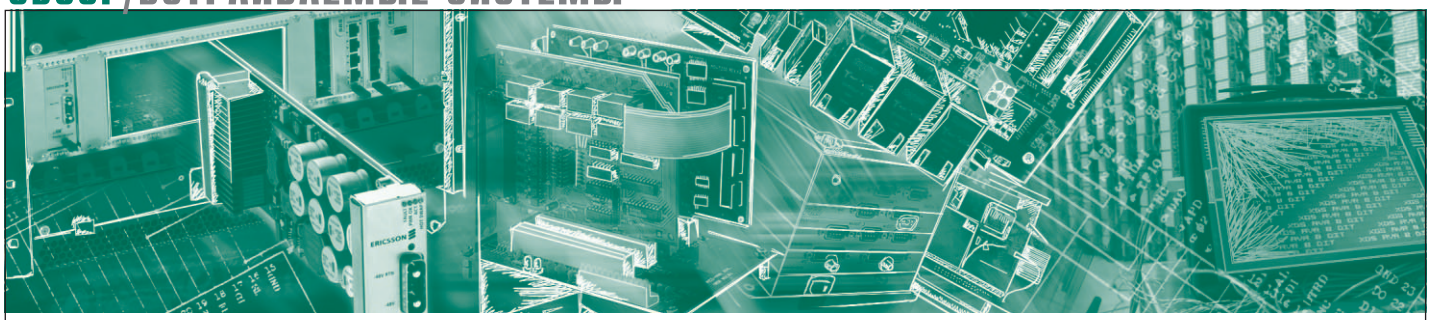

Николай Горбунов

О выборе встраиваемой ОС для проекта

В статье приводится обзор типовых требований и ограничений проектов по разработке встраиваемых приложений, а также их проекция на основные доступные характеристики встраиваемых ОС. Предлагается унифицированная система критериев сравнения ОС и варианты рекомендаций для конкретных случаев на примере ОС VxWorks, QNX, Wind River Linux, Windows Embedded и RTOS-32.

Проблема выбора, или Далеко ли до трактора

Есть мнение, что свобода – это отсутствие выбора; к процедуре выбора встраиваемой ОС этот тезис подходит как нельзя лучше. Действительно, если выбор из одного варианта всегда очевиден, то как только вариантов становится десять или больше (а число доступных на рынке в настоящий момент встраиваемых ОС измеряется десятками), встает вопрос оптимальности. Ситуация усугубляется тем, что каждый производитель всегда стремится показать положительные стороны своего продукта и завуалировать его ограничения, в результате получается, что один продукт самый производительный, другой самый компактный, а третий самый надёжный.

Как следствие, разработчикам приходится либо самостоятельно проводить сложную аналитическую работу и строить собственную систему критериев (пример такой системы, приведённой в аналитическом обзоре рынка встраиваемых ОС за 2010 год от компании VDC, представлен на рис. 1), либо (что происходит гораздо чаще) делать выбор иррационально, основываясь на моде, привычках, личных симпатиях и прочих факторах, к конечной задаче непосредственного отношения не имеющих. Между тем, опыт многих инженерных проектов подсказывает, что опрометчиво выбирать сердцем то, что

Рис. 1. Основные характеристики, используемые при выборе встраиваемой ОС (% респондентов, по данным опроса VDC за 2010 год)

будет использоваться руками, потому что, как известно, чем круче джип, тем дольше идти за трактором. Иными словами, соблазн сделать выбор побыстрее, чтобы сразу приступить к делу, всегда чреват необходимостью иметь дело с последствиями этого выбора на протяжении всей жизни проекта.

