

# ЧТО ТАКОЕ QNX?

QNX - это семейство операционных систем реального времени и программных инструментальных средств, разработанных компанией [QNX Software Systems Limited](#).

В настоящее время для коммерческого использования доступно программное обеспечение QNX двух поколений:

- Платформа QNX 6
- Платформа QNX 4

## Платформа QNX 6

В основе QNX 6 лежат ОСРВ QNX Neutrino и комплект инструментов QNX Momentics, поставляемых в виде комплектов [QNX Software Development Platform \(QNX SDP\)](#).

[Скачать официальную брошюру о QNX Neutrino](#)

[Скачать официальную брошюру о QNX Momentics](#)

ОСРВ QNX Neutrino изначально предназначена для управления системами ответственного назначения, т.е. такими системами, сбой которых может привести к крупному материальному ущербу, ущербу для окружающей среды или ущербу для здоровья людей. Столь ответственная область применения требует не просто высокого качества программирования, но и принятия комплекса мер по обеспечению надёжности.

**Механизмы ОСРВ QNX Neutrino для обеспечения надежности:**

- Микроядерная архитектура с полной изоляцией модулей в ОЗУ
- Механизм обеспечения горячего резервирования и прозрачной модификации программных компонентов
- Механизм адаптивного квотирования ресурсов ЦПУ и ОЗУ (QNX Adaptive Partitioning)
- Технология поддержки многопроцессорности (QNX MultiCore Technology)
- Технология быстрой активации устройств (QNX Instant Device Activation)
- Механизм формирования распределенной вычислительной среды (QNX Transparent Distributed Networking)
- Технология автоматического восстановления процессов и логических соединений (QNX High Availability Toolkit)
- Механизмы мониторинга (QNX System Analysis Toolkit)

**Примечание:** Подробнее об основных функциональных возможностях QNX Neutrino можно прочитать в документе [Системная архитектура](#).

Для создания, тестирования, конфигурирования средств исполнения QNX Neutrino, а так же для разработки, отладки, тестирования и верификации приложений используются инструментальные средства QNX Momentics, устанавливаемые на выбор в QNX Neutrino, Microsoft Windows или Linux. Инструментальные средства QNX Momentics не требуются во время эксплуатации целевых систем QNX Neutrino.

На базе ОСРВ QNX Neutrino 6.5.0 компанией «СВД Встраиваемые Системы» разработана российская защищённая операционная система реального времени (ЗОСРВ) «Нейтрин» [КПДА.10964-01](#)

## Платформа QNX 4

QNX 4 - классическая программная платформа для систем автоматизированного управления технологическими процессами (АСУ ТП) промышленных предприятий.

**Основу платформы QNX 4 составляют:**

- ОСРВ QNX 4.25
- графическая оболочка Photon microGUI 1.14
- Средства поддержки стека протоколов TCP/IP
- Система программирования Watcom C/C++ 10.6

**Примечание:** Подробнее об основных функциональных возможностях QNX 4 можно прочитать в документе [Системная архитектура](#).

Поддержка и развитие QNX4 обусловлены длительными - иногда более 20 лет - жизненными циклами изделий, построенных на основе QNX.

На базе ОСРВ QNX 4.25, стека TCP/IP и интерфейса Photon компанией «СВД Встраиваемые Системы» разработана российская защищённая операционная система реального времени (ЗОСРВ) «QNX» [КПДА.00002-01](#)

Источник <<http://www.kpda.ru/products/qnx/>>

## ПРОДУКТЫ

Компания "СВД Встраиваемые Системы" является эксклюзивным поставщиком сертифицированной продукции собственной разработки и производства. Кроме того, наше предприятие является дилером продукции разработки и производства компании QNX Software Systems (QSS) для предприятий российского государственного сектора.

Для получения дополнительной информации о продукции и для заказа необходимо обращаться по телефону (812) 346-89-56 или по электронной [почте](#).



### ЗОСРВ «Нейтрино» КПДА.10964-01

Защищенная операционная система реального времени (ЗОСРВ) «Нейтрино» КПДА.10964-01, сертифицированная МО РФ и ФСТЭК России.

ЗОСРВ «Нейтрино» построена на базе операционной системы реального времени QNX Neutrino RTOS.

[Подробнее](#)



### ЗОСРВ «Нейтрино-Э» КПДА.10965-01

Защищенная операционная система реального времени (ЗОСРВ) «Нейтрино-Э» КПДА.10965-01, сертифицированная МО РФ.

ЗОСРВ «Нейтрино-Э» предназначена для работы на архитектуре Эльбрус.

[Подробнее](#)



## ЗОСРВ «QNX» КПДА.00002-01

Защищенная операционная система реального времени (ЗОСРВ) «QNX» КПДА.00002-01, сертифицирована ФСТЭК России и МО РФ.

Изделие КПДА.00002-01 построено на базе операционной системы реального времени QNX 4.25 RTOS

[Подробнее](#)



## Комплект разработчика для ЗОСРВ «Нейтрино»

Основными продуктами этой группы являются комплект разработчика "QNX Software Development Platform" (QNX SDP) и среды исполнения операционной системы реального времени "QNX Neutrino"

[Подробнее](#)



## Комплект разработчика для ЗОСРВ «Нейтрино-Э»

Продукт, обеспечивающий функционирование ЗОСРВ «Нейтрино-Э» на вычислительных комплексах на базе микропроцессоров "Эльбрус" в режиме реального времени.

[Подробнее](#)



## Программный комплекс антивирусной защиты – КПДА.94201-01

Антивирус обеспечивает сканирование системных и пользовательских файлов, а так же архивы различных форматов, на предмет обнаружения потенциальных угроз.

[Подробнее](#)



## Пакеты поддержки процессоров КОМДИВ для ЗОСРВ

### «Нейтрино»

Продукт, обеспечивающий функционирование ЗОСРВ «Нейтрино» на вычислительных комплексах на базе микропроцессоров КОМДИВ в режиме реального времени.

[Подробнее](#)



## Система диспетчерского контроля и управления (СДКУ)

### «Фокус»

Продукт предназначен для сбора, обработки, анализа, визуализации и хранения данных технологических процессов в режиме реального времени.

[Подробнее](#)



## Программная платформа бортовых информационных систем «ТРОЙКА»

Программная платформа бортовых информационных систем «ТРОЙКА» версии 2.0 - новое поколение передовой российской программной платформы для бортовых АСУ и САУ ответственного назначения.

[Подробнее](#)



## Решение для разработки графических интерфейсов Crank Storyboard Suite

Crank Storyboard Suite - это программный комплекс для эффективной разработки и отладки современных динамических графических пользовательских интерфейсов.

[Подробнее](#)

Источник <<http://www.kpda.ru/products/>>

## QNX NEUTRINO

Операционная система реального времени



Непревзойденная надежность, масштабируемость и производительность реального времени благодаря микроядерной архитектуре с защитой памяти.

**QNX**



В течение более 30 лет технология OCPB QNX помогала разработчикам создавать самые надежные в мире приложения реального времени – всевозможные системы, начиная от высокопроизводительного сетевого оборудования и автомобильных телематических систем до промышленных распределенных систем управления. Теперь самая надежная в мире ОС реального времени поддерживает еще и самый современный инструментарий.

## СОДЕРЖАНИЕ

<b>OCPB QNX Neutrino: мощь настоящей микроядерной технологии</b>	1
Производительность в реальном времени	
<b>Обмен сообщениями</b>	3
Автоматическая синхронизация	
<b>Соответствие стандарту POSIX</b>	4
Прозрачные распределенные вычисления	5
<b>Инструментальное ядро</b>	6
<b>Поддержка многоядерности</b>	7
<b>Графические решения</b>	8
Технология QNX Core Graphics	
Графическая оболочка QNX Photon microGUI	
Построитель приложений Photon Application Builder	
<b>Сетевые технологии</b>	10
Стеки протоколов	
<b>Платформа высокой готовности</b>	11
<b>Технология адаптивного квотирования</b>	12
<b>Файловые системы</b>	13
Встраиваемые файловые системы	
Дисковые файловые системы	
Файловые системы со сжатием	
Сетевые файловые системы	
Встраиваемая транзакционная файловая система	
<b>Среда управления ресурсами</b>	15
<b>Технология минидрайверов</b>	16

# ОСРВ QNX Neutrino: мощь настоящей микроядерной технологии

Начиная с 1980 г., разработчики полагались на технологию QNX при построении систем, требующих безотказного функционирования: медицинских приборов, телематических устройств, интернет-маршрутизаторов, call-центров службы 911, систем управления технологическими процессами и даже систем управления для международной космической станции (МКС). Вне зависимости от их масштаба и сложности эти системы объединяет одно: все они работают непрерывно по 24 часа в день, 365 дней в году, без перерывов.

Как QNX Neutrino делает это возможным? Ответ прост – эта ОС четко следует принципам микроядерной архитектуры. В QNX Neutrino ядро выполняет только несколько базовых функций (управление потоками, управление сигналами, обмен сообщениями, синхронизация, планирование, управление таймерами, управление процессами). Все остальные компоненты системы: драйверы, файловые системы, стеки протоколов, пользовательские приложения – выполняются вне пределов ядра как отдельные процессы, каждый в своем защищенном адресном пространстве. Такая схема изначально обладает исключительной, "встроенной", отказоустойчивостью.

Однако, и это еще не все. Все компоненты QNX Neutrino используют для общения друг с другом единый, четко детерминированный механизм – обмен сообщениями. Он образует между компонентами системы виртуальную "программную шину", позволяющую подключать к ней или, наоборот, отключать любой компонент "на лету". Мало того, сообщения могут свободно передаваться между узлами вычислительной сети, предоставляя прозрачный доступ к любому ресурсу, где бы он ни находился.

С QNX Neutrino вы можете:

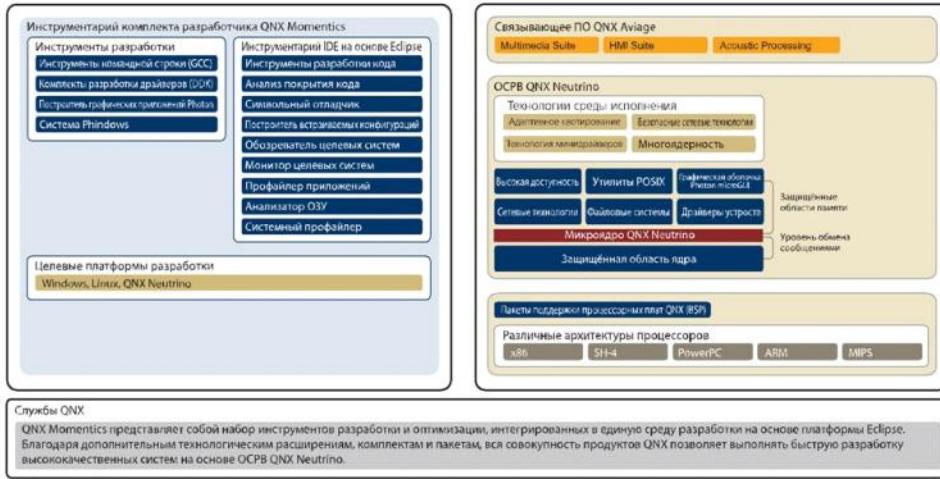
- **разрабатывать надежные, модернизируемые системы** – поскольку любой компонент в QNX Neutrino может быть добавлен или удален динамически, ваша система может продолжать работать

даже в процессе замены или добавления в нее новых приложений, драйверов или стеков протоколов.

■ **создавать системы, способные к самовосстановлению** – в QNX Neutrino любой компонент в случае отказа может быть перезапущен динамически, не нарушая работу микроядра и других компонентов. Например, если драйвер попытается обратиться к памяти за пределами своего адресного пространства (что для большинства ОС является фатальной ошибкой), QNX Neutrino корректно прекратит выполнение этого драйвера и освободит все занятые им ресурсы. Вы сможете даже автоматически перезапустить этот драйвер, используя администратор систем высокой готовности (CPM) ОСРВ QNX Neutrino.

