

WINDOWS КАК АЛЬТЕРНАТИВА КЛАССИЧЕСКИМ ОС РВ

А.В. Исаев (ЗАО "РТСофт"),

Л.Г. Акиншин (Журнал "МКА: мир ВКТ")

Желание использовать стандартные ОС настольного класса во встраиваемых системах и в приложениях РВ существует на рынке едва ли не с того самого момента, как появились самые первые ОС. Благодаря расширению RTX компании IntervalZero данное желание может быть удовлетворено в широком диапазоне прикладных задач.

Ключевые слова: ОС, РВ, Windows, надежность, детерминированность, разделение ресурсов, прерывания.

Фантастическая ситуация, когда в мире осталась бы всего одна "сверхуниверсальная" ОС, удовлетворяющая требованиям абсолютно всех задач — офисных, встраиваемых, ответственных, РВ и др., устроила бы практически всех. Приобретая опыт работы с какой-то одной ОС в какой-то одной прикладной сфере, люди желают использовать этот опыт и эту ОС в других областях, включая и те, на которые она изначально не рассчитана. Ближе всего к реализации мечты человечества о "сверхуниверсальной" ОС позволило подойти расширение RTX (Real Time eXtension), добавляющее в стандартные офисные и встраиваемые ОС семейства Windows (кроме Windows CE) поддержку РВ. Но до появления ОС Windows NT выход на рынок такого продукта, как RTX был невозможен.

Первая устойчивая многозадачная ОС общего назначения

Приход Windows NT означал реальный шанс для воплощения основной идеи RTX: сделать так, чтобы популярную недорогую ОС общего назначения можно было *комфортно* использовать во встраиваемых системах и системах жесткого РВ. В 90-е годы XX века продукция корпорации Microsoft была все рекорды популярности. У линейки Windows была лишь одна очень серьезная проблема: она не отличалась высокой надежностью, в связи с чем об использовании во встраиваемых приложениях и тем более в задачах РВ не могло быть и речи.

Платформа Windows NT стала первой устойчивой многозадачной ОС общего назначения и открыла новую эру в истории корпорации Microsoft и ОС вообще. А с легкой руки компании VenturCom, создавшей расширение RTX — и в истории систем РВ.

Роль Windows NT в формировании и развитии рынка встраиваемых компьютерных технологий также весьма велика. В середине 90-х годов XX века встраиваемые проекты на базе Windows NT стали возникать массово. Для комфортного использования этой ОС во встраиваемых системах и задачах РВ разработчикам не хватало лишь:

1) возможности уменьшать размеры ОС до приемлемых величин;

2) средств обеспечения работы ОС в бездисковых конфигурациях (из флеш-памяти), в том числе без монитора и клавиатуры;

2) средств обеспечения детерминированного поведения.

Решения компании VenturCom

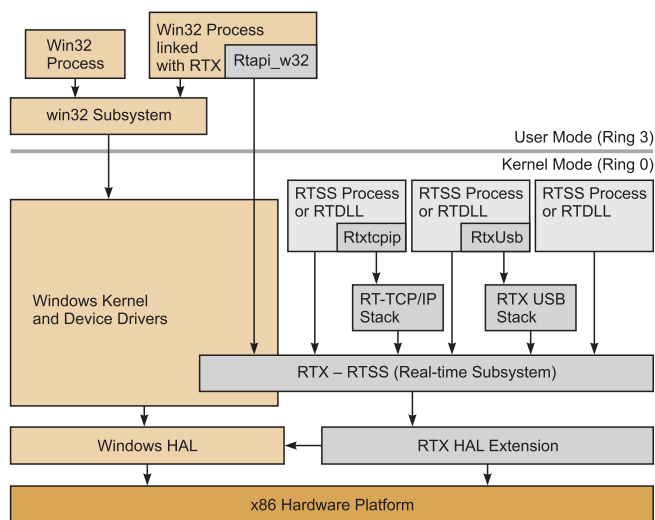
В ответ компания VenturCom выпустила свои легендарные продукты Component Integrator и RTX. Первый представлял разработчику ОС Windows NT в качестве отдельных компонентов, из которых можно собрать ОС с необходимой функциональностью. Расширение RTX позволяло добавлять в "настольную" ОС Windows NT поддержку работы в режиме жесткого РВ.