Рациональный выбор всегда сложнее, потому что чёрт, как известно, кроется в деталях, а деталей в технических системах много. Но и это ещё не всё: чтобы оценить применимость продукта в той или иной задаче, нужно не просто знать его характеристики, но и выразить их в терминах поставленного технического задания (ТЗ), потому что иначе получается «в огороде бузина, а в Киеве дядька». Настоящая статья подходит к выбору ОС «от задачи», предполагая, что в наличии имеются следующие исходные данные:

- требуемые характеристики системы/изделия;
- необходимые технологии;
- готовые наработки и опыт, имеющиеся у компании-разработчика;
- выбранное оборудование;
- целевой рынок системы/изделия;
- экономические ограничения проекта.

В каждой группе выделяются измеримые (а значит, позволяющие произвести сравнение) характеристики ОС, влияющие на конечный выбор, и приводится несколько конкретных иллюстраций. В качестве примеров встраиваемых ОС, доступных для выбора, используются:

- VxWorks и Wind River Linux американской компании Wind River;
- QNX канадской компании QNX Software Systems;
- Windows Embedded Standard и Windows Embedded Compact американской корпорации Microsoft;
- RTOS-32 немецкой компании On Time Informatik.

Итак, пойдем по порядку.

Выбор ОС и требуемые характеристики системы

Для любой системы существуют критерии качества, которым она должна соответствовать, чтобы успешно выполнять свою задачу. Критерии эти в различных случаях индивидуальны и могут включать в себя пропускную способность, допустимое время внепланового простоя, параметры удобства интерфейса, скорость обработки внешнего события и т.п. Это измеримые величины, которые прописываются в ТЗ и контролируются на приёмо-сдаточных испытаниях; возможность их обеспечить тесно связана с техническими характеристиками используемой ОС. Со стороны ОС в число заявляемых производителями характеристик обычно входят:

- архитектурные принципы (тип ядра и планировщика, модель многозадачности и т.п.);
- временны́е характеристики (время реакции на прерывание, время переключения контекста, время перепланирования и т.п.);
- поддержка реального времени («жёсткое», «мягкое» или отсутствует);
- показатели надёжности (например, среднее время восстановления после отказа);
- метрики производительности (например, пропускная способность файловой системы на заданном носителе).

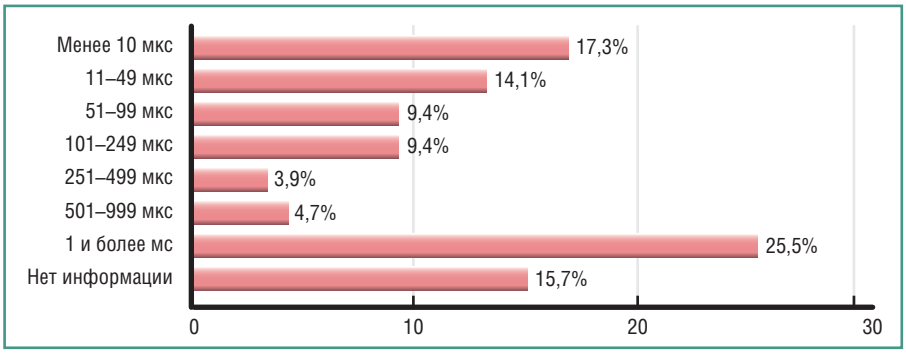

Рис. 2. Технические требования к времени реакции на прерывание для текущего проекта (% респондентов, по данным опроса VDC за 2010 год)

Все эти показатели могут использоваться для оценочных расчётов конечных свойств системы, причём полезными здесь могут оказаться не только численные характеристики, но и общие архитектурные принципы, потому что многие свойства ОС непосредственно вытекают из её архитектуры (как принято говорить у химиков, строение определяет свойства).

К примеру, чтобы оценить, справится ли система с расчётной нагрузкой, нужно обращать внимание на время реакции и метрики производительности ОС, а также на их детерминированность, поскольку обработать событие быстро – это одно дело, а обработать его *вовремя* – совсем другое. Если в исходной задаче жёстко регламентировано максимальное время отработки внешнего события, то для решения задачи необходима ОС «жёсткого» реального времени, временны́е характеристики которой детерминированы; в противном случае можно обойтись ОС «мягкого» реального времени или ОС общего назначения. Примечательно, что по результатам опросов в большинстве проектов от ОС требуется либо очень медленная (от единиц мс), либо, наоборот, очень быстрая (до десятков мкс) реакция (рис. 2).

