

АНАЛИЗ ОПЕРАЦИОННЫХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

Институт компьютерных технологий
Национального авиационного университета

Рассмотрены операционные системы реального времени, основные параметры и механизмы, обеспечивающие требования реального времени. Обсуждаются проблемы и тенденции развития операционных системах реального времени. Сделана попытка классификации систем реального времени. Проанализированы продукты в мире операционных системах реального времени

Введение

В последнее время все чаще приходится сталкиваться с задачами, требующими управления сложными процессами или оборудованием при помощи ЭВМ. Компьютер может выполнять лишь конечное число операций в конечное время, поэтому возникает вопрос: а успеет ли компьютер с нужной скоростью обработать ситуацию и выдать конкретные управляющие действия, которые были бы адекватны именно в определенный момент времени. Проблемы подобного рода возникли из-за использования очень больших скоростей в современном производстве. Ведь каким бы современным компьютер бы ни был - ему физически нужны хотя бы доли секунды, чтобы выполнить небольшую простую группу команд, а иногда этого времени слишком много. Таким образом, имеет смысл говорить о системе реального времени.

Постановка проблемы

Для повышения эффективности работы вычислительных систем необходимо сделать анализ и обосновать требования к операционным системам реального времени.

На длительность обработки информации, превышение которой может привести к потере качества управления или к неуправляемости процесса в вычислительных системах (ВС) реального времени, накладываются ограничения со стороны внешних факторов. Таким образом, ВС реального времени должны обеспе-

чить обработку информации за фиксированный промежуток времени, превышение которого недопустимо. Для обслуживания экстремальных ситуаций так же необходимо предусмотреть запас времени.

Пути решения проблемы

Различают системы реального времени двух типов – системы жесткого реального времени и системы мягкого реального времени.

Системы жесткого реального времени не допускают никаких задержек реакции системы ни при каких условиях.

Примеры систем жесткого реального времени - системы аварийной защиты.

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.

Пример – работа сети. Если система не успела обработать очередной принятый пакет, это приведет к таймауту на передающей стороне и повторной посылке (в зависимости от протокола, конечно). 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 опаздывает 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.

Определение и предназначение ОСРВ

Операционные системы реального времени (ОСРВ) предназначены для обеспечения интерфейса к ресурсам критических по времени систем реального времени. Основной задачей в таких системах является своевременность выполнения обработки данных.

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 даже беглый взгляд на возможные встроенные системы позволяет утверждать, что большинство встроенных систем нуждается в предсказуемом поведении, по крайней мере, для некоторой функциональности, и таким образом, эти системы можно отнести к системам реального времени.

Требования к ОСРВ

В качестве основного требования к ОСРВ выдвигается 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.

Операционная система должна быть многозадачной и допускающей вытеснение, обладать понятием приоритета для потоков.

Поведение ОС должно быть известным и предсказуемым. Это значит, что во всех сценариях рабочей нагрузки системы должно быть определено максимальное время отклика.

Структура операционных систем

Большинство современных ОСРВ построено на основе микроядра, которое обеспечивает планирование и диспетчеризацию задач, а также осуществляет их взаимодействие. Несмотря на сведение к минимуму в ядре абстракций ОС, микроядро все же должно иметь представление об абстракции процесса. Все остальные концептуальные абстракции операционных систем вынесены за пределы ядра, вызываются по запросу и выполняются как приложения.

Задачи, процессы, потоки

Существуют различные определения термина «задача» для многозадачной ОСРВ. Мы будем считать задачей набор операций, предназначенный для выполнения логически законченной функции системы. Принято различать две разновидности задач: процессы и потоки. Процесс представляет собой отдельный загружаемый программный модуль, который, как правило, во время исполнения имеет в памяти свои независимые области для кода и данных. В отличие от этого потоки могут пользоваться общими участками кода и данных в рамках единого программного модуля.

Планирование задач

Важной частью любой ОСРВ является планировщик задач. Его функции остаются теми же: определить, какая из задач должна выполняться в системе в каждый конкретный момент времени. Самым простым методом планирования, не требующим никакого специального ПО и планировщика как такового, является использование циклического алгоритма в стиле *round robin*.

Каждая «задача», представляющая собой отдельную подпрограмму, выполняется циклически.

К недостаткам циклического алгоритма можно отнести отсутствие приоритетности и очередей. К тому же задачи вызываются независимо от того, должны ли они в данный момент что-либо делать или нет, а на прикладного программиста

ложится максимальная ответственность за работоспособность системы.

Ядра реального времени

Ядра предоставляют пользователю такие базовые функции, как планирование и синхронизация задач, межзадачная коммуникация, управление памятью и т. п.

По своей внутренней архитектуре ОСРВ можно разделить на монолитные ОС (рис. 1), ОС на основе микроядра (рис. 2) и объектно-ориентированные ОС (рис. 3).