Результатом применения инструмента Component Integrator являлось значительное (нередко в несколько раз) снижение требований к аппаратуре, а также дополнительное повышение надежности и производительности. Впоследствии чрезвычайно успешная технология декомпозиции и сборки нужного образа для встраиваемой системы Component Integrator была куплена корпорацией Microsoft и под именем Component Designer стала основой нового бизнес-направления Microsoft: производство и поставка встраиваемых ОС (продуктовая серия Windows Embedded). В настоящее время поставками расширения RTX занимается компания IntervalZero Inc., штат которой почти полностью укомплектован бывшими сотрудниками компании VenturCom.

Сегодня об инструменте Component Designer и расширении РВ RTX известно далеко за пределами профессиональных кругов. И все это время продукты успешно развиваются, обрастая новыми функциями и возможностями.

Как это работает

Продукт RTX состоит из двух частей. Первая часть — пакет разработчика RTX SDK (Software Development Kit) — предназначен для создания собственных приложений для работы в среде RTX. Вторая — подсистема исполнения RTX Runtime — представляет собой непосредственно подсистему РВ для обслуживания приложений RTX. Подсистема RTX



Архитектура RTX

Runtime устанавливается на целевые системы, где предполагается запуск RTX-приложений. Пакет разработчика также включает подсистему PB, поэтому разрабатываемые приложения RTX можно запускать и отлаживать непосредственно на локальном месте разработчика. Кроме того, очень ценным свойством SDK является его тесная интеграция со средой разработки Visual Studio от компании Microsoft.

Последней версией RTX является RTX 2009. Ранее версии имели номера (вплоть до 8.1.1), теперь вместо номеров используются годы выпуска. Компания IntervalZero обеспечивает 10-летний срок поддержки своих продуктов, поэтому, например, приобрести системы исполнения к ранее купленной версии не составляет труда. Самая ранняя поддерживаемая на данный момент версия – это RTX 6.0.1.

Стоит отметить, что все предлагаемые на рынке решения не меняют функциональность и возможности самой ОС Windows; в ее поведении и работе ничего не меняется. Подсистема, реализующая функции ОС PB, работает "рядом" с Windows. Обычные Windows-задачи выполняются в недетерминированной среде, как и раньше. Один из вариантов реализации подобного механизма – разделение ресурсов между ОС Windows и подсистемой PB. Подобный подход реализовала, например, компания TenAsys, предлагающая продукт INTime. При использовании INTime на одной аппаратной платформе запускаются две виртуальные машины – с ОС Windows и подсистемой PB INTime. Здесь в целом неважно, какая ОС выполняется параллельно с подсистемой PB, поддержка Windows определяется только наличием модифицированного HAL для работы в таком режиме и наличием API для связи задач, работающих в режиме PB и в Windows-среде. С популяризацией многоядерных архитектур стало возможным и просто выделение отдельных ядер под Windows и под задачи PB. Так поступает компания RadiSys для параллельного запуска ОС PB OS-9 и ОС Linux на двухядерном процессоре. Однако довольно сложная архитектура того же INTime не позволяет добиться высокой производительности в задачах PB.

В отличие от описанного подхода RTX тесно интегрируется с ОС Windows (рисунок) и дополняет стандартный HAL Windows расширением RTX HAL Extension. На этом уровне кроме организации доступа к аппаратуре обрабатываются прерывания от таймера подсистемы PB. Непосредственно функциональность PB реализует слой RTSS (Real-Time SubSystem) – ядро всей подсистемы PB. Здесь находится свой планировщик, который оперирует выполнением задач PB и предоставлением ресурсов задачам Windows-среды. Фактически, любая задача RTSS имеет более высокий приоритет, нежели любая задача Windows. Также этот слой полностью реализует API PB (Real-Time API – RTAPI), на основе которого создаются приложения подсистемы RTSS.

