

Сравнительный анализ: MINIX3 / QNX6

Цилюрник О.И.

< olej@front.ru >

от 01.01.2010

Редакция 1.07

Оглавление

Технические детали.....	2
Микроядро.....	2
Обмен сообщениями.....	2
Надёжность, живучесть и реальное время.....	2
Средства высокой готовности.....	3
Объём и полнота.....	3
Состав программных инструментов.....	3
Проработанность API (POSIX).....	3
Обозримость и обучение.....	4
Поддерживаемые аппаратные платформы.....	4
Не технические факторы сравнения.....	4
Производитель / разработчик.....	4
Дистрибуция.....	5
Стоимость.....	5
Лицензия.....	6
Внедрения.....	6
Выводы.....	6
Аргументы в пользу использования MINIX.....	6
Первостепенные задачи развития MINIX.....	7
Источники информации.....	7

Операционные системы MINIX (v3) и QNX (v6) — микроядерные по архитектуре, и очень близки друг-другу, если сказать «близнецы-братья», то это не будет большим преувеличением, по крайней мере, если соотнести это с той разительной разницей, которая отличает их от других, моноядерных ОС. Исходить в сравнении нужно, в первую очередь, именно из этого: то, что будет присуще прикладным системам в одной ОС — будет, пусть и с некоторыми отличиями, присуще и аналогичным системам в другой ОС. Естественно, с некоторыми особенностями в каждой системе. Но отличия будут намного меньше, чем если бы аналогичные программные системы функционировали под управлением любой из моноядерных ОС.

Поэтому первоначально рассмотрим набор, достаточно широкий, технических отличий MINIX и QNX, которые будут иметь свой вес в альтернативном выборе той или иной ОС. В техническом сравнении станет ясно: а). что в паре «близнецов-братьев» QNX, несомненно, является старшим братом (не по возрасту, а по широте охвата), и б). что при намного большем разнообразии механизмов QNX, большинство из них можно воспроизвести и под MINIX, но это потребует некоторой изобретательности.

Далее, помимо набора технических показателей (и особенно - в силу не столь разительных технических отличий) — на выбор должны сильно влиять и другие аспекты, внетехнического характера: стоимостные, маркетинговые, политические и другие. К оценке таких показателей, и прогнозам соответствующих рисков, вернёмся, но уже после рассмотрения технических деталей.

Технические детали

Для начала я зафиксирую технические отличия двух ОС (которые и определяют эти «некоторые отличия»), а только после этого проведём сравнительную оценку по широкому кругу других критериев.

Микроядро

Микроядерная архитектура присуща и той и другой ОС. И именно на этом основывается декларируемая экстремально высокая надёжность. В QNX микроядро объединено в одно адресное пространство (задача-файл **procnto**) с а).менеджером процессов и б).менеджером памяти. В MINIX менеджер процессов (процесс **PM**) объединённый с менеджером памяти (процесс **MM**) находятся вне микроядра, в некотором смысле можно считать, что эта ОС является несколько (незначительно) «более микроядерной», и, возможно (?), это обеспечит ещё некоторый дополнительный запас надёжности. Все остальные программные компоненты (драйверы устройств, файловые системы, сетевые подсистемы) и в той и в другой системе работают вне адресного пространства микроядра.

Обмен сообщениями

Взаимодействие задач производится обменом сообщений. Адресация сообщений реализована несколько по-разному. QNX **MsgSend(...)** имеет 3-х компонентный адрес назначения: hid (host ID) — pid (process ID) — chid (channel ID). MINIX **send(...)** имеет 1-но компонентный адрес: endpoint (примерный аналог PID). Адресация QNX с hid позволяет создать однородную систему рассылки сообщений в сети, и совершенно уникальную (по сравнению со всеми ОС) распределённую вычислительную среду. В MINIX такой возможности нет, но а).она может быть добавлена «по аналогии» и б).предложены другие, косвенные механизмы отправки сообщений по сети.