Аналогично, когда речь идет об оценке надёжности, следует иметь в виду, что надёжность программной системы складывается из двух основных составляющих:

- *безотказности;*
- *отказоустойчивости.*

Безотказность, в свою очередь, складывается из безотказности как самой ОС, так и прикладного кода, а значит, зависит не только от ОС, но и от принятого для нее инструментария прикладного программирования (чем совершеннее инструментарий разработки, отладки и диагностики, тем больше у разработчика возможностей написать код с минимальным количеством дефектов). К факторам, определяющим безотказность, можно также отнести используемый в ОС механизм защиты от проблемы инверсии приоритетов, которой страдают все ОС с вытесняющим планировщиком, – если ОС не умеет корректно обрабатывать ситуацию инверсии приоритетов, то некорректно спроектированное взаимодействие программных компонентов может стать причиной потери детерминированности, что в системах жёсткого реального времени зачастую ведёт к катастрофе.

Что касается отказоустойчивости, то здесь большую роль играет архитектура ОС. В частности, ОС на основе микроядра считаются более отказоустойчивыми за счёт того, что в кольце ядра (а значит, с расширенными привилегиями) выполняется только критический системный код; аналогично, повышенную отказоустойчивость обеспечивают защита памяти и другие механизмы

Таблица 1

Сравнительные характеристики встраиваемых ОС

ХАРАКТЕРИСТИКИ	ВСТРАИВАЕМЫЕ ОС					
	QNX Neutrino	Wind River VxWorks	Wind River Linux	Windows Embedded Standard	Windows Embedded Compact	On Time RTOS-32
Архитектура						
Тип ядра	Микроядро	Гибридное	Монолитное	Гибридное	Гибридное	Монолитное
Защита памяти	Да	Настраиваемая	Да	Да	Да	Настраиваемая
Реальное время	Жёсткое	Жёсткое	Мягкое/жёсткое	Нет	Жёсткое	Жёсткое
Время отклика	Единицы мкс	Единицы мкс	Единицы мкс	Не регламентировано	Десятки мкс	Сотни нс
Количество уровней приоритета	64	256	140	32	256	64
Дисциплины планирования	FIFO, карусельная, спорадическая	Карусельная	FIFO, карусельная, адаптивная по типу HPRN*	Карусельная, адаптивная по типу HPRN*	Карусельная	Карусельная, адаптивная по типу HPRN*
Поддержка квотирования ресурсов процессора (ARINC 653)	Да	Да	Нет	Нет	Нет	Нет
Требуемый объем ПЗУ	Единицы Мбайт	Сотни кбайт	Десятки Мбайт	Сотни Мбайт	Десятки Мбайт	Сотни кбайт
Возможность динамического восстановления при отказе	Любые системные и пользовательские процессы	Только пользовательские процессы	Только пользовательские процессы	Только пользовательские процессы	Только пользовательские процессы	Только система целиком
Поддерживаемые технологии (в том числе доступные через партнёрскую сеть)						
Поддержка многопроцессорности	SMP, AMP	SMP, AMP	SMP, AMP	SMP	Нет	SMP
Стек IPv4	Да	Да	Да	Да	Да	Да
Стек IPv6	Да	Да	Да	Да	Да	Нет
Средства графического интерфейса	Штатная графическая оболочка, Qt, Adobe Flash Lite	Штатная графическая библиотека, Qt, пакет Tilcon Graphics Suite	Графические оболочки X.org и GNOME, пакет Tilcon Graphics Suite, Qt, Adobe Flash Lite	Штатная графическая оболочка, Qt, Adobe Flash, Microsoft Silverlight	Штатная графическая оболочка, Qt, Adobe Flash Lite, Microsoft Silverlight	Штатная графическая библиотека