■ **использовать одну и ту же ОС во всей своей линейке продуктов** – благодаря исключительной модульности QNX Neutrino, любые уже испытанные и проверенные компоненты – драйверы, приложения, дополнительные сервисы ОС – вы можете использовать повторно в других своих продуктах. Фактически, вы можете создать универсальный набор бинарных модулей, а затем применять его либо в однопроцессорном устройстве, либо в SMP-системе, либо в вычислительном кластере. Вне зависимости от масштаба и сложности вашей системы, вы будете использовать одну и ту же ОС, один и тот же программный интерфейс приложения (API) и один и тот же инструментарий разработчика.

## Продуктовая линейка QNX



Продуктовый набор QNX обеспечивает быструю разработку качественных систем на основе ОСРВ QNX Neutrino.

1 > QNX Software Systems

- Упростить проектирование отказоустойчивых кластеров - создавать высоконадежные, отказоустойчивые системы, использующие ресурсы нескольких процессоров посредством технологий прозрачных распределенных вычислений. Если один из процессоров в данный момент времени недоступен, запрос будет обработан подобным ресурсом другого процессора, обеспечивая тем самым встроенную отказоустойчивость и балансировку нагрузки.
- Создавать переносимые приложения - расширенная поддержка стандарта POSIX 1003.1-2001 позволяет портировать в QNX приложения из UNIX или Linux, а также использовать все многообразие программ с открытым исходным кодом.
- Оснастить Вашу систему развитым графическим интерфейсом - с помощью QNX Photon microGUI, встраиваемой оконной системы, поддерживающей передовые веб-технологии, многослойность, 3D-графику и мультимедийные средства.
- Использовать наиболее подходящее для Ваших задач оборудование - QNX Neutrino поддерживает среду исполнения и BSP-пакеты для множества чипсетов, включая MIPS, PowerPC, SH-4, ARM и x86.

## Производительность в реальном времени

Будучи основанной на микроядре и используя вытесняющую приоритетную многозадачность, QNX Neutrino предоставляет одновременно быстрое и предсказуемое время реакции на событие. Таким образом, потоки с высоким приоритетом всегда выполняют свои задачи вовремя, при любом уровне загрузки системы.

QNX Neutrino предоставляют высочайшую производительность реального времени, поскольку в ней реализованы:

- Сверхмалые задержки обработки прерывания и переключения контекста, позволяющие "выжать" максимум производительности из ваших аппаратных средств.
- Распределенный механизм наследования приоритетов и эмуляция протокола изменения приоритета для исключения инверсии приоритетов.
- Свобода выбора дисциплины планирования потоков (адаптивная, FIFO, карусельная, спорадическая) позволяет назначать каждому потоку свою дисциплину.
- Гарантирующая доступность процессора для задач с жестким графиком.
- Вложенные прерывания с фиксированной верхней границей времени реакции обеспечивают обработку наиболее приоритетных прерываний первыми и за предсказуемое время.

## Сравнительный анализ специализированных систем реального времени

### Результаты сравнения для OCPB QNX Neutrino

Архитектура ОСРВ	
Документирование функций ОС	
Конфигурирование ОС	
Интернет компоненты	
Инструменты разработки	
BSP-пакеты	
Техническая поддержка	

Источник данных: Dedicated Systems Expert, 8 сентября 2011 года.

### Главные особенности

- Прекрасная архитектура для создания распределенных и отказоустойчивых систем
- Очень высокая производительность и предсказуемое поведение
- Эффективная и дружелюбная среда разработки
- Высокое качество технической документации

В ходе независимого анализа шести ведущих встраиваемых систем журнал Dedicated Systems Experts присудил OCPB QNX Neutrino самые высокие баллы по сравнению с другими операционными системами.

## Обмен сообщениями

В QNX Neutrino обмен сообщениями формирует виртуальную «программную» шину, которая позволяет производить модификации «на лету» - любой компонент в QNX Neutrino может быть добавлен или удален динамически. Использование этой технологии позволяет оставить в ядре операционной системы только основные сервисы, такие как сигналы, таймеры и планировщик.

Все другие компоненты – файловые системы, драйверы, стеки протоколов, приложения – выполняются в безопасном пространстве с защитой памяти. Реализуется встроенная устойчивость к сбоям. Кроме того, происходит прозрачное распределение сообщений между процессами, обеспечивая непрерывный доступ к ресурсам в любой части сети.

Технология обмена сообщениями на основе микроядра позволяет:

- автоматически синхронизировать выполнение взаимодействующих компонентов;
- избавить вас от необходимости следить за очередностью доставки данных;
- разбить сложное приложение на четко разграниченные функциональные блоки, которые можно разрабатывать и тестировать по отдельности;
- придавать системам изящность, делающую их легкими в эксплуатации и обслуживании;
- действовать по всей сети, предоставляя вашим приложениям прозрачный доступ к сервисам и ресурсам удаленных узлов.

### Автоматическая синхронизация

Обмен сообщениями в QNX также чрезвычайно эффективен, поскольку каждая операция происходит непосредственно между отправителем и получателем. Соответственно, не происходит никакого промежуточного копирования данных и не требуется никаких дополнительных механизмов межзадачного взаимодействия для синхронизации.

Кроме того, нет необходимости реализовывать дополнительные уровни для обработки сложных сообщений. Чтобы обмениваться сообщениями с системными сервисами, приложения могут использовать стандартные вызовы POSIX.

Как это работает? В QNX Neutrino каждая программа, предоставляющая некий сервис (например, драйвер), может зарегистрировать в пространстве имен путем «файл» или «каталог». В дальнейшем любое приложение может соединиться с этим драйвером, осуществив по отношению к данному файлу или каталогу стандартную операцию

`open()`. Результатом будет получение приложением обычного файлового дескриптора, через который приложение сможет обращаться к сервисам драйвера при помощи вызовов POSIX, предназначенных для работы с файловыми дескрипторами – `read()`, `write()`, `Iseek()` и т.п. Библиотека языка Си автоматически преобразует эти вызовы в соответствующие сообщения и передает их драйверу. Например, когда приложение вызывает стандартную функцию `read()`, чтобы считать готовые данные, библиотека преобразует этот вызов в сообщение «запрос на чтение». На самом деле в QNX Neutrino приложения используют преимущества обмена сообщениями каждый раз, когда работают с файловыми дескрипторами или указателями на файлы.

Такой подход позволяет вам:

- **Упростить обслуживание систем** – поскольку пространство имен путей четко отделяет сервисы от клиентских приложений, обновление для ваших систем становится элементарной задачей. Любой сервис можно заменить на его новую версию, в том числе в процессе эксплуатации – и клиенты сами автоматически найдут его.
- **Расширять ОС для нестандартных задач** – QNX Neutrino предоставляет разработчикам библиотеку администратора ресурсов, которая позволяет сервисным программам регистрировать свои имена в пространстве имен путей и обрабатывать запросы от клиентских приложений.

В дополнение ко всему, в QNX Neutrino любые системные сервисы, включая драйверы, являются программами пользователяского кольца, а значит, разрабатываются точно так же, как и любые другие приложения. В результате вы получаете возможность легко расширять ОС совершенно новыми, специфичными для ваших приложений возможностями.

Обратите внимание, что обмен сообщениями можно использовать напрямую, при помощи функций `MsgSend()`, `MsgReceive()` и `MsgReply()`. В QNX Neutrino доступны также и традиционные формы межзадачного взаимодействия – программные каналы, FIFO, очереди сообщений POSIX, разделяемая память и сигналы.

## Соответствие стандарту POSIX

Программные интерфейсы приложений собственной разработки, распространенные в традиционных моделях операционных систем, ограничивают переносимость программного кода и могут значительно снизить экономические показатели эффективности. Перенос кода в поддерживающую стандарт POSIX операционную систему реального времени, такую как OCPB QNX Neutrino, позволяет компилировать программные приложения, уже созданные ранее для других систем, сокращая тем самым возможные затраты на их приобретение, увеличивая производительность и ускоряя процесс разработки новых продуктов.

Многие операционные системы реального времени, претендующие на POSIX-совместимость, часто на самом деле поддерживают только его небольшие подмножества. QNX Neutrino, в отличие от них, была изначально задумана как POSIX-совместимая ОСРВ - в ней реализована "встроенная" поддержка POSIX. Этот подход исключает необходимость в сложных промежуточных уровнях, позволяя вам достигнуть лучшей производительности и сэкономить на объеме памяти.

QNX Neutrino соответствует стандарту POSIX 1003.13-2003 (POSIX.1), включая многопоточность, расширения реального времени и ряд других опций (см. ниже). В результате, в QNX Neutrino очень просто переносить программы с открытым исходным текстом из, например, UNIX или Linux - в большинстве случаев перенос сводится к перекомпиляции и компоновке с библиотеками QNX Neutrino. Более того, опыт показывает, что программисты с опытом работы в UNIX или Linux осваиваются в QNX Neutrino практически мгновенно.

QNX Neutrino обеспечивает:

- Простоту переноса и функциональную совместимость приложений между системами, поддерживающими стандарт POSIX, включая Linux и Unix - в большинстве случаев перенос сводится к перекомпиляции и компоновке с библиотеками QNX Neutrino.
- Чистоту реализации стека протоколов IP, который получает гибкость прикладной среды открытого стандарта POSIX, снижая риск нарушения авторских прав.
- Знакомую среду разработки, позволяющую программистам с опытом работы в UNIX или Linux освоиться в QNX Neutrino практически мгновенно.

## Поддержка широкого спектра процессоров

Используя OCPB QNX Neutrino, Вы можете выбирать системы на основе различных процессоров: ARM, MIPS, PowerPC, SH-4.

На самом деле, OCPB QNX Neutrino предоставляет функции и макросы, которые позволяют Вам создавать приложения и драйверы, не зависящие от процессоров. В некоторых случаях, Вам просто надо указать новый целевой процессор чтобы сгенерировать двоичный код для него - Вы можете даже создавать программный код одновременно для целого ряда процессоров. В результате, Вы сможете отлаживать и тестировать приложения до того, как будет выбран целевой процессор, или указать другой процессор, чтобы получить выигрыш в стоимости системы или соотношении цена/производительность для неё.

Специалисты QNX Software Systems постоянно обновляют список поддерживаемых процессоров и процессорных плат. Посмотреть самую последнюю версию списка можно на сайте [www.qnx.com](http://www.qnx.com).

## Прозрачные распределенные вычисления

Система прозрачных распределенных вычислений OCPB QNX Neutrino дополняет стандартную инфраструктуру обмена сообщениями, необходимую для передачи информации между процессорами - сохраняя время и сокращая затраты, связанные со специализированными разработками и совершенствованием аппаратных средств.

Прозрачные распределенные вычисления обеспечивают структуру для динамического соединения ресурсов оборудования и программного обеспечения, расположенных на удаленных узлах, посредством стандартных сообщений.

Процессы, запущенные на одном процессоре, будут продолжать обмениваться данными друг с другом, даже если будут последовательно распределены между несколькими процессорами, обеспечивая возможность расширять сервисы и упростить проектирование многоузловых систем.

Используя эту уникальную возможность, можно создавать высоконадежные, отказоустойчивые системы, предоставляющие по запросу доступ к ресурсам нескольких процессоров.