Принципиальной разницей подходов к обмену сообщениями является то, что:

- в QNX любой процесс (клиент) может посылать сообщения любому процессу (сервер), возможна, в принципе, и обратная посылка...
- в MINIX сообщения могут посылаться только специальным процессам, написанным как сервер;
- в результате этого, при **неправильном проектировании** прикладной системы в QNX, процесс А может послать запрос процессу Б, для обслуживания которого процесс Б (прямо, или косвенно, через некоторые промежуточные процессы) пошлёт запрос процессу А; в этом случае целевая система войдёт в мёртвый клинч (deadlock);
- в MINIX такая ситуация невозможно (из-за однонаправленности, о которой сказано выше); но за такую «страховку» придётся расплачиваться гораздо более изощрённым архитектурным программированием целевой системы.

Надёжность, живучесть и реальное время

QNX декларируется как RealTime OS (RTOS – Real Time OS), в ней реализуются практически все расширения POSIX 1003. Но главное в обеспечении свойств реального времени — это а). реализация дисциплин планирования потоков (POSIX): FIFO, round-robin (RR), sporadic (пришедшая на смену adaptive), и б). наследование приоритетов при передаче сообщений микроядра.

MINIX3 не позиционируется отдельно в качестве RTOS. В MINIX3 реализована в качестве дисциплины планирования некоторая модификация RR с элементами, свойственными sporadic. По крайней мере, дисциплина планирования параллельных процессов не предусматривает никакого «динамического перепланирования», свойственного системам общего назначения (GPOS – General Purpose OS), например, Windows или Linux. При такой организации планирования не видится принципиальных сложностей «не детерминированности», свойственных GPOS, которые могли бы возникать в MINIX3.

В отношении других механизмов, которые обычно соотносят с RTOS, таких например, как инверсия приоритетов [9], должны быть проведены дополнительные изучения и тестирования, отдельно по каждому механизму. В случае выявления присутствия отрицательных факторов, может быть проделана работа по модификации MINIX3, благо весь код открыт, и легко подлечит полной сборке из исходных кодов.

Некоторые авторы [8] вообще весьма критически относятся к терминологии «реальное время», утверждая, что, в первую очередь, для систем, позиционируемых как RTOS, требуется сконцентрироваться на понятиях: предсказуемость, детерминированность, надёжность, живучесть, но безотносительно к «времени». Именно с таких точек зрения и построено наше дальнейшее рассмотрение.

Вот по этой ссылке: <http://www.rts.ua/rus/articles/360/20/41/%20?PHPSESSID=54ff3609e9c233eb1339e0a9c8d6f3b1> — размещена статья Поля Леру (Paul N. Leroux, технический аналитик компании QNX Software Systems) и Ромейна Саха (Romain Saha, менеджер сегмента сетевых систем компании QNX Software Systems), «Реальное время или реальная Linux? Делаем трезвый выбор» от разработчиков QNX, в которой формулируются критерии реального времени, и из неё же и вытекает, что критерии реального времени могут присутствовать в любой хорошо спроектированной системе, и сводятся к высокой надёжности, готовности, предсказуемости и живучести (что, собственно, и обеспечивается архитектурой микроядра).

Средства высокой готовности

Средства обеспечения высокой готовности присутствуют в обеих системах. В MINIX — это присутствующий изначально в системе сервер реинкарнаций (**RS**). В QNX это добавленный позже монитор высокой готовности HAM (High Available Monitor, <http://www.swd.ru/index.php3?pid=379>). Принцип один и тот же: мониторинг критических процессов, и в случае их неработоспособности — перезапуск новой копии процесса.

К средствам обеспечения высокой готовности MINIX3 можно отнести и малое время (в сравнении с другими ОС, и, в частности, и QNX) полной перезагрузки ОС, это может потребоваться только в предельно экстремальных ситуациях (например, при отслеживании по watch-dog), но способствует поддержанию более высокого коэффициента готовности целевой системы. В принципе, целесообразно даже провести небольшое изучение, направленное на уменьшение времени приведения системы в рабочее состояние (изменение набора модулей образа системы, смена порядка их загрузки ...).