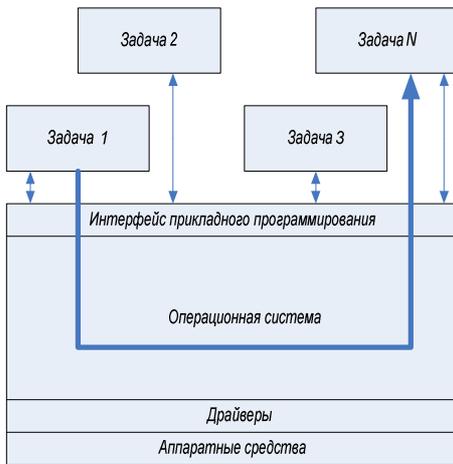


Рис.1. ОС РВ с монолитной архитектурой

Системы с монолитным ядром содержат реализацию всех механизмов реального времени.

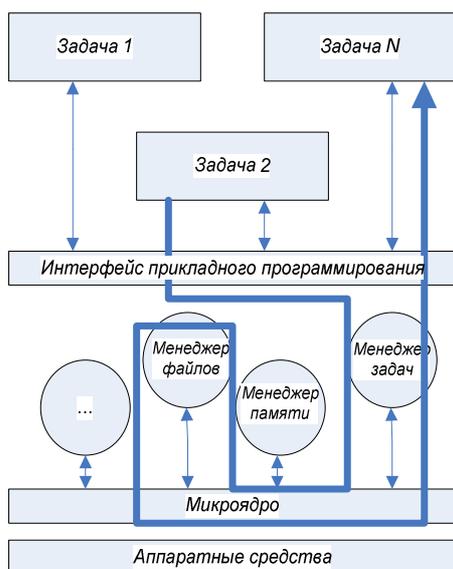


Рис.2. ОС РВ на основе микроядра

Одна из их особенностей – высокая степень масштабируемости. На базе этих ОС можно построить как компактные системы реального времени, так и большие системы серверного класса.

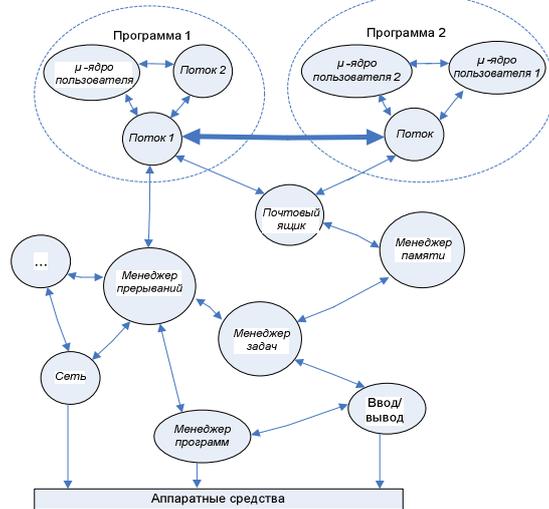


Рис. 3. Объектно-ориентированная ОС РВ

Как правило, ядра реального времени имеют два типа систем разработки – кроссовую и резидентную.

Обзор ОС РВ

Рассмотрим краткое описание некоторых ОСРВ:

1. Операционные системы реального времени семейства *VxWorks* корпорации *WindRiver Systems* предназначены для разработки программного обеспечения встраиваемых компьютеров, работающих в системах жесткого РВ. *VxWorks* имеет архитектуру клиент-сервер и построена в соответствии с технологией микроядра, т.е. на самом нижнем непрерываемом уровне ядра (*WIND Microkernel*) обрабатываются только планирование задач и управление их взаимодействием.

2. Операционная система *QNX Neutrino Realtime Operating System (RTOS)* корпорации *QNX Software Systems* является микроядерной операционной системой, которая обеспечивает многозадачный режим с приоритетами. *QNX Neutrino RTOS* имеет клиент-серверную архитектуру. Состоит из ядра, планировщика процессов и расширенных сервисов на уровне пользователя. Как истинная микроядерная операционная система, *QNX Neutrino*

RTOS реализует в ядре ОС только наиболее фундаментальные сервисы, такие как передача сообщений, сигналы, таймеры, планирование потоков, объекты синхронизации. Все другие сервисы ОС, драйверы и приложения выполняются как отдельные процессы, которые взаимодействуют через синхронную передачу сообщений.

3. *QNX Neutrino RTOS* способна быстро и с высокой предсказуемостью реагировать на события РВ. Высокоприоритетные потоки обрабатывают дедлайны своевременно даже при большой загрузке системы.

4. *RTEMS (Real-Time Executive for Multiprocessor Systems)* – это некоммерческая ОСРВ для глубоко встраиваемых систем. Система разрабатывается для многопроцессорных систем на основе открытого исходного кода в противовес аналогичным системам с закрытым кодом. Система рассчитана на платформы *MS-Windows* и *Unix (GNU/Linux, FreeBSD, Solaris, MacOS X)*.