Используя прозрачные распределенные вычисления QNX Neutrino, вы можете:

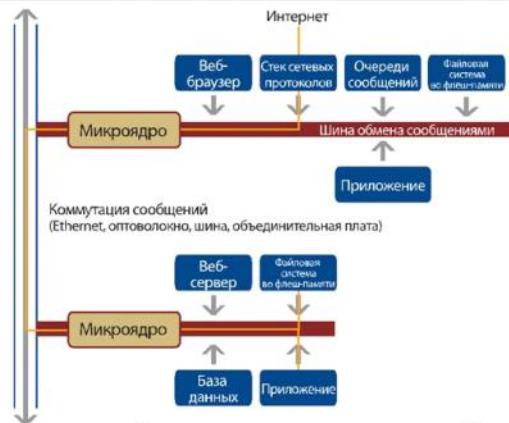
- **Сократить затраты на оборудование** - при использовании распределенных вычислений узлы сети могут совместно использовать ресурсы вместо их дублирования. Например, если на одном узле расположена большая файловая система во флеш-памяти, другим узлам иметь такую же не обязательно - они смогут использовать файловую систему того узла, на котором она уже есть. Аналогично, если на одном узле запущен стек TCP/IP, то все остальные узлы смогут использовать этот узел как TCP/IP-шлюз, исключая необходимость в настройке нескольких IP-адресов.
- **Обеспечить встроенную отказоустойчивость и балансировку нагрузки** - поскольку обмен сообщениями в QNX Neutrino предоставляет прозрачный доступ к сервисам вне зависимости от их местоположения, приложения могут полностью абстрагироваться от принятия решений о том, кто будет

обрабатывать запрос от клиента, где этот сервис расположен, и есть ли другие сервисы, способные обработать этот запрос (например, в случае дублирования сервиса на нескольких узлах для обеспечения отказоустойчивости или балансировки нагрузки).

**Добавлять вычислительную мощность без дополнительных разработок** - чтобы добавить в распоряжение вашего приложения больше вычислительных мощностей или больше физических интерфейсов, достаточно просто вставить дополнительную процессорную карту или добавить в сеть еще один компьютер. Приложения на уже имеющихся узлах смогут пользоваться ресурсами нового узла без внесения в них каких-либо изменений.

**Использовать любые средства сообщения** - поскольку механизм распределенных вычислений QNX Neutrino функционирует над транспортным уровнем, он одинаково хорошо работает через локальные сети, объединительные панели, собственные коммутируемые сети (switch fabrics) и шинные интерфейсы типа CAN и MOST.

### Прозрачные распределенные вычисления



Технология прозрачных распределенных вычислений позволяет приложению получить доступ к любому узлу в сети. Приложения и сервисы могут мгновенно стать распределенными по сети без разработки специального кода.

## Инструментальное ядро

Используя инструментальное ядро в ОСРВ QNX Neutrino, вы можете быстро разрешать конфликты синхронизации, обнаруживать взаимные блокировки потоков, прослеживать корни семантических ошибок, выявлять скрытые дефекты программного и аппаратного обеспечения - причем как в однопроцессорных, так и в многопроцессорных системах.

Инструментальное ядро может отслеживать и протоколировать каждое системное событие, включая вызовы ядра, прерывания, сообщения, смену состояний потоков и действия планировщика. Фактически, инструментальное ядро представляет собой логический анализатор для всей вашей системы: если что-то сработает не так, ядро поможет вам выяснить, когда произошло событие, какие программные модули были в нем задействованы, что они делали, и, самое главное, как интерпретировать это событие.

Инструментальное ядро по функциональности абсолютно аналогично стандартному ядру QNX Neutrino, так что вам не придется изменять свой код. Более того, его можно использовать в системах, находящихся в эксплуатации, поскольку оно лишь немногим медленнее обычного микроядра. Влияние на производительность системы проявляется только в процессе трассировки событий, но и здесь вы можете свести это влияние к минимуму, производя динамическое протоколирование и фильтруя информацию по уровням важности.

Инструментальное ядро позволяет вам:

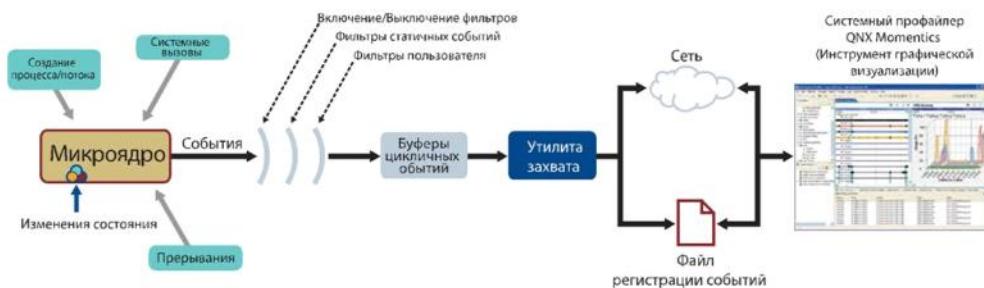
- **Получать информацию о каждом системном событии** - Инструментальное ядро может отслеживать и протоколировать каждое системное событие, включая вызовы

ядра, прерывания, сообщения, смену состояний потоков и планирование.

■ **Визуализировать системные события для быстрой локализации проблем** - Обладая способностью генерировать огромные объемы системной информации, инструментальное ядро одновременно позволяет вам четко контролировать, какие события и когда записывать, таким образом, вы можете уделять внимание в точности нужным областям. А чтобы сделать задачу интерпретации данных еще проще, ядро тесно интегрировано с системным профайлером QNX Momentics, позволяя визуально анализировать нужные участки и прослеживать сложные взаимодействия.

■ **Создавать свои собственные фильтры событий** - Если готовые фильтры событий, предоставляемые инструментальным ядром, не соответствуют ситуациям, которые вы хотели бы изучить, это не проблема: ядро поддерживает динамические определяемые пользователем фильтры для отслеживания сложных условий, характерных для конкретного приложения. Ваше приложение также может само вставлять специальные события в систему трассировки для упреждающего влияния на процесс протоколирования.

### Инструментальное ядро



Используя инструментальное ядро вместе с системным профайлером QNX Momentics, вы можете обнаруживать взаимные блокировки потоков, прослеживать корни семантических ошибок, а также целый ряд причин, снижающих производительность системы.

## Поддержка многоядерности

QNX предлагает наиболее функциональную программную платформу для многоядерных систем. Это полностью интегрированное решение поддерживает асимметричную (AMP), симметричную (SMP) и исключительную (BMP) модели многопроцессорности. Впервые представленная компанией QNX Software Systems, исключительная многопроцессорность является революционной технологией, которая упрощает перенос программного кода и способствует созданию перспективных программных продуктов.

Многие годы QNX доказывала мощь симметричной многопроцессорности (SMP) посредством ведущей в отрасли ОСРВ QNX Neutrino. При разработке новых приложений симметричная многопроцессорность (SMP) предлагает способ извлечь максимум производительности из многоядерных процессоров. Однако, не все унаследованные приложения совместимы с моделью симметричной многопроцессорности. Чтобы исправить это, QNX Software Systems создала модель исключительной многопроцессорности (BMP). Исключительная многопроцессорность обеспечивает возможность назначать приложениям или процессам собственное ядро. Это позволяет запускать программы, написанные для однопроцессорной среды, на многоядерных процессорах без дополнительной модификации кода. Унаследованные приложения, таким образом, могут сосуществовать с приложениями, разработанными специально для многоядерных процессоров. Унаследованные приложения могут быть назначены конкретным процессорам, в то же время приложения, разработанные для параллельной обработки данных, могут управляться операционной системой с целью повышения производительности.

Комплект разработчика QNX Momentics® обеспечивает разработчиков инструментарием, необходимым для того, чтобы создавать, отлаживать и оптимизировать многоядерные системы. Инструментарий включает в себя:

- Полную инструментальную линейку с поддержкой многопроцессорности для многопроцессорных отладчиков, компиляторов и встраиваемых инструментов.
- Профайлер приложений QNX Momentics, позволяющий производить анализ программного кода и находить слабые места, влияющие на производительность, что помогает получить максимум выгоды из многоядерной параллельной обработки данных.
- Системный профайлер QNX Momentics для увеличения производительности в многоядерных системах путем визуализации загрузки многоядерного процессора, передачи данных и переключения потоков между ядрами.

### Возможности

	SMP	BMP	AMP
Прозрачное разделяемое использование ресурсов	✓	✓	—
Масштабирование на платформы с более чем двумя ядрами	✓	✓	ограничения
Применение разных операционных сред	—	—	✓
Привязка задач к назначенным процессорам	—	✓	✓
Обмен сообщениями между процессорными ядрами	Быстрый (на уровне ядра ОС)	Быстрый (на уровне ядра ОС)	Менее быстрый (на уровне приложений)
Синхронизация потоков, выполняемых на разных ядрах	✓	✓	—
Динамическое выравнивание загрузки	✓	✓	—
Комплексные инструменты системной отладки и оптимизации	✓	✓	—

Только QNX Software Systems предоставляет возможность выбора подходящей многопроцессорной модели для Вашей многоядерной архитектуры. Используйте многоядерную технологию ОСРВ QNX Neutrino, чтобы достичь идеального баланса между производительностью, масштабированием и легкостью переноса программного кода.

## Графические решения

Компания QNX Software Systems предоставляет комплекс технологий для быстрой разработки и реализации графических интерфейсов встраиваемых систем. Мощная встраиваемая оконная система QNX Photon MicroGUI, доступные инструменты разработки, поддержка технологий многослойности, стандарта OpenGL ES и множества мультимедийных форматов позволяют создавать на базе QNX Neutrino сложные, быстродействующие и надежные графические среды для встраиваемых систем и обеспечивать полноценный графический интерфейс даже самым компактным встраиваемым устройствам.

### Технология QNX Core Graphics

Технология QNX Core Graphics предназначена для построения сложных графических интерфейсов, в которых требуется быстрая прорисовка двух- и трехмерных изображений.

Суть технологии CoreGraphics заключается в том, чтобы предоставить программам доступ непосредственно к видеодрайверам без передачи сообщений и переключения контекста при рисовании. Это существенно повышает скорость графических приложений при компактности, необходимой для встроенных систем. Физически QNX CoreGraphics представляет собой набор компонентов разработки (библиотеки и заголовочные файлы) и компонентов среды исполнения (библиотеки времени исполнения, OpenKODE-совместимый композитный менеджер и др.).

Платформа QNX Graphics Framework позволяет создавать приложения различной сложности, от простейших интерфейсов миниатюрных встраиваемых систем до сложных оконных интерфейсов. При этом не требуется графическая оболочка (приложения получают непосредственный доступ к видеодрайверам), что обеспечивает быстродействие приложений графического интерфейса.

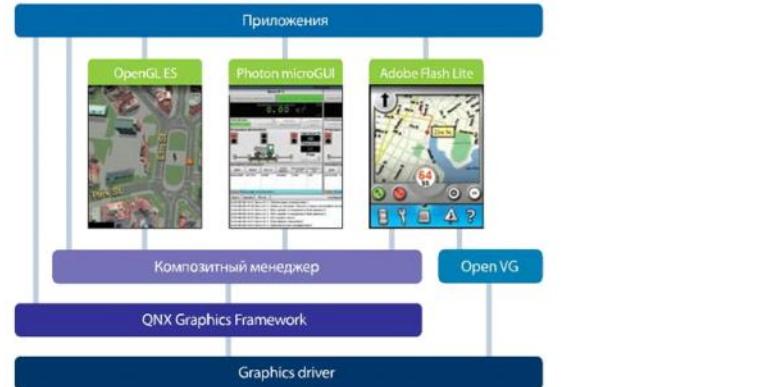
### OpenGL ES

Основу QNX Core Graphics составляет сертифицированная реализация OpenGL ES API. Реализация QNX Neutrino поддерживает профили Common и EGL. Благодаря этому, вы можете использовать в своих проектах существующий код OpenGL, продукты третьих сторон, а также опираться на обширный опыт применения данного стандарта в различных отраслях.