Объём и полнота

В этой части мы сконцентрируемся на том, какие компоненты уже есть готовый в составе каждой из ОС.

Состав программных инструментов

Широкий набор программных утилит и средств, предусматриваемых POSIX и портированных из GNU — представлены и в QNX и в MINIX3 в весьма широкой, и практически в одинаковой номенклатуре (отличаясь, естественно в деталях: опции запуска, покрываемые возможности...).

Относительно программных компонент, уже наработанных в составе QNX, аналоги которых ещё не созданы для MINIX3, то здесь перечислим только целевые **группы** таких программных средств, которые присутствуют в QNX, и функциональные аналоги которых отсутствуют в MINIX:

- «лёгкая» графическая система (GUI) Photon — в дополнение (или скорее, применительно к QNX - взамен) к тяжеловесной UNIX GUI на основе X-сервера. Изделие достаточно полезное, но первейшая цель его создания, так видится, всё таки создание **закрытого** GUI (в отличие от X), проприетарного и коммерческого. Это вторая радикальная попытка со стороны QSSL создать GUI, отличный от X: первым был Qwindows QNX v.4, исторически ушедший в небытие.
- визуальная система проектирования приложений для Photon — PhAB (Photon Application Builder). Весьма удобное средство проектирования GUI-приложений, но применительное исключительно для GUI Photon (см.пред.п.).
- адаптированная под QNX широко известная визуальная система проектирования приложений Eclipse; правда, начиная с версии 6.4 и далее, такая адаптированная версия есть только для кросс-разработки приложений QNX из-под операционных систем Windows или Linux.
- специфический сетевой протокол QNET, и построенные на его базе сетевые средства обмена сообщениями

микроядра в сетевой среде; это предоставляет уникальные возможности для кластерной, распределённой обработки.

Проработанность API (POSIX)

Обе ОС декларируют совместимость API с POSIX — и в этом тоже проявляется общность подходов. Но совместимость с POSIX — это всегда предполагает реализацию только какого-то подмножества POSIX (это иллюстрируется, например, и Linux, и Sun Solaris).

QNX предлагает практически полную реализацию нескольких POSIX стандартов, включая и расширения POSIX реального времени (1003.b, 1003.g, 1003.h).

MINIX3 представляет ограниченное, но весьма широкое подмножество POSIX API. К числу ограничений предложений POSIX, не представленных в MINIX3 относятся:

- расширения реального времени (1003.b, 1003.g, 1003.h), что совершенно естественно, так как MINIX не декларирует свойств реального времени;
- всё, что связано с потоками выполнения (`pthread_t`) в общем адресном пространстве, и, соответственно, связанные понятия: мютексы (`pthread_mutex_t`), условные переменные (`pthread_cond_t`), ждущих блокировок (`pthread_sleepon_t`), барьеров (`pthread_barrier_t`), блокировок чтения/записи (`pthread_rwlock_t`), спин-блокировок (`pthread_spinlock_t`). Но все эти механизмы главным образом предназначены для синхронизации потоков, которых просто нет в MINIX — параллелизм MINIX обеспечивается на уровне процессов и механизма `fork()`. Кроме того, большинство этих примитивов синхронизации и определены в расширениях реального времени (1003.b, 1003.g, 1003.h), о чём уже обсуждено выше.

Обозримость и обучение

Увеличение объёма и возможностей системы, что обсуждалось выше, увеличивает её мощь, но имеет и две оборотные стороны:

- обозримость: при некотором объёме возможностей системы (утилиты, API, протоколы, ...) выше оптимального (достаточного) становятся уже трудно обозримые для единого разработчика, и в прикладных системах начинают закладываться не оптимальные механизмы;
- обучение: с ростом громоздкости системы увеличивается время её изучения, и втягивания в проект новых (молодых) исполнителей, что может стать существенным тормозом крупных проектов.

