к этому, кстати, стремятся почти все производители ОС реального времени), однако устроена она совершенно иначе.

3. Кроме того, применение операционных системах реального времени всегда конкретно. Если ОС общего назначения обычно воспринимается пользователями (не разработчиками) уже готовый набор приложений, то ОСРВ служит только инструментом для создания того или иного аппаратно-программного комплекса реального времени. И поэтому наиболее широкий класс пользователей ОСРВ составляют разработчики таких комплексов - люди, проектирующие системы управления и сбора данных. Проектируя и разрабатывая конкретную систему реального времени, программист всегда знает точно, какие события могут произойти на объекте, знает критические сроки обработки каждого из этих событий.

Назовем системой реального времени (СРВ) аппаратно-программный комплекс, реагирующий в течение предсказуемого времени на непредсказуемый поток внешних событий.

Это определение означает следующее:

1. Во-первых, система должна успеть отреагировать на событие, произошедшее на объекте, в течение времени, критического для этого события (meet deadline). Критическое время для каждого события определяется объектом и самим событием и, естественно, может быть разным, но время реакции системы должно быть предсказано (вычислено) при создании системы. Отсутствие реакции в течение предсказанного времени считается ошибкой для СРВ.

2. Во-вторых, система должна успевать реагировать на одновременно происходящие события. Если два или несколько внешних событий происходят одновременно, ей нужно успеть среагировать на каждое из них в течение интервалов времени, критических для этих событий.