Поддержка мультимедиа	Да	Да	Да	Да	Да	Нет
Функции управления энергопотреблением	Да	Да	Да	Да	Да	Нет
Поддержка реляционных баз данных	Да	Да	Да	Да	Да	Нет
Поддержка OPC/ OPC-шлюзов	Да	Да	Да	Да	Да	Нет
Поддержка эмуляторов ПЛК (Soft-PLC, МЭК 61131-3)	Да	Да	Да	Да	Да	Да
Поддержка промышленных шин	Да	Да	Да	Да	Да	Да
Инструментарий разработчика						
Интерфейс прикладного программирования (API)	POSIX	POSIX	POSIX	Win32	Win32	Win32
Интегрированная среда разработки	Штатная IDE на базе Eclipse	Штатная IDE на базе Eclipse	Штатная IDE на базе Eclipse	MS Visual Studio, Visual Studio Express	MS Visual Studio, Embedded Visual C++	MS Visual Studio, MS Visual C++, Borland C/C++, Borland C++ Builder, Borland Delphi
Поддерживаемые языки программирования/ моделирования	C/C++ , Java, UML, Python, Ruby, Ada, Fortran	C/C++ , Java, UML, Ada, Fortran	C/C++ , Java, UML, Python, Ruby, Ada, Fortran, Pascal	C/C++, C#, Visual Basic .NET, F#, Ruby, Java, UML, Ada, Fortran, Pascal	C/C++, C#, Visual Basic .NET, Java, UML, Python, Ruby, Fortran, Pascal	C/C++ , Ada, Pascal
Поддержка JTAG-отладчиков	Да	Да	Да	Нет	Да	Нет
Поддерживаемые инструментальные ОС	Windows, Linux	Windows, Linux, Solaris	Windows, Linux, Solaris	Windows	Windows	Windows
Поддерживаемое оборудование						
Поддерживаемые архитектуры процессоров	x86, ARM, MIPS, PowerPC	x86, ARM, MIPS, PowerPC, ColdFire	x86, ARM, MIPS, PowerPC, SPARC	x86	x86, ARM, MIPS, SuperH-4	x86
Типовые применения и сертификация						
Типовые применения	Ответственные системы, авиация/ космонавтика, промышленные контроллеры, военные приложения, коммуникационные устройства, медицинские приборы	Ответственные системы, авиация/ космонавтика, промышленные контроллеры, военные приложения, коммуникационные устройства, медицинские приборы	Коммуникационные устройства, мобильные устройства, потребительская электроника, промышленные и военные приложения	Операторские терминалы, точки обслуживания, информационные киоски, торговые автоматы	Связанные мобильные устройства, потребительская электроника, операторские терминалы, промышленные контроллеры, медицинские приборы, мультимедийные устройства	Промышленные контроллеры, робототехника, транспорт, станки с ЧПУ
Сертификаты и сертификационные пакеты	МЭК 15408 («Общие критерии») EAL 4+, МЭК 61508 SIL 3	МЭК 15408 («Общие критерии») EAL 4/4+/6+, DO-178B уровень A, МЭК 61508 SIL 3, CENELEC EN 50128, FDA 510(k)	CGL 4.0, МЭК 15408 («Общие критерии») EAL 4+	Нет	Нет	Нет
Лицензирование						
Доступность исходного текста	Полностью	Полностью	Полностью	Нет	Частично	Полностью
Лицензионные отчисления	Да	Да	Нет	Да	Да	Нет

* Дисциплина планирования HPRN (Highest Penalty Rate Next) подразумевает, что процессор предоставляется задаче, которая на настоящий момент наиболее «обделена» процессорным временем.

изоляции компонентов, позволяющие, во-первых, локализовать отказ, а во-вторых, восстановить нормальное функционирование отказавшего компонента, не прерывая работу всей системы.