Операционная система QNX v.6 содержит множество компонент «на все случаи жизни». Но в этом и её слабость — система определённо перешла тот рубеж, который отмечает «громоздкость». Оптимальное соотношение потенциала и объёма система QNX переживала, наверное, в период QNX v.4 (~2000г.).

На квалифицированное изучение и освоение всех возможностей QNX квалифицированному Linux/POSIX разработчику, при условии постоянной активной работы с системой, нужно порядка 12-18 месяцев. По QNX достаточно мало печатных материалов для изучения её особенностей. По QNX есть коммерческие центры обучения (потому, собственно, и не ширится число публикаций по QNX), некоторые университеты (С.-Петербург, Брянск, Магнитогорск) включили её в свои учебные планы, но «территориальность» системы и наличие специалистов и концентрируется только вокруг названных выше центров.

Система MINIX, при том, что она предоставляет достаточно сходные возможности, намного компактнее. Квалифицированному Linux/POSIX разработчику, при тех же условиях, достаточно порядка 2-3 месяцев для включения в продуктивную работу. Система MINIX заметно лучше описана в публикациях, кроме того, весь её открытый исходный код добросовестно комментирован. По этой системе обучаются уже не один десяток лет. Именно поэтому столь разительно проще обучение MINIX.

Поддерживаемые аппаратные платформы

QNX поддерживает более полутора десятков аппаратных платформ (x86, PPC, ARM, MIPS и др.). MINIX гарантировано поддерживает только платформу x86, есть эксперименты по реализации на PPC, и попытки переноса на ARM. QNX содержит инструментарий для построения BSP (Board Support Package) - построения программного слоя поддержки (HAL) для самых разнообразных и новых чипсетов и подвидов поддерживаемых процессоров. Всё это делается для встраиваемости, до уровня «под крышкой чипа» - создания специализированных ИМС под задачу.

Не технические факторы сравнения

Производитель / разработчик

QNX — сугубо проприетарный проект от одного производителя: QSSL (QNX Software Systems Ltd. - Канада, Отава). До середины 90-х годов (QNX v.2 и v.4) этот проект производился в интересах, по заказу и при поддержке МО США, до этого времени на поставки QNX вне США распространялся запрет КОКОМ.

С начала 2000-х основным продуктом объявляется QNX v.6, названная ещё QNX Neutrino, которую мы и рассматриваем сравнительно с MINIX. Объявляется о том, что развитие предыдущей линии v.4 — прекращено, и что с 2004-2006г.г. будет прекращена и её поддержка. Судя по всему, 2003-2004г.г. являются «пиком пассионарности» в развитии QNX.

В конце октября 2004г. фирма QSSL вместе со своим детищем QNX была куплена крупным индийским концерном Harman, известным концерном по производству аудио-оборудования для автомобилей и вообще аудио-оборудования. Реально Harman по многим позициям не является даже производителем, а является финансовой «крышей» отдельных производителей. Вот здесь: <http://www.volvo tuning.ru/rubrics/information/mostnet.html>, в публикации того времени, были попытки объяснить мотивацию приобретений Harman различных компаний, но это мало убедительно. Покупка QSSL со стороны Harman была большой неожиданностью для всего мира промышленных систем.

Примерно в это время из QSSL уходят состоявшие в ней разработчики-идеологи QNX от начала разработки, в частности технический писатель Роб Кёртен, написавший большую часть фирменной техдокументации QNX, и несколько книг, собственно и принесших известность и популярность системе. Из старой команды, закладывавшей архитектуру QNX, остаётся только основоположник QSSL Дэн Додж. Неоднократно сообщалось (по комментариям в исходном коде, или из самих источников в QSSL), что для написания многих и значительных частей комплекта QNX последнего времени (исключая ключевые микроядерные и Photon компоненты) — привлекаются студенты канадских университетов; и уже не раз указывалось на грубейшие просчёты в коде таких частей, по характеру своему указывающих именно на недостаток практических навыков.