Приложения PB (RTSS Process) здесь выполняются на уровне ядра Windows, имеют те же привилегии и ограничения, что и драйверы устройств. Этот подход отличается от других реализаций (например, INTime обеспечивает приложениям PB пользовательский режим, собственное адресное пространство и изоляцию). Но за счет этого достигается очень быстрое переключение контекста задач.

Доступ к функциям RTAPI возможен как для процессов RTSS, так и для "обычных" приложений Win32. Это позволяет выделять в Windows-задачах отдельные критичные к времени выполнения участки. Такой возможности лишены системы, где Windows и подсистема PB работают параллельно. Кроме того, расширение интерфейса Win32 API функциями RTSS позволяет приложениям Win32 и приложениям RTSS обмениваться данными.

В составе RTX присутствуют компоненты, организующие для приложений RTSS сеть PB и работу в PB с устройствами USB – стек протоколов RT-TCP/IP и стек RTX USB. Первый обеспечивает работу всех сетевых протоколов и физического уровня в детерминированном режиме, компоненты второго также работают с полностью предсказуемыми задержками.

Фактически, наличие стека PB означает работу всех функций и драйверов с гарантированными временами. То есть разработчик может быть уверен, что входящий пакет будет обработан, и исходящие данные будут доставлены до физического уровня за ограниченный промежуток времени. При этом наличие стека RT-TCP/IP не означает, что вся сеть будет работать в PB, даже если данный стек будет стоять на всех узлах сети. Сеть состоит из многих элементов, в частности, в ней есть оборудование, которое реализует ее топологию. На физическом уровне сети могут происходить различные процессы, дополнительное оборудование может вносить свои задержки, кроме того, ОС Windows тоже не бездействует в плане сетевой активности. Поэтому сетевой стек PB позволяет получить гарантированную производительность, но не гарантирует работу всей сети без сбоев и задержек. Стек RT-TCP/IP предназначен только для достижения максимума производительности в сетевых задачах.

Несмотря на тесную интеграцию с ОС, RTX обладает важным свойством устойчивости к сбоям среды Windows. Вся подсистема РВ продолжает работать, даже если Windows аварийно завершилась и выдала "синий экран смерти". Естественно, работать продолжают только слой RTSS, драйверы РВ и процессы RTSS.

Что RTX может дать разработчику? Во-первых, использование стандартной ОС Windows позволяет обойтись без дополнительного аппаратного обеспечения и специализированных ОС РВ. Весь проект от начала до конца может быть реализован специалистами, ранее разрабатывавшими только обычные Windows-приложения, так как API RTX максимально приближен к API Win32 и не требует большого времени на освоение. Простоте использования способствует и интеграция со знакомой системой разработки Microsoft Visual Studio. Все это в совокупности позволяет сократить затраты на развертывание проекта, поскольку стоимость владения и применения RTX сопоставима (а в большинстве случаев ниже) со стоимостью традиционных ОС РВ.

Во-вторых, применение RTX не является неким компромиссом или ограничением возможностей. Разработчик получает современную, мощную, быструю и гибкую подсистему РВ, не уступающую по функциональности классическим ОС РВ. Абсолютно все процессы в подсистеме RTSS детерминированы, а это является основой любой системы РВ.

Детерминированность процессов позволяет говорить, что RTX обладает свойствами системы жесткого РВ. Ограниченное время работы имеют все механизмы синхронизации, обработчики прерываний, сетевые функции, драйвера. Также для получения детерминированности планировщик RTX может в любое время вытеснить подсистему Windows для запуска своих задач. Это достигается благодаря тому, что расширение RTX HAL перехватывает управление прерываниями у Windows HAL. Ни одно из прерываний Windows-среды не может прервать работу или замаскировать прерывание RTX. И, наоборот, во время работы RTX-задач маскируются все прерывания, обрабатываемые Windows, и приостанавливается работа Windows. Фактически, по отношению к задачам RTX вся ОС Windows работает в фоновом режиме.