### Композитный менеджер QNX

Композитный менеджер QNX (QNX Neutrino Composition Manager) образует унифицированную среду для управления окнами и создания сложных изображений на экране. Он позволяет совмещать в одном окне различные средства визуализации (Flash-анимацию, технологии мультимедиа, веб-браузер, двух- и трехмерную графику и т.д.), что позволяет создавать более привлекательные интерфейсы. При этом обеспечивается высокая эффективность использования аппаратных ресурсов и простота модернизации оконной системы. В зависимости от особенностей каждой конкретной системы, используя менеджер композиции QNX, разработчик может перераспределять задачи рендеринга между видеоконтроллерами и аппаратно-независимым уровнем абстракции.

Композитный менеджер QNX базируется на открытом стандартном OpenKODE API, что позволяет наименьшими усилиями портировать готовые приложения и графические элементы и модернизировать всю систему (для изменения расположения и размеров окон не надо вручную переписывать исходные коды).



### Графическая оболочка QNX Photon microGUI

Модульная графическая оболочка QNX Photon microGUI способна обеспечить современный графический интерфейс даже самому маленькому встраиваемому устройству. В QNX Photon microGUI представлен большой набор повторно используемых экранных компонентов (виджетов), множество шрифтов, интегрированная поддержка многодисплейных систем и широкая многоязыковая поддержка для локализации продуктов.

Окнная система QNX Photon microGUI также поддерживает графические средства разработки, такие как построитель приложений Photon Application Builder, инструменты визуального макетирования и проектирования (для быстрой разработки прототипов приложений) и средства удаленного доступа к приложениям на основе Photon с удаленной машины под управлением Windows.

#### ■ строить сложные многослойные изображения

Photon позволяет создавать плавно работающие многослойные дисплеи, способные отображать комбинацию из графики и видео в реальном масштабе времени – это прекрасное решение для разнообразных встраиваемых, промышленных, транспортных и медицинских систем.

#### ■ создавать уникальный интерфейс

Используя механизм, называемый стилями виджетов (widget styles), вы можете настраивать внешний вид кнопок, меню, окон и других элементов интерфейса – как индивидуально, так и на глобальном уровне.

#### ■ отображать и вводить текст на нескольких языках одновременно

Photon поддерживает стандарт Unicode, благодаря чему пользователи могут, например, вводить русские, китайские или корейские символы, в то время как система отображает текст на английском или любом другом языке.

#### ■ отображать высококачественные шрифты на дисплеях любого размера

Администратор шрифтов Photon поддерживает множество шрифтовых форматов, включая растровые и TrueType. Вы можете масштабировать шрифты до любого размера и отображать их с использованием технологии сглаживания, добиваясь четко читаемых изображений на жидкокристаллических индикаторах и других устройствах с низким разрешением.

#### ■ подключать компоненты новых форматов

Используя систему плагинов и высокоразвитый API, вы легко сможете добавить в систему ваши собственные мультимедийные компоненты.

#### ■ обновлять графический интерфейс «на лету»

По аналогии с самой QNX Neutrino, Photon основан на компактном микроядре и предоставляет большинство своих сервисов (видеодрайверов, менеджеров окон, драйверов устройств ввода и т.п.) посредством опциональных процессов, работающих в защищенных областях памяти. Таким образом, вы можете, не перезагружая и не останавливая работу системы, заменять или обновлять практически любой компонент вашего интерфейса.

#### ■ одновременно отображать встроенные графические приложения и Java-приложения

– каждое приложение запускается в своем собственном окне.

### Построитель приложений Photon Application Builder

Используя Photon Application Builder (PhAB), визуальное средство разработки приложений QNX Photon microGUI, вы можете создавать полнофункциональные пользовательские интерфейсы с простотой щелчка мыши. Фактически, PhAB может автоматически генерировать полностью работающий прототип графического интерфейса вашего приложения, и вам для этого не придется писать никакого кода.

#### Используя Photon Application Builder, вы можете:

- начать разработку, взяв за основу готовые шаблоны PhAB, либо строить интерфейс из обширной палитры доступных элементов управления (виджетов);
- привязывать диалоговые окна непосредственно к виджетам или добавлять вызовы функций (предопределенные, по горячей клавише или по низкоуровневым событиям), наполняя ваше приложение функциональностью;
- однажды создав виджет, потом динамически генерировать столько его копий, сколько нужно вашему приложению, при этом объем кода, необходимый для создания виджета, сводится всего к одной строке;
- редактировать, компилировать, тестировать и отлаживать ваше графическое приложение, используя весь спектр инструментов, предоставляемых QNX Momentics.

## Сетевые технологии

Компания QNX Software Systems обеспечивает в своих продуктах поддержку широкого набора сетевых протоколов на основе индустриальных стандартов. Стеки протоколов прошли полное тестирование и интегрированы в OCPB QNX Neutrino. Это позволяет значительно сократить время разработки и ускорить создание конечного продукта.

В OCPB QNX Neutrino сетевые протоколы выполняются вне ядра в виде отдельных процессов с защитой памяти. В результате одновременно можно использовать любое количество поддерживаемых сетевых протоколов, в том числе TCP/IP и технологии прозрачных распределенных вычислений. Также предусмотрена возможность одновременного выполнения множества приложений (например, виртуальных маршрутизаторов) с использованием одного стека протоколов и на одной физической платформе, что повышает степень защиты и управляемости отдельных приложений.

Используя сетевые технологии OCPB QNX Neutrino, Вы сможете:

- Сократить время вывода продукта на рынок благодаря готовым программным решениям. Полная интеграция сетевых протоколов в OCPB QNX Neutrino и дополнительные модули исходного кода упрощают интеграцию и оптимизацию оборудования.
- Выбрать стеки протоколов IPv4 и IPv6.
- Получить надежные решения на основе индустриальных стандартов; Реализация протоколов базируется на стандартном программном интерфейсе BSD и поддерживает все существующие индустриальные стандарты. Также выполнено полное тестирование на функциональную совместимость.
- Упростить переносимость приложений и использовать богатую базу сетевого кода "третьих" производителей, основанного на стандарте POSIX и BSD API.
- Запускать, останавливать и обновлять любой сетевой драйвер "на лету".
- Сочетать любое количество сетевых протоколов, включая TCP/IP и сеть прозрачных распределенных вычислений QNet.
- Создавать многочисленные виртуальные сети (VLAN), запуская несколько копий стека TCP/IP на одном и том же физическом интерфейсе.

### Стеки протоколов

QNX Neutrino предлагает использовать следующие протоколы и стеки:

**■ Стек NetBSD TCP/IP** - поддерживает IP-маршрутизацию, сокеты управления маршрутизацией, широковещательные и многоабонентские передачи, ARP, ICMP и IGMP, а также CIFS, DHCP, DNS, NFS, PPP, PPPoE, UDP и встраиваемый веб-сервер. При разработке приложений для этого стека, программисты используют сокет программного интерфейса приложений индустриального стандарта BSD.

**■ Усовершенствованный стек NetBSD TCP/IP с поддержкой IPSec и IPv6** - включает в себя все возможности стандартного стека, а также функции, предназначенные для мобильных и безопасных коммуникаций нового поколения. Предоставляет полную поддержку IPv6 и IPSec за счет расширений KAME, а также поддержку VPN через тоннельные каналы IPSec. Также содержит в себе оптимизированный код для переадресации для дополнительной производительности.

**■ Встраиваемый стек TCP/IP для систем с ограниченным объемом памяти** - несмотря на свой небольшой объем (около 500 КБ), этот стек предоставляет поддержку IP, TCP и UDP через интерфейсы Ethernet, PPP и PPPoE. Поскольку в приложениях используется интерфейс BSD-сокетов, для переключения между встраиваемым стеком и стеком NetBSD не нужно перекомпилировать код.

**■ IP-фильтрация и трансляция сетевых адресов (NAT)** - Если вам необходимо разработать высоконадежную службу межсетевой защиты (firewall), корпоративный или домашний сетевой шлюз, вы можете применить готовое программное решение, которое обеспечивает защиту информационных систем от несанкционированного доступа извне. Технология IP-фильтрации и трансляции сетевых адресов (Network Address Translation, NAT) в OCPB QNX® Neutrino® предоставляет собой эффективный механизм для проверки пакетов с высокой производительностью. IP-фильтр имеет поддержку протокола NAT и прозрачным образом работает со следующими стеками TCP/IP: Core Networking (IPv4) и Enhanced Dual Networking (Ipv4, IPv6, IPSec).

## Платформа высокой готовности

ОСРВ QNX Neutrino позволяет достичь коэффициента готовности, превышающего значение 99,999%, и обеспечивает комплексный подход к выявлению отказов и восстановлению системы.

Модульная микроадерная архитектура ОСРВ QNX Neutrino позволяет изолировать сбои на уровне драйверов. Совместно с менеджером высокой готовности реализована технология «умный сторож», которая помогает системе автоматически восстанавливаться после сбоев. Такой подход позволяет создавать по-настоящему самовосстанавливающиеся системы.

Что касается достижения высокой готовности (ВГ), ОСРВ QNX Neutrino воплощает простой принцип: быстрое перезапускание один компонент, чем всю систему. Например, если возникает проблема с драйвером или стеком протоколов ОСРВ QNX Neutrino, он может быть немедленно отключен и восстановлен с помощью менеджера высокой готовности, и это занимает всего несколько секунд, соответственно, нет необходимости перезагружать всю систему. Этот выверенный подход к изоляции сбоев и восстановлению четко отработан, поэтому, по сравнению с другими операционными системами, с помощью ОСРВ QNX Neutrino можно достичь минимальных значений среднего времени восстановления системы (MTTR) после сбоев.

Технология менеджера высокой готовности ОСРВ QNX Neutrino обеспечивает:

- **Мгновенные уведомления об отказах** - в менеджере высокой готовности реализован механизм квитаций работоспособности, следящий за состоянием каждого компонента системы и позволяющий обнаруживать отказы на самой ранней стадии. Если менеджер высокой готовности обнаруживает определенное стечие обстоятельств или отказ, он может автоматически мгновенно оповестить об этом другие компоненты.

- **Настраиваемые сценарии восстановления** - используя библиотеку менеджера высокой готовности, ваше приложение может явно указать менеджеру высокой готовности, какие действия по восстановлению и в каком порядке следует предпринять в случае сбоя.

- **Автоматическое восстановление соединений** - менеджер высокой готовности также предоставляет клиентскую библиотеку, которая позволяет вашей системе в случае отказа моментально восстановить прерванные соединения.

- **«Посмертный» анализ** - если процесс завершается некорректно, менеджер высокой готовности может сохранить его образ для последующей обработки. Анализируя этот образ, вы сможете моментально определить, какая строка кода вызвала сбой, а также узнать содержимое переменных, чтобы точно определить, что именно произошло.

- **Устойчивость к внутренним сбоям** - менеджер высокой готовности чрезвычайно устойчив к внутренним сбоям, так как имеет сервис собственного мониторинга, копию которого записывает параллельный «процесс-хранитель». Если он по какой-либо причине завершается некорректно, то моментально и полностью восстанавливает свое предыдущее состояние.

### Архитектура менеджера высокой готовности



Когда менеджер высокой готовности обнаруживает сбой компонента, он извещает систему и запускает алгоритм восстановления. Совместно с собственным «Дублером», менеджер высокой готовности представляет собой основу для самовосстанавливающейся системы.

## Технология адаптивного квотирования

ОСРВ QNX Neutrino - это единственная встраиваемая операционная система, обеспечивающая защищенный режим гарантированного реального времени без компромиссов для производительности и гибкости. Используя запатентованную технологию адаптивной декомпозиции, вы сможете обеспечить вашим приложениям гарантированное реальное время, нейтрализовать угрозы и защитить вашу систему.