В сентябре 2007г. QSSL объявляет о поэтапном открытии исходного кода QNX, но это мера определённо вынужденная, в стремлении создать вокруг QNX широкое сообщество, и привлечь такое сообщество к разработкам, пусть не ключевых, компонентов программного набора (на манер сообществ Linux, *BSD, Open Solaris). Но это намерение не получило широкого (ожидаемо широкого) отклика: QSSL не воспользовалась и проигнорировала такое сообщество, самоорганизовавшееся к 2003-2004г.г., а к 2007г. время было упущено.

К показателям того же рода можно отнести и то, что с 2007-2008г.г. - возрождается развитие линии QNX v.4, о которой за несколько лет до того было официально объявлено о прекращении; это может свидетельствовать только о: а). желании финансово поэксплуатировать ещё то, что можно эксплуатировать и б). что новая линия QNX v.6 не восполнила все потребности и ожидания пользователей v.4.

Развитие и разработка MINIX исходит из университетских кругов Голландии (проф.Таненбаум и коллеги). Разработка поддерживается широким кругом общественности (в частности, из университетов Южной Америки). С 2004г., когда свободная лицензия была объявлена на компилятор АСК, можно считать, что разработка MINIX — свободная.

Дистрибуция

К однозначно негативным аспектам ориентации на QNX нужно отнести её систему дистрибуции. Ещё с 1991г. право **эксклюзивной** дистрибуции QNX на всех территориях xUSSR закреплено **юридически** (с QSSL) за одной, специально созданной под эти цели компанией: SWD Software Ltd., www.swd.ru, г.С.-Петербург. С течением времени права дистрибуции были распространены ещё на компанию АО "Системы реального времени — Украина" (RTSU), <http://rts.ua>, г.Днепропетровск (основано в 1997г.), но RTSU является просто зависимой (дочерней) структурой SWD.

SWD является достаточно энергичной и деятельной компанией (лоббирующей, в частности, применение QNX в ВПК через правительственные структуры), но ... монопольное положение обязывает — цены (см. далее) на применение QNX в России, к примеру, в 2 раза выше, чем цены дистрибуции в Германии, но воспользоваться поставщиками из Германии (и любого другого места) невозможно из-за эксклюзивности прав SWD.

Это ставит перспективы долгосрочного использования QNX в некоторое сомнительное положение, в зависимости от политики единственной компании SWD.

Относительно MINIX можно сказать, что под «дистрибуцией» здесь нужно понимать (ввиду свободы и открытости проекта) наличие разнообразных клонов, от разных разработчиков (по аналогии, как это имеет место с Linux), развивающихся, в первую очередь, каковую-то из сторон системы. В линии развития MINIX3 таких «дистрибуций» весьма значительное число:

- основной проект, концентрирующийся вокруг <http://www.minix3.org>;
- объявлено о развитии проекта «русский MINIX3», объединяющийся вокруг <http://minix3.ru/>, основной упор которого нацелен на вопросы руссификации системы в различных аспектах, и, главным образом, в создании объёмной и полноценной русскоязычной документации;
- открыто великое множество проектов в направлении «realtime MINIX3», вот только совсем некоторые из них: <http://realtimeminix.sourceforge.net/>, <http://www.rtminix3.org/>, <http://sites.google.com/site/minix4rt/>; наличие альтернативных реализаций, в условиях открытости кода и лицензии MINIX, позволяет заимствовать отдельные аспекты из различных реализаций.

Стоимость

Ценовая политика SWD/QSSL весьма сложная. Для коммерческих применений QNX должны приобретаться: а). лицензии рабочих мест разработчика и б). лицензия на хост исполнения (runtime). По состоянию на 2006/2007г.г. Эти стоимости составляли:

- лицензии на рабочее место разработчика - \$14.000 на каждое рабочее место;
- лицензия на хост исполнения - \$150-250 и более, на каждый хост, работающий под QNX; большой размах стоимостей определяется комплектацией: сеть QNET, графическая система Photon...