Как уже упоминалось, подсистема RTSS продолжает работать в случае "краха" Windows. Она умеет корректно обрабатывать завершения Windows штатные и аварийные. Обработчики завершения работы в случае завершения по ошибке выполняются с небольшими ограничениями — им нельзя пользоваться сервисами Windows, в том числе запрашивать выделенные динамической памяти. Подсистема RTSS работает до тех пор, пока не завершатся все обработчики. С практической точки зрения обработчики должны корректно заканчивать работу с аппаратурой, оставлять рабочее состояние остальной системы, а также могут передавать управление резервной системе, если используется резервирование.

Невозможность обработчика завершения работы получить новый пул динамической памяти связана с тем, что RTX использует общую с Windows динамическую память. Запросы на выделение и освобождение памяти обслуживаются Windows, и потому в этой части поведение задачи RTX может быть недетерминированным. Это необходимо для корректной работы Windows и RTSS задач.

Естественно, такой способ работы с памятью не является единственным. Для обеспечения детерминированности поведения задач RTSS при операциях с памятью в RTX есть "локальный пул памяти" (Local Memory Pool). Этот объем памяти выделяется сразу при старте подсистемы RTSS, и задачи RTSS могут выделять себе необходимую память из этого пула. В этом случае работа с памятью является детерминированной.

Также в RTX имеется механизм, с помощью которого можно сделать доступной память, которую "не видит" Windows. Этой цели служат, в частности, функции PAE (Physical Address Extensions), расширяющие адресное пространство до 36 бит и позволяющие получать на 32-разрядных системах доступ к 64 Гбайт памяти.

Крайне важной особенностью является организация связи между задачами Win32 и RTSS. Механизмы взаимодействия процессов (IPC) являются привычными событиями, мьютексы, семафоры и общая память. При этом взаимодействие между детерминированной подсистемой RTSS и недетерминированной подсистемой Windows организовано так, чтобы исключить перекрестное влияние на их работу. Для этого механизм IPC реализован по принципу работы системы прерываний. Драйвер RTX со стороны Windows и окружение RTSS взаимодействуют через очереди. В каждом направлении организовано по две очереди и при необходимости иницилирующая сторона помещает в одну из очередей необходимую команду и прерывание запроса обслуживания (SRI — Service Request Interrupt) для обработки команды на другой стороне. Сервер на принимающей стороне обрабатывает запрос и помещает ответ в другой буфер. Такая организация взаимодействия обеспечивает отсутствие взаимных блокировок подсистем. Архитектура SRI направлена на достижение минимального времени отклика, что несколько снижает возможную пропускную способность обмена, но гарантирует, что запрос будет обработан в минимально возможное время.

Механизмом, направленным на обеспечение безопасности взаимодействия подсистем, являются прокси-процессы, относящиеся к воздействию из среды Windows на RTSS-окружение. Когда задача Win32 хочет получить доступ к объекту RTSS, создается так называемый прокси-процесс, с которым процесс Win32 взаимодействует непосредственно. А уже внутри себя RTX организует необходимое воздействие на реальную задачу RTSS. Создание, управление и освобождения прокси-процесса происходит полностью прозрачно для задачи Win32.

Самым важным свойством RTX 2009 стала поддержка для задач RTSS работы в режиме SMP – симметричной мультипроцессности. Версии с поддержкой SMP и без нее теперь есть как у SDK, так и Runtime-части. В итоге возможен запуск RTX в различных режимах.

1. *Режим без мультипроцессности* – RTX и Windows выполняются на одно- или многопроцессорной (многоядерной) системе. В этом случае возможны два варианта:

- *разделяемый*, когда RTX и Windows работают на одном ядре процессора, разделяя его ресурсы. Возможности такой системы довольно ограничены особенно при большой нагрузке. Такой вариант применим для систем с не очень большими требованиями к подсистеме РВ. Если нагрузка довольно серьезная, то лучше применить другой режим;

- *с выделенным процессором*, когда RTX занимает полностью одно процессорное ядро. Естественно, для этого необходимо, чтобы в системе было более одного ядра (или процессора). На остальных ядрах выполняется Windows. В такой системе RTX уже не требуется делить вычислительные ресурсы с Windows, потому здесь уже возможны реализации более серьезных систем РВ. В современных условиях, когда широко распространены процессоры с двумя и более ядрами, выделение отдельного ядра под RTX не влечет за собой каких-либо серьезных затрат.