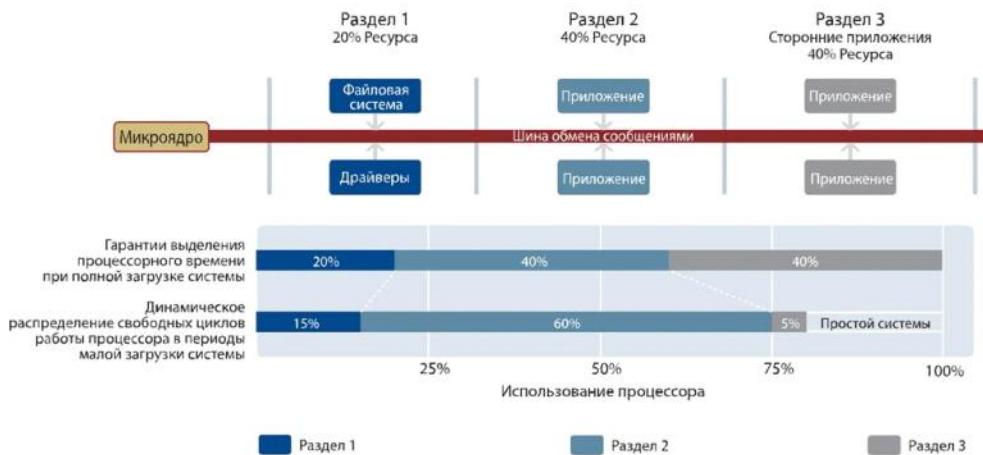
Используя технологию адаптивного квотирования, Вы можете:

- **Создавать защищенные системы** - вредоносное программное обеспечение может помешать нормальному работе важных системных функций, ограничивая время их доступа к процессорной обработке. Чтобы избежать этого, технология адаптивного квотирования QNX позволяет вам создавать раздел между ключевыми процессами вашей системы и сторонними программным обеспечением, таким образом, защищая систему.
- **Повысить эффективность использования процессора** - в отличие от статичных подходов к квотированию с помощью циклического планировщика, технология адаптивного квотирования распознает циклы загрузки процессора и периоды, когда система работает вхолостую. С технологией адаптивного квотирования исключается неэффективное расходование времени при простое системы. Так, неиспользуемые циклы работы процессора с одного раздела могут быть динамически перераспределены на

другие, следовательно, общее время эффективной работы процессора увеличивается. Стандартный планировщик, работающий на основании приоритетов работает лишь тогда, когда система не полностью загружена и не подвергается атаке. Технология адаптивного квотирования обеспечивает простое, надежное решение для систем, выполняющих ограниченный круг задач с интенсивной загрузкой процессора.

- **Быстро начать работу** - Технология адаптивного квотирования использует стандартную модель программирования POSIX, у Вас будет возможность использовать знакомую среду разработки, привычную для любой встраиваемой системы технику программирования и отладки. Вы можете применить технологию адаптивного квотирования, просто определив размер раздела и решив, какое приложение или процесс будет использовать определенный раздел. С технологией адаптивного квотирования QNX приложения и системные службы будут запускаться соответственно в своих блоках.

### Создайте защищенные разделы для Вашего программного обеспечения, используя технологию адаптивного квотирования



Запатентованная технология адаптивного квотирования от компании QNX Software Systems позволяет выделять приложениям и процессам гарантированный ресурс процессорного времени при полной загрузке системы, а также динамически распределяет свободные циклы работы процессора в периоды его малой загрузки.

## Файловые системы

В традиционных ОС файловые системы встроены в ядро. В QNX Neutrino файловые системы расположены вне пределов ядра и выполняются в отдельных защищенных областях памяти как пользовательские процессы. Используя эту уникальную архитектуру, вы можете запустить, остановить или обновить поддержку той или иной файловой системы "на лету", без необходимости в перезагрузке.

В дополнение, несколько файловых систем: дисковая, встраиваемая в ППЗУ, CIFS и т.д. - могут выполняться одновременно на одной и той же целевой системе. Они даже могут работать совместно, расширяя возможности друг друга. Например, файловая система со сжатием может работать совместно со встраиваемой файловой системой, существенно снижая потребности вашего устройства в объеме ППЗУ.

### Встраиваемые файловые системы

Встраиваемые файловые системы ОСРВ QNX Neutrino обеспечивают хранение данных и информации для их возможного последующего восстановления.

Возможности включают в себя использование:

- **Образной структуры файловой системы** - поддерживается "выполнение с места" (XIP) для уменьшения потребностей в ОЗУ и для поддержки сверхбыстрой загрузки устройств высокой готовности.
- **В ОЗУ** - для систем, требующих быстродействующего временного хранилища данных с доступом по чтению и записи.

■ **NOR структура файловой системы во флеш-памяти** - предоставляет автоматическое восстановление после сбоев, фоновое стирание и освобождение памяти, а также полное приоритетное прерывание обслуживания для обеспечения производительности в реальном времени. Все операции записи упорядочиваются, чтобы гарантировать целостность файловой системы.

■ **NAND структура файловой системы во флеш-памяти** - позволяет стандартным файловым системам (QNX, Linux или DOS) выполнять непосредственно поверх "сырых" разделов ППЗУ. Поддерживает динамическое распределение и освобождение блоков с полной коррекцией ошибок на основе ECC для продления срока функционирования.

■ **Встраиваемая транзакционная файловая система (ETFS)** - для устройств флеш-памяти типа NAND обеспечивает полную защиту данных во встраиваемых системах при внезапных отключениях электропитания, быстрое время запуска, исправление ошибок в двоичном коде, а также динамическое и статистическое выравнивание степени износа.

Примечание: файловые системы, как в линейном, так и в страничном flash-ППЗУ, являются полнофункциональными, с поддержкой иерархии каталогов, символьных связей и прав доступа в стандарте POSIX.

### Обзор файловых систем

Встраиваемые	Дисковые	Специальные	Сетевые
<b>Образная</b> ROM/Flash "Выполнение с места" (Execute-in-place)	<b>QNX</b> QNX4, QNX6	<b>Со сжатием</b> Декомпрессия "на лету"	<b>NFS</b> Возможность соединения с Unix
<b>В ОЗУ</b> Временное хранилище	<b>Linux</b> Ext2		<b>CIFS</b> Возможность соединения с Microsoft
<b>NOR структура файловой системы во флеш памяти</b> Линейное flash-ППЗУ	<b>DOS</b> FAT 12, 16, 32		
<b>NAND структура файловой системы во флеш памяти</b> Страницное flash-ППЗУ	<b>CD-ROM</b> ISO9660, Joliet		

## Дисковые файловые системы

Все низкоперечисленные файловые системы поддерживают IDE- и SCSI-диски больших размеров:

- **QNX** – полностью реализует семантику POSIX. Подходит для инструментальных рабочих станций и для встраиваемых целевых систем, использующих диски или страничное флаш-ППЗУ.
- **Linux и DOS** – для прозрачного доступа к другим разделам жесткого диска, а также для обмена данными с другими машинами посредством сменных носителей.
- **CD-ROM / DVD** – поддерживает все соответствующие стандарты, включая ISO9660, Joliet и Rock Ridge.

## Файловые системы со сжатием

Используйте эти файловые системы для расширения возможностей других файловых систем:

**Со сжатием** – позволяет разворачивать "на лету" сильно сжатые файлы. Плотность сжатия колеблется от 35% до 55% для исполняемых файлов и до 95% для текста. Используйте эту файловую систему для уменьшения используемой flash-памяти и снижения загрузки сети при доступе к удаленным файлам через NFS или CIFS.

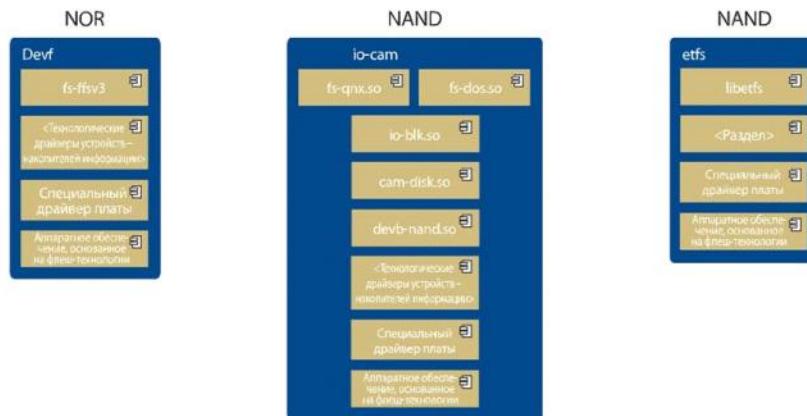
## Сетевые файловые системы

Используйте эти файловые системы для организации прозрачного доступа к файлам на стороне систем под управлением Windows, Unix и Linux. Поддерживаются как CIFS, так и NFS.

## Встраиваемая транзакционная файловая система

Встраиваемая транзакционная файловая система (ETFS) OCPB QNX Neutrino для устройств flash-памяти типа NAND обеспечивает полную защиту данных во встраиваемых системах при внезапных отключении электропитания, быстрое время запуска, исправление ошибок в двоичном коде, а также динамическое и статистическое выравнивание степени износа.

### Структура NOR и NAND файловых систем во флаш-памяти и встраиваемой транзакционной файловой системы (ETFS)



Вы сможете увеличить срок службы ваших устройств, используя функцию выравнивание степени износа, которая увеличивает среднее время между отказами (MTBF) в системах с флаш-памятью.

## Среда управления ресурсами

Среда управления ресурсами в ОСРВ QNX Neutrino может значительно упростить работу в тех случаях, когда необходимо разработать специализированный драйвер для определенного устройства. В сочетании с пакетом разработки драйверов QNX, среда управления ресурсами позволяет Вам ускорить интеграцию пользовательских и стандартных типов драйверов для Вашей встраиваемой системы.

Администратор ресурсов в ОСРВ QNX Neutrino представляет собой не что иное, как пользовательскую программу, которая предоставляет сервисы другим программам и при необходимости взаимодействует с оборудованием. В других операционных системах большая часть этой работы реализуется драйверами устройств. Однако, в отличие от обычных драйверов, администратор ресурсов действует в пользовательском пространстве с защитой памяти, что позволяет запускать и останавливать эту программу динамически, а также отлаживать ее, как любое другое пользовательское приложение.

Ускорению разработки способствует наличие библиотеки, с помощью которой администратор ресурсов может легко отображаться в системное пространство путевых имен и обрабатывать запросы от клиентских программ. Среда управления ресурсами также упрощает создание и освобождение каналов обмена информацией, что идеально подходит для любого служебного процесса, который должен взаимодействовать со множеством клиентских приложений.

Среда управления ресурсами ОСРВ QNX Neutrino:

- упрощает написание многопоточных драйверов благодаря наличию библиотеки пула потоков, которая автоматически создает и удаляет потоки в зависимости от системной нагрузки;
- в сочетании с комплектами разработки драйверов позволяет ускорить разработку драйверов для аудиоустройств, символьных, графических, сетевых, дисковых устройств, устройств ввода, принтеров, USB-устройств и др.;
- позволяет использовать драйверы как перезагружаемые процессы, которые работают в пространстве с защищенной памятью и могут быть отложены с помощью стандартных инструментов на уровне исходных кодов.

### Архитектура для разработки драйверов



Комплекты разработки драйверов (DDK-комплекты) сокращают процедуру написания драйверов до минимума – во многих случаях большинство вашей работы уже сделано.

# Технология минидрайверов

Технология минидрайверов (Instant Device Activation) позволяет создавать системы, которые могут реализовывать часть своего функционала уже на стадии загрузки системы. Вы можете обеспечить выполнение критических функций, таких как управление устройствами, сохранение данных, обеспечение критического времени отклика, менее чем за 50 мс.