Примеры выбора встраиваемой ОС по заданным характеристикам (табл. 1)

Пример 1. Согласно ТЗ система должна обрабатывать входной аналоговый сигнал с заданной граничной частотой спектра, обеспечивая заданную разрешающую способность. С учётом выбранного оборудования анализ показал, что для корректной обработки сигнала без потери отсчётов необходимо обеспечить время реакции на прерывание «прерывание – поток» не более 8 мкс. Временем отклика, способным обеспечить гарантированную обработку в таких ограничениях, обладают ОС VxWorks, QNX и RTOS-32.

Пример 2. ТЗ содержит ограничения на время внепланового простоя системы в минутах в год (так называемый коэффициент готовности). С учётом выбранного оборудования анализ показал, что необходимо обеспечить возможность восстановления отказавшего программного компонента за время, не превышающее 100 мс. Учитывая, что время перезагрузки ОС обычно составляет не менее единиц секунд, необходимо применение ОС, способной восстанавливать отказавшие компоненты индивидуально, без полной перезагрузки. Таким свойством обладает ОС QNX.

Выбор ОС и необходимые технологии

Чтобы реализовать требуемую функциональность, приложению может потребоваться поддержка тех или иных технологий, например, организация OPC-туннелирования для интеграции со SCADA-приложением или поддержка маршрутизации OSPF. Перечень необходимых технологий обычно является следствием требуемой функциональности приложения и указывается в ТЗ; его можно и нужно сравнивать с технологическими возможностями ОС в процессе выбора. Описание технологических возможностей ОС, в свою очередь, складывается из ответов на следующие вопросы:

- что входит в дистрибутив (то есть какую функциональность ОС предлагает «из коробки»);
- что доступно через партнёрскую сеть производителя (то есть что разработано сторонними компаниями и приобретается отдельно).

Из того, какие технологии поддерживаются данной ОС (стеки протоколов, базы данных, 3D-графика, управление энергопотреблением, Java и т.п.), непосредственно следует, какую часть требуемой функциональности можно получить в готовом виде, какую придётся дополнительно приобретать, а какую необходимо разрабатывать (или переносить) самостоятельно.

Примеры выбора встраиваемой ОС по поддерживаемым технологиям (табл. 1)

Пример 3. Согласно ТЗ система должна обеспечивать высокий уровень детерминизма, для чего предлагается использовать многоядерную вычислительную среду и назначить ряду задач выделенные ядра. С точки зрения технических требований, это требует от ОС поддержки асимметричной многопроцессорности (Asymmetric multiprocessing – AMP), её поддерживают не все ОС. В нашем случае для реализации задачи подойдут ОС VxWorks, QNX и Wind River Linux.

Пример 4. Согласно ТЗ система будет содержать сторонние компоненты, разработанные на Java, требуемый профиль Java – J2EE. Большинство встраиваемых ОС поддерживает только профиль J2ME; в данном случае для решения задачи подойдет только ОС Windows Embedded Standard.

Выбор ОС и имеющиеся наработки/опыт

Любое предприятие состоит из людей. Эти люди обладают определённым опытом и квалификацией, используют привычные инструменты и технологии и за время существования предприятия накопили некоторое количество наработок, в частности, алгоритмов, реализованных в существующей кодовой базе. Алгоритмы эти описаны на определённых языках с использованием соответствующего интерфейса прикладного программирования (Application Programming Interface – API), и перенос их на другой язык/API означает потенциальное внесение ошибок в стабильный отлаженный код. Соответственно, при выборе ОС для проекта нужно чётко понимать, какая ОС способна принять имеющиеся опыт и наработки с минимальной модификацией существующего кода и минимальным обучением персонала.