5. *ОСРВ RTEMS* можно рассматривать как набор компонентов, обеспечивающих ряд базовых сервисных функций для программ пользователя. Программный интерфейс приложения состоит из директив, распределенных по логическим наборам соответствующих менеджеров. Функции, используемые несколькими менеджерами, такие как распределение процессорного времени, диспетчеризация и управление объектами, реализованы в ядре. Ядро содержит также небольшой набор процедур, зависящих от типа используемого процессора: доступ к физической памяти, инициализация контроллера пре-

рываний и периферийных устройств, специфичных для данного процессорного ядра, и т.д.

6. Операционная система *ChorusOS* – это масштабируемая встраиваемая ОС, широко применяемая в телекоммуникационной индустрии.

7. ОС *ChorusOS* поддерживает на одной аппаратной платформе широкий набор телекоммуникационных протоколов, унаследованных приложений, приложений режима реального времени и *Java*-технологии.

8. ОСРВ *pSOS* была разработана корпорацией *Integrated Systems*.

9. Имя *pSOSsystem* присвоено операционной системе, имя *pSOS+* – ее ядру. *pRISM+* – это интегрированная среда разработки для создания приложений.

10. *pSOS+* – это маленькое ядро встраиваемых приложений, представляющее собой некий вариант клиент-серверной архитектуры. Однако оно не имеет протокола взаимодействия, основанного на сообщениях. Для взаимодействия модулей используется программная шина. Есть возможность выбрать и встроить модули в систему во время компиляции.

11. Операционная система *LynxOS RTOS* является операционной системой жесткого РВ, которая предназначена для специализированной и телекоммуникационной аппаратуры. Эта ОС является полностью детерминированной и обладает *POSIX*-, *UNIX*- и *Linux*-совместимостью. Области применения ОС *LynxOS* являются сложные системы безопасности. Сводная таблица характеристик свойств ОСРВ приведена в табл. 1.

Таблица 1. Характеристики ОСРВ

ОСРВ	Архитектура	Предсказуемая производительность реального времени	Что реализует микроядро, размер (мин., макс.)
<i>VxWorks</i>	Клиент-сервер, микроядро <i>WIND</i> <i>Microkernel</i>	Приоритетное планирование в двух вариантах, наследование приоритетов	Многозадачность, планирование, переключение контекста, взаимодействие /синхронизация задач

продолжение табл. 1.

<i>QNX</i>	Клиент-сервер, микроядро и взаимодействующие процессы	Приоритетное планирование с выбором методов планирования. Наследование приоритетов	Потоки, сигналы, передача сообщений, синхронизация, планирование, временные сервисы
<i>pSOS</i>	Клиент-сервер, отсутствует протокол взаимодействия на основе сообщений, вместо него используется программная шина	Приоритетное планирование, отсутствует наследование приоритетов	
<i>ChorusOS</i>	Многослойная	Приоритетное планирование, мьютексы реального времени, таймеры с высокой разрешающей способностью, <i>MIPC</i>	Многозадачность, поддержка акторов, управление потоками, управление <i>LAP</i> , управление исключительными ситуациями, минимальное управления прерываниями
<i>OS-9</i>		Приоритетное планирование, механизм предотвращения инверсии приоритетов	размер – 128К, 4МВ
<i>LynxOS</i>			размер – 280К, 4М

12. Операционная система реального времени *OS-9* корпорации *Microware System* является многозадачной, многопользовательской операционной системой для встраиваемых приложений, работающих в режиме РВ. Эта система предназначена для работы в таких системах, как мобильные телекоммуникационные устройства, встраиваемые терминалы доступа в *Internet*, интерактивные цифровые телевизионные приставки. *OS-9* работает на таких процессорах, как *Motorola 68К*, *ARM/StrongARM*.

13. *OSEK/VDX* является комбинацией стандартов компьютерных систем реального времени, разработанных консорциумами *OSEK* и *VDX* для автомобильной промышленности.

Выводы

Обзор ОСРВ был подобран с целью показать широкий спектр технологических решений, которые могут быть выбраны при проектировании систем реального времени.

1. Исследован комплексный подход к рассмотрению операционных систем реального времени, основных параметров и механизмов;

2. проклассифицировано системы реального времени;

3. проанализированы продукты в мире операционных систем реального времени

4. Отметим, что «времязависимость» есть неизменным атрибутом системы РВ. Но выполнение этого требования еще недостаточно для эффективной работы системы.

Список литературы

1. *Жабин В.И.* Архитектура вычислительных систем реального времени. - К.: ВЕК+, 2003. – 176 с.

2. *Симоненко В.П.* Организация вычислительных процессов в ЭВМ, комплексах, сетях и системах. – К.: “Век +”, 1997. – 304 с.

3. *Поспелов Д.А.* Введение в теорию вычислительных систем. – М.: Сов. радио, 1972. – 280 с.