2. *Режим с поддержкой SMP*, для реализации которого необходимо наличие нескольких ядер или процессоров. Режим SMP поддерживает выделение под задачи RTX 2...7 процессорных ядер, которые используются по возможности с полностью симметричной нагрузкой. На оставшихся ядрах будет выполняться ОС Windows. Реализация такого режима позволяет строить системы РВ, которые крайне требовательны к производительности и вычислительным мощностям и занимают очень мощными вычислениями.

Поддержка SMP в RTX реализована с оптимизацией под встраиваемые приложения. Разработчику доступны механизмы по управлению выполнением нитей/процессов на конкретном процессоре (или нескольких процессорах) и по обработке процессорами прерываний. Исходя из своих установок и свойств задач, назначенных разработчиком, RTX обеспечивает оптимальную загрузку выделенных под подсистему РВ ядер.

Для пользователя общение с RTX является простым и прозрачным. В составе RTX SDK имеется обширная документация по работе с RTX, скомпонованная в виде одного файла справки и охватывающая как объяснение структуры RTX, так и приемы программирования, настроек, включая полное описание API и требований к разработке программ.

Сфера влияния RTX весьма обширна, известно о более чем 750 тыс. применений продукта. В промышленности, медицине и телекоммуникациях пользователями RTX являются такие компании, как Siemens, ABB, Quanser, National Instruments, VMIC, General Electric. В аэрокосмической отрасли RTX используется, например, компаниями Raytheon, Boeing, Honeywell. Автомобильная промышленность также не обошла этот продукт вниманием: реализованные RTX-проекты есть у Porsche, у Renault, у Peugeot-Citroen и др. Хорошим практическим примером из промышленного сектора может служить проект Siemens по переводу своей Windows-платформы для контроллеров на базу RTX, реализация которого позволила значительно сократить затраты на портирование продукта в среду РВ и ускорить время разработки. В военной сфере RTX использовался, в частности, компанией Raytheon в системах обеспечения станций по тестированию ракет Patriot. Здесь RTX позволил отказаться от разработки и поддержки частных аппаратных решений, которые были заменены программным решением, работающим в режиме реального времени. В России расширение RTX чаще всего используется в промышленной автоматизации, а также системах контроля, тестирования и измерений.

Мечта, ставшая реальностью

В мире растет понимание того, что прибегать к традиционному ОС РВ во всех тех задачах, где требуется поддержка РВ – дорого и во многих смыслах неразумно. Поэтому встраиваемые проекты на базе ОС Windows демонстрируют положительную динамику распространения. Можно не сомневаться, что различные версии ОС Windows в комбинации с соответствующими версиями RTX будут успешно конкурировать с традиционными ОС РВ и в последующие годы.

Правильная стратегия выбора ОС должна выглядеть следующим образом: прежде чем покупать ОС РВ, удостоверьтесь в том, что нет возможности решить задачу с использованием "стандартной" ОС. Если никакая из разновидностей классических ОС Windows и Linux не подходит, подумайте над тем, как интегрировать в компьютер, работающий под управлением Windows/Linux подсистему РВ. И лишь в самую последнюю очередь, вполне убедившись, что ни первый, ни второй вариант вам не подходит, подумайте о том, как решить свою задачу при помощи ОС РВ. Сегодня на рынке предлагается относительно много ОС РВ, различающихся по производительности, сетевым возможностям, потреблению ресурсов, надежности, наличию сертификатов безопасности, адекватности свойств ряду вертикальных рынков типа авионики и т. п. И все же ОС РВ – это крайнее средство, к которому следует прибегать лишь тогда, когда все иные пути отрезаны.

*Исаев Алексей Валерьевич – Директор бизнес-направления системного ПО
Отделения базовых аппаратных и программных средств ЗАО "РТСофт",*

Акиншин Леонид Геннадьевич – канд. физ.-мат. наук, обозреватель журнала "МКА: мир ВКТ".

Контактные телефоны: (495) 967-15-05, 742-68-28. E-mail: pr@rtsoft.ru Http://www.rtsoft.ru