В разрезе API мир встраиваемых ОС чётко поделён на два основных лаге-

ря – POSIX и Win32. (Естественно, поскольку не все тонкости программирования удачно описаны в стандартах, везде существуют свои частнофирменные расширения, но основа почти всегда строится либо на POSIX, либо на Win32.) Соответственно, например, если большинство наработок предприятия написано с использованием POSIX API (скажем, в среде настольной Linux), то перенести кодовую базу на POSIX-совместимую встраиваемую ОС (QNX, VxWorks, Wind River Linux) будет намного проще; аналогично дело обстоит с проектами, реализованными с использованием Win32 API.

Кроме API, при выборе ОС будет также играть роль, хоть и меньшую, набор поддерживаемых для неё средств разработки: компиляторов, отладчиков, интегрированных сред, диагностического инструментария, средств визуального моделирования и т.п. Один и тот же инструмент может поддерживать несколько целевых ОС, но ни один не поддерживает все сразу; из того, какие средства разработки доступны, можно понять, как для данной ОС можно разрабатывать (или переносить) код. Большинство средств разработки для POSIX-совместимых встраиваемых ОС базируется на линейке GCC и платформе Eclipse; в Win32 разработка ведётся преимущественно в Microsoft Visual Studio. Здесь, как и в случае с API, играет роль простой принцип: что привычнее, то не потребует длительного переучивания, а значит, позволит быстрее начать продуктивную деятельность, плюс, если кодовая база написана не на C/C++ (эти языки наиболее популярны во встраиваемых приложениях и широко поддерживаются), нужно иметь в виду наличие для выбираемой встраиваемой ОС соответствующего транслятора.

Примеры выбора встраиваемой ОС с сохранением имеющихся наработок (табл. 1)

Пример 5. Согласно ТЗ интерфейс оператора системы реализован в SCADA-приложении GENESIS32 (без web-компонентов). Необходимо обеспечить работу этого интерфейса во встраиваемой среде. GENESIS32 существует только для Windows; соответственно, во встраиваемом приложении ему будет необходима ОС Windows Embedded Standard.

Пример 6. Алгоритмы, требуемые для работы изделия, были разработаны и отлажены в среде настольной Linux; необходимо перенести их во встраиваемую вычислительную систему. Применительно к выбору встраиваемой ОС это означает поддержку POSIX API с Linux-специфичными расширениями при минимальной ресурсоёмкости; лучше всего в данном случае подходит ОС Wind River Linux.

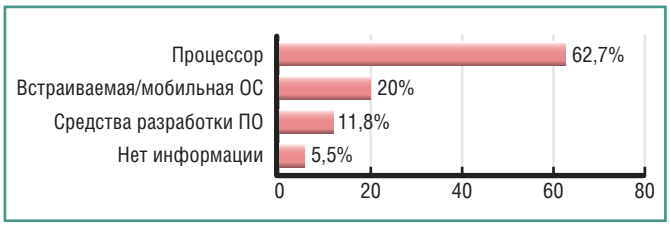

Рис. 3. Продукты, выбираемые для проекта в первую очередь (% респондентов, по данным опроса VDC за 2010 год)

Выбор ОС и оборудование

В большинстве проектов оборудование выбирается первым, поскольку именно от его характеристик зависит физическая возможность реализации поставленной задачи (это подтверждается результатами опросов – рис. 3). Таким образом, на момент принятия решения о выборе ОС список оборудования обычно уже известен, что позволяет сразу оценить, во-первых, требуемую ресурсоёмкость ОС, а во-вторых, объём работ по её адаптации (они могут включать в себя разработку драйверов, BSP, а при необходимости и портирование ядра). Совместимость ОС с оборудованием складывается из трёх факторов:

- совместимость на уровне архитектуры процессора;
- совместимость на уровне периферийных устройств;
- требуемый объём ресурсов (то есть необходимая вычислительная мощность и объём занимаемой памяти).