## Быстрая активация устройств

Так как минидрайвер (код IDA - Instant Device Activation) является частью программы начальной загрузки, вы можете, например, обеспечить обработку сообщений, поступающих по шине CAN в пределах от 30 до 100 мс и отклик на эти сообщения менее чем за 100 мс. Как только управление передается полной версии драйвера, код IDA может продолжить или завершить работу – без каких либо задержек или потерь данных.

## Снижение стоимости оборудования

Как правило, для «холодного старта» системы или загрузки из режима ожидания требуются дополнительные вычислительные ресурсы, позволяющие удовлетворить требования по минимальному времени отклика. Используя технологию минидрайверов, вы можете решить эту задачу на программном уровне без помощи дополнительного оборудования.

## Простота конфигурирования загрузочных файлов

Разработчикам предоставляются исходные коды, модифицированные загрузочные файлы и документация, позволяющие значительно упростить разработку минидрайверов для различных приложений и аппаратных платформ. Например, вы можете адаптировать стандартные интервалы опроса в соответствии с собственными требованиями, определять объем данных, сохраняемых в буфере минидрайвера, разрабатывать индивидуальные модели переключения между полным и минидрайвером.

### Процесс загрузки Instant Device Activation



Технология минидрайверов устройств позволяет обеспечивать работу критических функций, таких как отклик на внешние события, критическое время отклика, доступ к аппаратному обеспечению менее чем за 50 мс. Как только управление передается полной версии драйвера, код IDA (называемый также «мини-драйвером») может продолжить или завершить собственную работу — без каких-либо задержек или потерь данных.

## Краткий обзор QNX Neutrino

ОСРВ QNX Neutrino обладает всем необходимым для того, чтобы помочь Вам разрабатывать надежные, масштабируемые и высокопроизводительные встраиваемые системы:

### Архитектура на основе микроядра

- Динамически обновляемые сервисы и приложения
- Точная изоляция сбоев и автоматическое самовосстановление системы
- Обмен сообщениями для построения модульных, согласованных систем

### Предсказуемость и производительность реального времени

- Вытесняющий планировщик с выбором дисциплины планирования
- Распределенное наследование приоритетов

### Технология адаптивного квотирования

- Гарантированное выделение системных ресурсов для создания защищенных, отказоустойчивых систем
- Простота системной интеграции посредством декомпозиции

### Поддержка стандарта POSIX

- Поддержка стандарта POSIX 1003.13-2003 с многопоточностью и расширениями реального времени

### Прозрачные распределенные вычисления

- Прозрачный доступ к удаленным ресурсам
- Простота проектирования отказоустойчивых кластеров

### Поддержка многоядерности

- Комплексная поддержка многоядерности
- Асимметричная (AMP), симметрическая (SMP) и исключительная (BMP) многопроцессорность
- Простота перехода с однопроцессорной архитектуры на многопроцессорную

### Расширенная графика

- Окнаная система QNX Photon microGUI
- Сложные многослойные изображения
- Расширяемая поддержка мультимедиа
- Поддержка 3D-графики

### Менеджер высокой готовности

- Отслеживание работоспособности для ранней диагностики отказов
- Интеллектуальное восстановление сбийных компонентов

### Поддержка управления питанием

- Простота создания определяемых приложениями схем управления питанием
- Полный контроль состояния процессора и периферийных устройств

### Файловые системы

- Образная, в ОЗУ, в flash-ППЗУ, QNX4/6, Linux, DOS, CD-ROM, DVD
- NFS, CIFS, со сжатием, ETFS

### Сетевые технологии

- NetBSD (IPv4, IPSec, IPv6) и встраиваемые стеки TCP/IP
- IP-фильтрация и трансляция сетевых адресов (NAT)

### Среда управления ресурсами

- Упрощает написание многопоточных драйверов
- Работает совместно с комплектами разработки драйверов (DDK)
- Позволяет запускать драйверы как перезапускаемые процессы защищенной памяти

### Инструментальное ядро

- Анализ производительности и оптимизация всей системы в целом
- Быстрое выявление ошибок синхронизации и скрытых дефектов

### Поддержка процессоров

- ARM, MIPS, PowerPC, SH-4, X86
- Расширенная поддержка памяти (более 4 ГБ) для x86, MIPS и PowerPC

### Комплект разработчика QNX Momentics

- Графическая интегрированная среды разработки (IDE), BSP-пакеты, инструментарий GNU
- Поддержка множества инструментальных и целевых платформ, а также различных языков программирования



**SWD Software Ltd.**

Платиновый дистрибутор компании QNX Software Systems

196210, Санкт-Петербург, ул. Внуковская, 2,

БЦ "Пулково Скай", офис С-507.

Тел: (812) 611-07-51, 611-07-59, факс: (812) 611-07-58

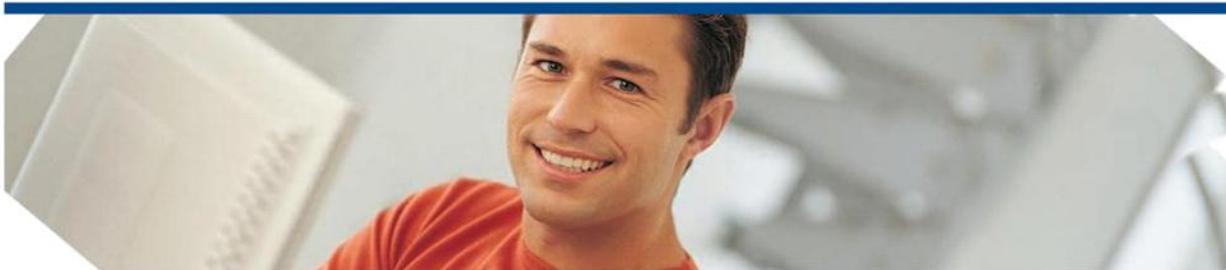
info@swd.ru • www.swd.ru



## QNX NEUTRINO

Операционная система реального времени

**QNX® Momentics®**  
КОМПЛЕКТ РАЗРАБОТЧИКА



Гибкая IDE-среда на основе платформы Eclipse, позволяющая проводить глубокий анализ работы системы с помощью инновационных инструментов профилирования, отладки и оптимизации.

**QNX®**



Уже более 30 лет технологии реального времени QNX помогают создавать самые надёжные в мире приложения: от комплексов сетевой маршрутизации, автомобильных телематических систем до широкомасштабных распределённых систем промышленного управления.

## СОДЕРЖАНИЕ

<b>Открытая платформа Eclipse</b>	2
<b>Средства разработки кода</b>	3
Мастера проектов	3
Редакторы кода	4
Навигатор по исходному тексту	4
Локальный журнал изменений	4
Инструмент управления исходными текстами	5
Структуры make-файлов	5
Инструменты построения	5
Мастер конфигурации запуска	5
<b>Символьный отладчик</b>	6
<b>Системный профайлер</b>	7
<b>Анализатор ОЗУ</b>	9
<b>Профайлер приложений</b>	11
<b>Анализ покрытия кода</b>	13
<b>Средства работы с целевой системой</b>	14
Агент целевой системы	15
Навигатор целевых систем	15
Монитор целевых систем	16
<b>Постройтель приложений</b>	17
<b>Комплекты разработки драйверов</b>	18
<b>Операционная система реального времени QNX Neutrino</b>	19
<b>Краткий обзор QNX Momentics</b>	20

Профессиональный комплект разработчика QNX Momentics® позволяет ускорить все этапы вашего проекта на основе QNX Neutrino: от встраивания на процессорную плату до системного анализа. Он включает в себя всё, что вам необходимо: интуитивно понятную среду разработки, полный набор инструментов, пакеты поддержки процессорных плат и множество готовых к использованию комплектов для разработки драйверов – и всё это в едином высокointегрированном комплекте. Он также обладает чрезвычайно широкой гибкостью, поскольку поддерживает множество языков программирования, множество инструментальных сред и множество целевых процессоров. Свяжитесь с нами прямо сейчас, чтобы узнать, как QNX Momentics® может помочь вам ускорить цикл разработки вашего проекта.

Прежде всего QNX Momentics® дает вам выбор. Вы можете применять разные языки программирования, инструментальные платформы и процессорные архитектуры. Писать программы на C, C++, Embedded C++. Вести разработку в среде Windows, Linux. Компилировать код для целевых процессоров ARM, MIPS, PowerPC, SH-4 или x86. И всё это в одной и той же среде разработки (IDE). Более того, вы можете работать даже с несколькими языками и архитектурами одновременно.

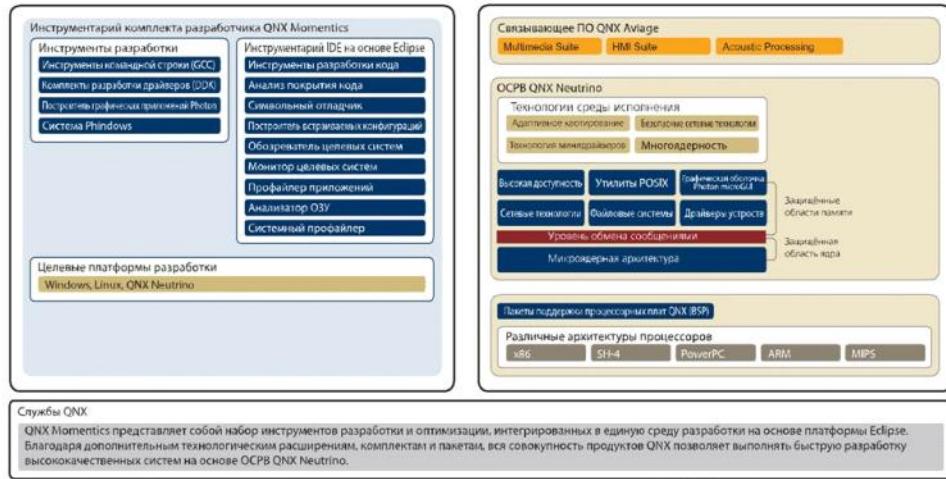
QNX Momentics® упрощает цикл разработки от начала до конца. Вам нужно "оживить" процессорную плату? Просто выберите пакет поддержки процессорных плат (BSP), импортируйте его файл описания в построитель встраиваемых систем и сгенерируйте целевой образ. Вы начинаете новый проект? Запустите специальный мастер, выберите нужные опции, и проект будет создан автоматически. Вы сразу же можете начать писать код с помощью редакторов, разработанных для C и C++.

Вы разрабатываете нестандартный драйвер? Вам помогут комплекты разработки драйверов (DDK), содержащие солидную базу исходных текстов и готовых шаблонов.

Вам нужно оптимизировать производительность? Профайлер приложений и анализатор ОЗУ тесно интегрированы с остальными инструментами среды разработки, что упрощает вашу работу. Вам нужна комплексная отладка и оптимизация системы на низком уровне? Системный профайлер и отладочная версия микроядра позволят визуально отслеживать проблемы синхронизации, взаимные блокировки и прочие часто возникающие ошибочные ситуации.

Профессиональный комплект разработчика QNX Momentics® содержит в себе всё необходимое для разработки и внедрения встраиваемых систем высшего качества.

#### Продуктовая линейка QNX



Продуктовый набор QNX обеспечивает быструю разработку качественных систем на основе ОСРВ QNX Neutrino.

## Открытая платформа Eclipse

С комплектом разработчика QNX Momentics® вы свободны в выборе инструментов благодаря тому, что среда разработки основана на Eclipse – открытой платформе для интеграции инструментария, поддерживаемой большим и постоянно расширяющимся сообществом компаний-производителей инструментов.