Получить детальнее (и более свежие) оценки достаточно сложно, так как компания SWD предпочитает формировать цены индивидуально, в ходе подготовки поставочных контрактов. Нужно отдать должное, что цены на runtime лицензии, как и заявлено, могут очень сильно (порядки) снижаться в зависимости от объёма закупок, но это реально сказывается только начиная с 1000 экземпляров в год.

Использование MINIX не требует прямых стоимостных отчислений. Но не следует забывать, что из-за ограниченности системных программных средств, долгосрочный проект под MINIX потребует содержать группу доработки системного программного обеспечения (по моим оценкам до 10 человек, этого достаточно, чтобы покрыть потребность в **доработках**). Расходы на поддержание таких доработок составляют дополнительные эксплуатационные стоимости выбора MINIX.

Лицензия

У QNX, как называет это фирма QSSL, «смешанная лицензия», но это не должно никого вводить в заблуждение: для

промышленных и коммерческих применений — это чисто проприетарная лицензия, требующая покупки лицензий на **каждое** рабочее место разработчика, и дополнительно runtime-лицензии на каждый экземпляр выпущенной продукции (о чём подробнее уже сказано выше).

Лицензии и открытость кода QNX — не предполагают внесения каких-либо изменений в код компонент, поставляемых в составе ОС (насколько мне известно, никому ещё не удалось собрать из якобы открытого кода QNX эквивалентную систему, некоторых фрагментов для сборки всё ещё недостаёт, сборка вообще не предусмотрена, а открытый код предоставляется более для изучения).

Применение: в лицензии QNX в явном виде присутствует ограничение на использование в проектах определённого рода, в частности, упоминается об проектах военного назначения. В таких случаях QSSL в одностороннем порядке оставляет за собой право отказать в предоставлении лицензий. Такого прецедента не было, но о существовании таких ограничений должно не забывать.

Открытый код MINIX распространяется под одной из самых либеральных из существующих лицензий: BSD, которая предусматривает: а). свободное использование кода в проектах любого рода и направленности; б). возможность внесения в этот код любых изменений под собственные потребности.

Внедрения

Внедрения QNX (в том числе и в России) куда многочисленнее чем MINIX3. Это обусловлено и нацеливанием MINIX3 на реализацию практических проектов только после 2005г.

Тем не менее, успешность внедрения проектов под QNX можно толковать и в смысле демонстрации успешности самой **идеологии микроядерных операционных систем**, вообще, безотносительно даже к конкретной системе — в конце концов, в QNX нет ничего такого особенно эксклюзивного, «серебряной пули», делающей её радикально отличающейся от других микроядерных систем.

Выводы

Аргументы в пользу использования MINIX

«Зона ответственности» обеих систем — построение прикладных систем экстремально высокой надёжности. В других областях и та и другая система значительно уступают уже существующим для этих задач платформам: Linux, Sun Solaris (Open Solaris), Windows. А в своём классе применений, от приложений под QNX и MINIX можно прогнозировать достижение примерно соизмеримого уровня надёжности, живучести и безотказности.

Применение MINIX должно быть признано целесообразным, в первую очередь, хотя бы из одного основного аргумента: «не складывать все яйца в одну корзину» (примером и напоминанием подобного провала должна быть ориентация **всего** МинЭлектронПррома СССР на линию микрокомпьютеров «Электроника», повторявших линию LSI/PDP-11 фирмы DEC).

Особенно настораживающим замыкание всех проектов на QNX покажется, если учитывать:

- а). проприетарность ОС QNX и замыкание его на единственного производителя, не очень крупного, и ощущающего реальный дефицит в разработческих кадрах;
- б). монополизации рынка QNX в xUSSR, жёсткая дистрибутивная политика фирмы SWD;
- в). спекулятивные и малопонятные тенденции развития QNX в условиях принадлежности её новому владельцу (фирма Harman).

