Лекция 2. Особенности и требования, предъявляемые к ОСРВ

План лекции

1 Классификация ОСРВ

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.

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).

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.

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.

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.

2 Особенности и требования, предъявляемые к ОСРВ

Основные требования, предъявляемые к ОСРВ следующие:

- требования по времени,
- возможность параллельного выполнения нескольких задач,
- предсказуемость,
- важно максимальное время отклика на событие, а не среднее,
- особые требования в вопросах безопасности,
- возможность безотказной работы в течение длительного периода времени.

Общие характеристики ОСРВ это:

- большие и сложные системы,
- распределенные системы,
- жесткое взаимодействие с аппаратурой,
- выполнение задач зависит от времени,
- сложность в тестировании.

3 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.

Упражнения к лекции 2.

1. Что такое системы мягкого реального времени?
2. Что такое системы жесткого реального времени?
3. Что общего и в чем различие систем мягкого и жесткого реального времени?
4. Self-Hosted ОСРВ. Их достоинства и недостатки.
5. Host/Target ОСРВ. Их достоинства и недостатки.
6. В зависимости от происхождения какие бывают ОСРВ. К какому типу (Self-Hosted или Host/Target) они относятся и почему?
7. В зависимости от системной программной среды на какие классы делятся ОСРВ? К какому типу (Self-Hosted или Host/Target) они относятся и почему?
8. В зависимости от внутреннего строения какие бывают ОСРВ?
9. Какие основные отличия ОСРВ от ОС общего назначения?

Литература к лекции 1.

1.1 Мартин ДЖ. Программирование для вычислительных систем реального времени. Пер. с англ. Изд-во "Наука", 1975.- 360с.

1.2 Тенанбаум Э. Современные операционные системы. пер. с англ. 2-е изд. – М.: СПБ.: Нижний Новгород: Питер, 2005. – 1037с.

1.3 Олифер В.Г. Олифер Н.А. Сетевые операционные системы. М.: СПБ.: Нижний Новгород: Питер, 2006. – 538с.

1.4 Грибанов В.П., Дробин С.В., Медведев В.Д. Операционные системы. - М.: Финансы и статистика, 1990. - 239 с.

1.5 Дейтел Х.М., Чофнес Р.Д. Операционные системы. пер. с англ. – М.: БИНОМ, 2006. – 704с.