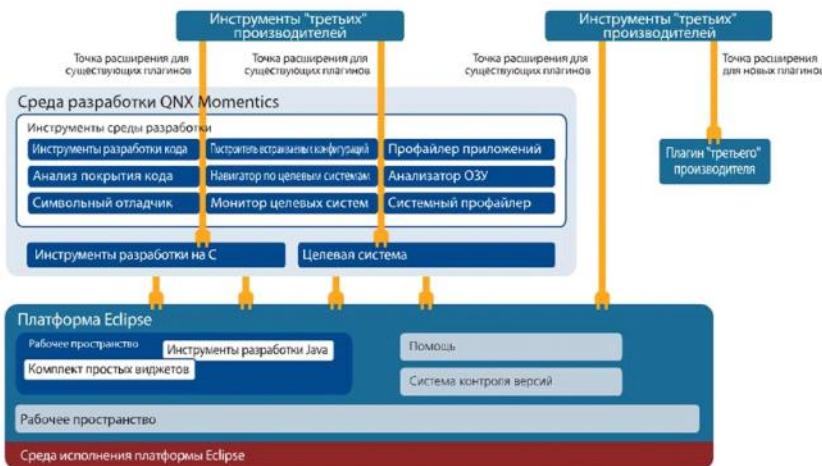
Платформа Eclipse обеспечивает интерфейсы (т. н. точки расширения) для прозрачной интеграции инструментов. В результате весь инструментарий QNX Momentics имеет единый графический стиль. Таким образом, вам не придётся заново изучать пользовательский интерфейс каждого отдельного инструмента или каждой инструментальной платформы.

Платформа Eclipse имеет расширяемую архитектуру подключаемых модулей (плагинов), позволяющую QNX Momentics работать практически с любым типом данных. Например, в состав QNX Momentics входит множество подключаемых модулей для навигации по исходным текстам на C/C++, анализа встраиваемых и загружаемых образов, анализа производительности систем в реальном времени. Вы также можете разрабатывать свои собственные подключаемые модули для работы с любыми другими объектами или использовать готовые модули "третьих" производителей – в обоих случаях новые подключаемые модули прозрачным образом полностью интегрируются с существующими инструментами.

Комплект разработчика QNX Momentics обеспечивает:

- полную интеграцию с Eclipse – открытой платформой для интеграции инструментов;
- поддержку набора функций Eclipse v3.5.x, что ускоряет загрузку IDE и плагинов;
- поддержку новейших возможностей платформы Eclipse C/C++ Development Tools (CDT) v6.x.

### Расширяемая архитектура платформы Eclipse



**Платформа Eclipse имеет расширяемую архитектуру и обеспечивает интерфейсы для прозрачной интеграции инструментов в единой графической среде.**

## Средства разработки кода

QNX Momentics® обеспечивает оптимизированную среду для разработки на C/C++ и Embedded C++. вы можете работать как в командной строке, так и в графической среде, в которой можно пользоваться целым рядом средств для повышения производительности (например, мастерами, редакторами кода, гибкими структурами make-файлов и др.).

### Мастера проектов

Создавайте новые проекты с помощью всего нескольких щелчков мышью. В QNX Momentics вы можете применять встроенные мастера для автоматизации всего процесса.

Мастера позволяют:

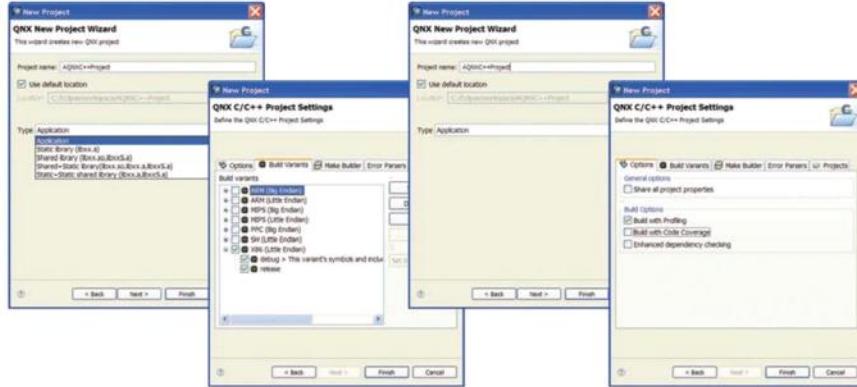
- выбрать тип проекта (C, C++, сборочный файл (makefile), проект QNX);
- пройти все шаги по созданию проекта для любой комбинации целевых процессоров;
- создать для проекта каталоги построения, файлы зависимостей и заготовки исходных текстов;
- автоматически скомпилировать весь проект для создания готового приложения.

### Редакторы кода

Пишите больше кода за меньшее время. Редакторы кода, входящие в состав QNX Momentics, предоставляют все необходимые возможности, включая операции "вырезать/вставить", отмену действий, добавление комментариев и формирование отступов, типичных для используемого языка. Кроме того, редакторы кода обеспечивают следующие функции:

- **Подсветка синтаксиса** – ключевые слова и синтаксис языков программирования автоматически выделяются цветом. Это применяется и к парным скобкам (что удобно при написании множества вложенных секций кода).
- **Контекстно-зависимая справка** – если подвести курсор мыши к имени функции, то редактор отобразит её описание, аргументы и все необходимые заголовки. Если нажать "горячую" клавишу при выбранной функции, то редактор автоматически вставит в текст все необходимые директивы #include.

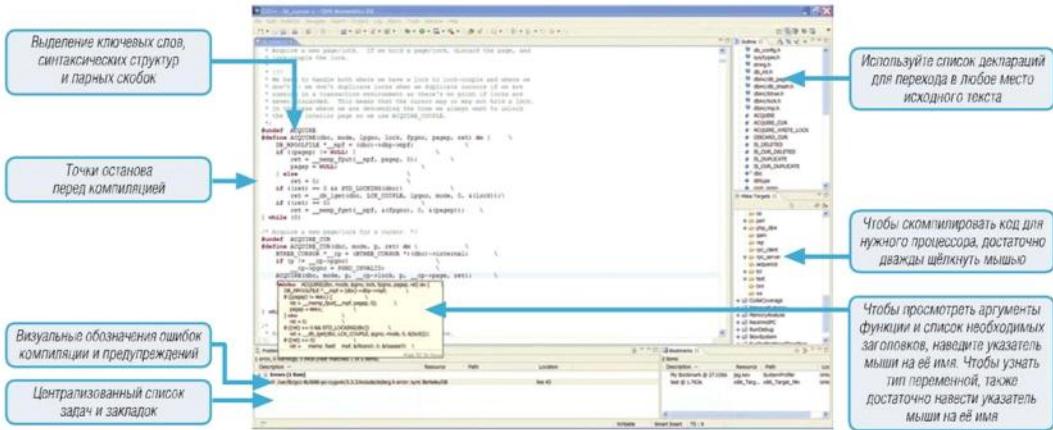
### Мастера проектов



**Встроенные мастера будут сопровождать вас на каждом этапе создания проекта – от выбора процессора до окончательной компиляции кода.**

- **Свертывание кода** – разработчики могут скрыть/отобразить фрагмент кода, над которым они не работают в настоящее время. Функцию «Свертывание кода» (Code folding) можно настроить на отображение отдельных участков кода, когда определенные участки отображаются даже если родительский процесс свернут.
- **Шаблоны кода** – просто нажмите “горячую” клавишу, чтобы вставить в текст типовые часто используемые блоки кода (циклы for, условные выражения if-then-else и т. п.). Каждый редактор содержит набор готовых шаблонов, которые вы можете редактировать или модифицировать.
- **Изменяемые настройки** – задавайте шрифты, цвета и прочие настройки, как вам удобно.
- **Расстановка маркеров в тексте** – каждый редактор отображает в тексте маркеры, связанные с текущим файлом, включая ошибки, указанные средствами построения. Вы также можете расставлять свои собственные маркеры в виде заданий или закладок.
- **Список задач, чтобы лучше организовывать работу** – представляет собой централизованный репозитарий для маркеров (точек останова, ошибок и т. п.) и заданий. Щелкните на ошибке, и редактор автоматически перейдет к нужной строке.

#### Редакторы кода



Редакторы кода имеют множество функций, включая дополнение кода, контекстно-зависимую справку и централизованный список задач, которые ускоряют и упрощают работу

#### Навигатор по исходному тексту

Навигатор по исходному тексту поддерживает контекстно-зависимый поиск ссылок и определений, а также позволяет упростить разработку кода.

Навигатор по исходному тексту выполняет:

- обзорное отображение для быстрой навигации по исходному коду и заголовочным файлам проекта;
- быстрый переход к декларациям и прототипам;
- автозавершение при вводе имен функций.

#### Локальный журнал изменений

Вы можете легко разрабатывать прототипы систем с помощью мощного инструмента “локальный журнал изменений” без необходимости обращения к репозитарию исходных текстов. Вы можете вносить пошаговые улучшения в прототип, при этом будет выполняться автоматическое отслеживание изменений в файле, с которым вы работаете. Таким образом, вы можете легко производить тестирование системы и, если нужно, отменять любые изменения в коде. Локальные версии работают так же, как и версии в репозитарии исходных текстов, поэтому вы можете легко использовать их в утилитах графического сравнения и слияния файлов.

## Инструменты управления исходными текстами

Управляйте всей базой исходных текстов из одной среды. Среда разработки включает в себя встроенную поддержку протокола управления исходными текстами CVS, включая поддержку удалённого сервера и доступ к защищенным репозиториям посредством SSH. Также поддерживаются другие системы управления версиями, такие как ClearCase, Perforce, SVN и т.д.

Поскольку управление версиями и конфигурациями встроено непосредственно в среду разработки, вам не нужно выходить из неё, чтобы управлять своими исходными текстами. Более того, вы можете работать в неоднородной среде управления версиями, используя разные протоколы для разных проектов, и даже для разных файлов в пределах одного и того же проекта.

С помощью инструментов управления исходными текстами вы можете:

- управлять версиями при обновлении своей базы исходных текстов;
- управлять версиями при коллективной разработке;
- просматривать журнал версий, чтобы узнать, кем и какие изменения были внесены;
- выполнять визуальное сравнение различных версий файлов;
- использовать интерактивное слияние версий, чтобы быстро разрешать конфликты между изменениями, внесёнными несколькими разработчиками в один и тот же файл.