Обеспечение совместимости на уровне процессора требует глубокой переработки ядра ОС, поэтому список поддерживаемых процессоров обычно заявляется производителем ОС как данность и меняется редко. Обеспечение поддержки периферийных устройств сопряжено с гораздо меньшим количеством трудностей, и наряду с пакетом готовых драйверов и пакетом поддержки процессорных плат (Board Support Package – BSP), которые входят в дистрибутив ОС, производители обычно в том или ином виде предоставляют инструментарий для их самостоятельной разработки. Разумеется, поскольку у каждой ОС своя архитектура, то и подход к разработке и отладке драйверов и BSP у каждой ОС будет разным; чем этот подход проще и прозрачнее, тем быстрее можно адаптировать ОС к требуемой аппаратуре. При этом, естественно, не надо забывать, что ОС должна «влезать» в предоставленные ей объём памяти и вычислительную мощность, так как в условиях недостатка ресурсов совместимость не имеет смысла.

Таким образом, при выборе встраиваемой ОС для имеющегося оборудования нужно обращать внимание на следующие её характеристики:

- список поддерживаемых процессорных архитектур;
- заявленную ресурсоёмкость;
- список поддерживаемых периферийных устройств и готовых BSP;
- архитектуру и методику разработки драйверов и BSP;
- наличие и состав предлагаемых комплектов разработки драйверов и BSP;
- доступность технической поддержки и сервисов разработки драйверов и BSP на заказ.

Первые три пункта описывают совместимость «из коробки», последние три – возможность произвести адаптацию ОС, если совместимость «из коробки» окажется недостаточной.

Примеры выбора встраиваемой ОС под заданное оборудование (табл. 1)

Пример 7. Разрабатываемое устройство представляет собой беспроводную IP-камеру видеонаблюдения; выбранный процессор – Freescale MCF54455. Данный процессор относится к семейству ColdFire и поддерживается только ОС VxWorks.

Пример 8. Выбранное оборудование позволяет выделить программному обеспечению не более единиц Мбайт ОЗУ. Расчёт показал, что из них для ОС будет доступно порядка 300 кбайт. В таких объёмах ОЗУ способны работать только ОС VxWorks или RTOS-32.

Выбор ОС и целевой рынок

Про любое приложение всегда заранее известно, для какого целевого рынка оно предназначается; из этого автоматически следует, каким отраслевым стандартам оно (и, возможно, его компоненты) должно соответствовать.

Если требование соответствия стандартам распространяется на программные компоненты (а значит, и на ОС), то выбор ОС будет зависеть от того, какие сертификаты и/или сертификационные пакеты для неё существуют. У каждой ОС этот перечень особый (он диктуется рыночной политикой производителя, поскольку именно производитель решает, на какие вертикальные рынки позиционировать свой продукт), и именно этим перечнем определяется, в каких приложениях данную ОС применять можно, а в каких нельзя.

Также, кроме нормативных документов, в каждой отрасли существуют сложившиеся традиции и тенденции; фактически это та же система стандартов, основанная на опыте успешных внедрений, но не оформленная в виде официальных документов. Эти традиции всегда можно проследить по открытым подборкам историй внедрения, публикуемых производителями ОС (хотя предоставляемая ими картина зачастую является неполной, потому что далеко не все производители встраиваемых приложений готовы раскрывать информацию о «начинке» своих изделий).

Примеры выбора встраиваемой ОС для заданного вертикального рынка (табл. 1)

Пример 9. К разрабатываемому устройству предъявляются повышенные требования по функциональной безопасности, в частности, ТЗ вводит ограничения на вероятность опасного отказа за час непрерывной работы. Данная постановка задачи подразумевает сертифицируемость по МЭК 61508 и требует наличия у применяемой ОС соответствующего сертификационного пакета. Сертификационные пакеты МЭК 61508 существуют для ОС QNX и VxWorks.

Пример 10. Разрабатываемое устройство представляет собой коммутатор операторского класса. Программные решения для устройств такого типа хорошо представлены в Linux, но требования индустрии подразумевают соответствие используемого дистрибутива Linux спецификации CGL. Сертификатом соответствия CGL 4.0 обладает ОС Wind River Linux.