QNX может оказаться абсолютно оптимальным выбором для единичного проекта, или краткосрочной (до 1-2 лет)

линейки сходных проектов, это обусловлено наличием чрезвычайно широкого спектра готового инструментария для разработки. На долгосрочных отраслевых программах (5 лет и более) риски плохо прогнозируемой динамики развития QNX могут сыграть с проектантами очень злую шутку.

Другим отрицательным фактором выбора QNX может стать затратная часть проекта. На единичных, даже весьма крупных, проектах дополнительные затраты не столь заметны (несколько десятков тыс. \$). Приемлемыми они могут быть и на высокотиражном производстве изделий со встраиваемым QNX (за счёт радикального снижения runtime лицензий), но это происходит только на тиражах от нескольких тысяч экземпляров в год. Если же планируется долгосрочная линейка типовых проектов (крупных, не изделий но систем) относительно невысокой плотности (десятки или сотни реализаций в год) дополнительная затратная часть лицензионных отчислений может стать непомерно большой.

Первостепенные задачи развития MINIX

1. Отработка на реальной системе MINIX3 приёмов практической работы и использования её возможностей, и по результатам написание руководств **по использованию**. Актуальность этой темы связана с тем, что разработчики MINIX3 проектировали её, главным образом, как учебную систему, написаны отличные руководства по структуре и строению системы, но практически отсутствуют руководства **для пользователей** по практическому использованию её возможностей (в своих целях и для своих проектов).

2. Подготовка тестов и проведение программы тестирования MINIX3 на предмет проявления известных, в общем, механизмов (таких например, как инверсия приоритетов [9]), снижающих детерминированность и предсказуемость поведения процессов в системе (свойства, приближающие систему к RTOS). В случае выявления присутствия отрицательных факторов, может быть проделана работа по модификации MINIX3, благо весь код открыт, и легко подлечит полной сборке из исходных кодов.

Источники информации

1. Олег Цилюрик, Дмитрий Алексеев, «Новое лицо операционной системы QNX»

<http://systemnews.com.ru/?mod=art&part=unix&id=013>

- общее состояние QNX v.6 на период её «пассионарного пика» (2003/2004г.г.).

2. «Реальное время или реальная Linux? Делаем трезвый выбор»

Пол Леру (Paul N. Leroux), технический аналитик компании QNX Software Systems, paul@qnx.com

Ромейн Саха (Romain Saha), менеджер сегмента сетевых систем компании QNX Software Systems, rsaha@qnx.com

<http://www.rts.ua/rus/articles/360/20/41/%20?PHPSESSID=54ff3609e9c233eb1339e0a9c8d6f3b1>

- обсуждение от аналитиков QSSL проблематики «реального времени».

3. SWD Software Ltd.

<http://www.swd.ru/index.php3?pid=10>

4. RTS Ukraine

<http://rts.ua/rus/about/?PHPSESSID=cde104d6543f4a0ed54256342151e276>

5. QNX Realtime Platform: Русский портал

<http://qnx.org.ru/forum/index.php>

- очень много детальных обсуждений и частных вопросов по QNX за период 2000-2009г.г.

6. Харьковский губернский портал:

<http://qnxclub.net/modules.php?name=Forums>

- то же, что и [5]... за период 2003-2008г.г.

7. MINIX3: Операционная система

<http://minix3.ru/>

8. Евгений Видревич «Real-Time OS – религиозная война компьютерной индустрии или заметки на полях одной статьи»

<http://qnxclub.net/files/articles/RemarksOnTheMargins/RemarksOnTheMargins.html>

- детально технически развенчиваются многие мифы о RTOS как некоторых совершенно особенных OS.

9. Олег Цилюрник, Егор Горошко «QNX/UNIX: анатомия параллелизма»

<http://www.books.ru/shop/books/357604>

<http://www.infanata.org/computers/prog/1146104000-qnxunix.-anatomija-parallelizma.html>

- в книге (288 стр.) детальнейшим образом разобраны все дисциплины планирования (поток) в QNX, что имеет прямое отношение к квалификации «реалтайм».