Выбор ОС и экономические ограничения проекта

План проекта всегда включает в себя бюджет, плановые объёмы производства и расчёт себестоимости, и в зависимости от применяемых моделей лицензирования и ценообразования даже абсолютно технически приемлемая для проекта ОС может не пройти по экономическим ограничениям.

Тонкость здесь заключается в том, что, несмотря на общий для встраиваемых ОС принцип разделения на средства разработки (development kit) и среду исполнения (runtime), модели лицензирования и принципы ценообразования у различных продуктов могут сильно отличаться. В одном случае комплект разработчика приобретается однократно, а техническая поддержка и обновления версий предоставляются по подписке, в другом сама среда разработки предоставляется по подписке на условиях «всё включено». У одних продуктов среда исполнения бесплатна, и их можно тиражировать неограниченно, другие требуют лицензионных отчислений за каждую копию. Таким образом, чтобы адекватно сравнить экономическую целесообразность применения различных ОС в заданном проекте определённого предприятия, их следует рассматривать с точки зрения стоимости жизненного цикла (Total Cost of Ownership – TCO), то есть всех затрат, которые организация понесёт в процессе использования конкретной ОС на протяжении всего проекта. TCO, в свою очередь, будет складываться из:

- стоимости лицензии на средства разработки;
- стоимости заказных работ по адаптации ОС к оборудованию (если они выполняются сторонней организацией);
- стоимости лицензии на среду исполнения ОС (умноженной на плановый тираж системы/изделия);
- стоимости сопутствующих сервисов (обучения персонала, технической поддержки, обновления версий и т.п.).

Строго говоря, при расчете TCO следует также учитывать стоимость оплаченных человеко-часов, затраченных на установку, настройку и обслуживание ПО, но здесь эти затраты не рассматриваются, так как сильно зависят от конкретного случая и поэтому трудно поддаются формальной оценке.

Примеры выбора встраиваемой ОС в заданных экономических ограничениях (табл. 1)

Пример 11. Разрабатываемое устройство предполагается выпускать в больших количествах, соответственно, важную роль играет минимизация себестоимости. Это налагает ограничения на стоимость среды исполнения применяемой ОС, и предпочтение будет отдано ОС с низкими или нулевыми лицензионными отчислениями. К таким ОС относятся Wind River Linux, RTOS-32 (без лицензионных отчислений) и Windows Embedded Compact (стоимость лицензии минимальна).

Пример 12. Предприятие-разработчик является многопрофильным и создаёт множество линеек продукции одновременно. Соответственно, использование ОС высокой степени универсальности с единым инструментарием позволило бы сократить сроки обучения при миграции специалистов с проекта на проект и снизить затраты на обслуживание ПО. Высокой универсальностью обладают ОС VxWorks и Wind River Linux.

Выводы

Каждая задача уникальна, и свести все факторы к единой формуле, выдающей универсальный ответ в заданных ограничениях, конечно же, невозможно. Однако, чтобы сделать оптимальный выбор, важно не потерять основные ключевые моменты, для этого полезно иметь под рукой список параметров, на которые обязательно нужно обратить внимание. В противном случае выбор будет либо однобоким, либо иррациональным, а ни то ни другое, как известно, успеху проекта не способствует (читай: впоследствии придётся в самый неподходящий момент идти за трактором).

В данной статье был предложен всего лишь краткий обзор важных параметров, влияющих на выбор встраиваемой ОС для проекта; более подробную сравнительную таблицу встраиваемых ОС с указанием их основных характеристик (в терминах классификации, используемой в данной статье) можно скачать с web-сайта компании ПРОСОФТ по адресу: www.prosoft.ru/rtos/. Там же можно найти подробные описания встраиваемых ОС и средств разработки для них, а также ссылки на полезные web-ресурсы (списки поддерживаемого оборудования, пробные версии и т.п.). ●

Автор – сотрудник фирмы ПРОСОФТ
Телефон: (495) 234-0636
E-mail: info@prosoft.ru