## Структуры make-файлов

Выбирайте структуру make-файла, оптимальную для вашего проекта. Когда вы создаете проект на C/C++ в QNX Momentics, вы можете применять следующие структуры make-файла:

- рекурсивная структура для многопроцессорных проектов позволяет быстро компилировать проект для одного или нескольких поддерживаемых целевых процессоров;

## Визуальное сравнение файлов

```

C Compare Viewer
Compare File
1.11
* see ACQUIRE macro on why we handle cursors that don't have
* locks
Bundl# ACQUIRE_WRITE_LOCK
#define ACQUIRE_WRITE_LOCK(dbc, ret) do { \
    while (1) { \
        /* Discard the current page/lock for a cursor. */ \
        #define DISCARD_CURSOR(dbc, ret) do { \
            DBTAK_CURSOR * _cp = (DBTAK_CURSOR *) (dbc->internal); \
            DB_MHFILE *_mpf = (dbc->dbp->mpf); \
            int _t_ret; \
            if (_cp->page) != NULL { \
                _t_ret = _mpf_fputi(_mpf, _cp->page, 0); \
                _cp->page = NULL; \
            } else { \
                _t_ret = 0; \
                if (_cp->page == 0 && _t_ret == 0) \
                    _t_ret = -1; \
                if (_cp->lock) \
                    _t_ret = -1; \
                if (_cp->page == 0 && _t_ret == 0) \
                    _t_ret = 0; \
                if (_t_ret == 0 && !LOCK_ISSET(_cp->lock)) \
                    _cp->lock_mode = DB_LOCK_NO; \
            } \
        } while (0); \
        if (CPD_LOCKED(dbc) && !LOCK_ISSET(_cp->lock)) \
            _t_ret = _mpf_flocki(_mpf, _cp->page, 0, _cp->lock); \
        if (_t_ret == 0) \
            break; \
    } \
} while (0);
* see ACQUIRE macro on why we handle cursors that don't have
* locks
Bundl# ACQUIRE_WRITE_LOCK
#define ACQUIRE_WRITE_LOCK(dbc, ret) do { \
    while (1) { \
        /* Discard the current page/lock for a cursor. */ \
        #define DISCARD_CURSOR(dbc, ret) do { \
            DBTAK_CURSOR * _cp = (DBTAK_CURSOR *) (dbc->internal); \
            DB_MHFILE *_mpf = (dbc->dbp->mpf); \
            int _t_ret; \
            if (_cp->page) != NULL { \
                _t_ret = _mpf_fputi(_mpf, _cp->page, 0); \
                _cp->page = NULL; \
            } else { \
                _t_ret = 0; \
                if (_cp->page == 0 && _t_ret == 0) \
                    _t_ret = -1; \
                if (_cp->lock) \
                    _t_ret = -1; \
                if (_cp->page == 0 && _t_ret == 0) \
                    _t_ret = 0; \
                if (_t_ret == 0 && !LOCK_ISSET(_cp->lock)) \
                    _cp->lock_mode = DB_LOCK_NO; \
            } \
        } while (0); \
        if (CPD_LOCKED(dbc) && !LOCK_ISSET(_cp->lock)) \
            _t_ret = _mpf_flocki(_mpf, _cp->page, 0, _cp->lock); \
        if (_t_ret == 0) \
            break; \
    } \
} while (0);

```

Утилита визуального сравнения файлов быстро находит отличия между двумя версиями файла и выделяет их цветом.

## Средства отладки

QNX Momentics® обеспечивает мощную унифицированную среду отладки для всех поддерживаемых языков программирования. Интуитивно понятный интерфейс отладчика полностью интегрирован с остальными инструментами IDE и даёт максимальную гибкость для решения возникающих задач.

Экранные панели любого другого инструмента среды разработки можно открыть прямо в символьном отладчике, что расширяет возможности представления информации о состоянии и данных приложения.

Символьный отладчик позволяет:

- одновременно отлаживать несколько приложений, написанных на С и С++;
- отлаживать многопоточные приложения, контролируя каждый поток по отдельности и отслеживая передачу управления между ними;
- отлаживать несколько процессов, выполняемых одновременно на разных процессорах, и отслеживать их распределение по процессорам;
- динамически подключаться отладчиком к любому выполняющемуся процессу;
- проводить "посмертный" анализ дамп-файлов.

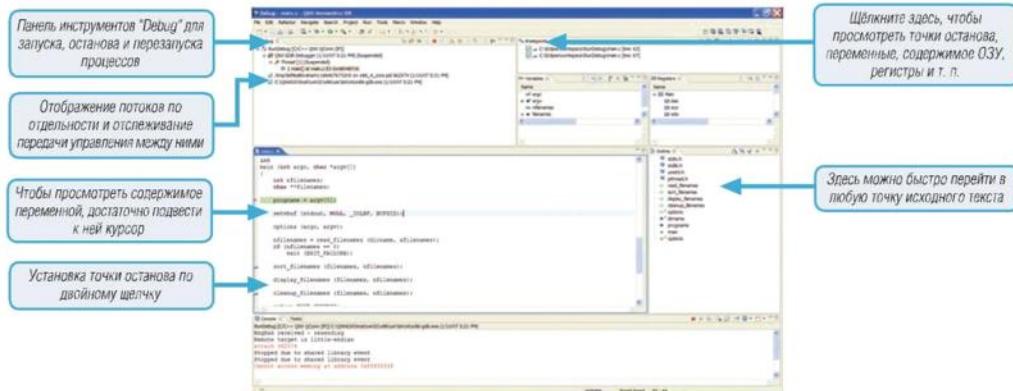
### Выбирайте нужные типы точек останова

Тип точки останова	Описание
Условная	Активизируется в соответствии со значением заданного выражения.
Счётная	Активизируется или вычисляет условное выражение по достижении определенного числа итераций.
Поступающее выполнение	Позволяет выполнять код в пошаговом режиме, а также либо входить в вызываемые функции, либо выполнять их целиком за один шаг. Кроме того, вы можете устанавливать точки останова на момент возврата из функции.
Выполнение до курсора	Позволяет создать временную точку останова, установленную на другую точку в той же функции. Программа будет выполняться до указанной точки.
Сторожевая	Останавливает выполнение программы при изменении значения переменной.

### Используйте дополнительные панели отображения

Экранная панель	Описание
Переменные	Вы можете просматривать переменные, приводя их к любому типу (имеющий, строковый, целый и т. д.). Указатели можно разыменовать или представить в виде массивов. Вы также можете выделять глобальные переменные и поместить их в панель отображения переменных.
Выражения	Эта панель отображает результаты вычисления любых выражений, в том числе переменных, констант, указателей.
ОЗУ	Позволяет редактировать любой блок памяти в пределах адресного пространства программы.
Стек	Отображает состояние стека для любого выбранного потока. На этой же панели можно для сравнения отобразить все выполняемые процессы.
Регистры	Позволяет просмотреть текущее состояние регистров процессора для любого потока.
Точки останова	Отображает все типы точек останова: условные, счётные, поступающие или с выполнением до курсора.

### Символьный отладчик



Символьный отладчик позволяет отлаживать несколько процессов и потоков одновременно, даже если эти процессы написаны на разных языках программирования.

## Системный профайлер

Системный профайлер QNX Momentics® обладает значительно более широкими возможностями, чем обычные инструменты отладки и анализа кода, и позволяет анализировать взаимодействие между всеми компонентами сложной системы реального времени.

С помощью системного профайлера вы можете разрешать конфликты синхронизации, обнаруживать ситуации взаимных блокировок, выявлять корни семантических ошибок, находить скрытые неполадки в программном и аппаратном обеспечении и оптимизировать производительность вашего приложения, причём как для однопроцессорных, так и для многопроцессорных целевых систем.

Системный профайлер позволяет выполнять следующие задачи:

- получать визуальное отображение взаимодействий между компонентами целевой системы, извлекая данные из отладочной версии микроядра;
- определять момент возникновения события, связанные с ним программные модули и действия, выполненные ими, а также интерпретировать возникшее событие;
- получать информацию о событиях на целевой системе, применять динамические фильтры событий, а также получать визуальное отображение и интерпретацию данных о процессорной активности и статистики по событиям;
- отображать сводные данные, в т. ч. относящиеся к многоядерной обработке (например, миграцию потоков между процессорами и обмен сообщениями между ядрами);
- применять специализированные инструменты для определения текущего статуса потоков и причин их работы;

- создавать несколько оконных панелей для одновременного отображения информации о разных аспектах трассировки системы; эти панели можно синхронизовать по масштабу времени.

### Визуальное наблюдение событий для быстрого выявления проблем

Системный профайлер может отображать огромные объёмы информации, включая информацию о вызовах ядра, аппаратных прерываниях, состояниях потоков, обмене сообщениями и событиях планировщика. Кроме этого, он даёт вам чёткий контроль над тем, какие события записывать и когда, чтобы вы могли сосредоточиться именно на том, что вам нужно. Он также позволяет применять фильтры событий и различные опции отображения, чтобы детально анализировать подозрительные фрагменты кода и легко прослеживать сложные взаимодействия.

### Отладка и оптимизация многоядерных систем

Системный профайлер работает на основе QNX Neutrino, первой ОСРВ, поддерживающей симметричную многопроцессорность (SMP). Он обладает данными обо всех потоках, запущенных на различных

#### Системный профайлер



С помощью системного профайлера вы можете разрешать конфликты синхронизации, определять ситуации взаимных блокировок, выявлять корни семантических ошибок и другие причины снижения производительности системы.

процессорах, и помечает их различными цветами, что существенно облегчает восприятие. Например, системный профайлер можно использовать для обнаружения сбоев в работе ядра, перенесенного с однодеревых устройств. Сводка миграций потоков выводит сведения о перемещении отдельно взятого потока между ядрами, которые запланированы планировщиком работы ядер, — помогает выявлять проблемы пробковки или кэширования. Сводка обмена сообщениями между ядрами показывает частоту обмена сообщениями между ядрами в системе — её удобно просматривать для выявления плотных взаимодействующих потоков. В каждом из случаев системный профайлер может вывести информацию по каждому процессору в отдельности.

## Полная информация о событиях в системе

При просмотре истории событий в системе часто задаётся вопрос: "Почему этот поток запущен именно сейчас?" Системный профайлер снабжён утилитой "причина запуска", которая записывает события, приведшие поток к данному состоянию, и при необходимости

прослеживает серию событий до самого начала следа. Также системный профайлер может предоставить статистику клиента/сервера, включающую точное количество времени, требуемое клиенту, и что требуется от серверов, которые он использует. Утилита просмотра мгновенного состояния потока в системном профайльере позволяет просмотреть сводку по каждому потоку и его состояние в выбранный момент времени — вы можете узнать текущее состояние любого процесса в системе в любой выбранный вами момент времени.

## Собственные фильтры событий

Если штатные фильтры событий, предоставляемые отладочной версией микроядра, не соответствуют ситуации, которую вы хотели бы проанализировать, это не проблема: ядро поддерживает систему динамических фильтров, определяемых пользователем, чтобы эффективно отслеживать сложные комбинации условий. Таким образом, вы можете сосредоточиться на отдельных событиях и уменьшить объём данных, выводимых для анализа.

Разделы системного профайльера	Описание
<b>Журнал истории событий</b>	Представляет подробную информацию по каждому событию в истории ядра. Журнал событий находится в постоянной синхронизации с выбранной временной шкалой.
<b>Закладки</b>	Позволяет вам отмечать отдельные диапазоны мест и событий, отображаемых в редакторе системного профайльера, и в последствии просматривать отмеченные события в разделе закладок.
<b>Статистика клиента/сервера</b>	Представляет более подробную информацию о работе системы, измеряя время, отведённое на каждый служебный поток клиентам.
<b>Выборочная статистика</b>	Выводит статистический счёт в виде таблицы событий в системе, которые отвечают заданным пользователем условиям. Статистика может быть основана на данных всего журнала событий или же только на выбранном диапазоне.
<b>Миграция между процессорами</b>	Планирует переход потоков с одного процессора на другой (миграцию), что необходимо при работе OCPB QNX Neutrino в режиме симметричной многопроцессорности для обеспечения полного использования ресурсов процессоров. В данном разделе представляются данные о степени миграции потоков в системе.
<b>Статистика принадлежности событий</b>	Выводит статистический отчёт в виде таблицы отдельных событий на основе принадлежности процесса/потока. Статистика может быть основана на данных всего журнала событий или же только на выбранном диапазоне.
<b>Общая статистика</b>	Выводит общий статистический отчёт в виде таблицы истории событий в ядре. Статистика может быть основана на данных всего журнала событий или же только на выбранном диапазоне.
<b>Обзор</b>	Выводит графическое представление использования процессора и распределения событий для всей истории событий в ядре в независимости от текущего выбранного диапазона.
<b>Сводка по декомпозиции</b>	Выводит сводку всего журнала событий, делая акцент на технологии аддитивной декомпозиции OCPB QNX Neutrino. Для каждой отдельной конфигурации групп, найденных в журнале событий, выводится распределение используемых группой процессорных ресурсов вместе с таблицей, отображающей распределение процессорных ресурсов по каждому событию внутри каждой группы.
<b>Мгновенное состояние потока</b>	Отображает количество потоков в каждом из возможных состояний на выбранный момент времени истории. В любом из состояний можно выделить отдельные потоки.
<b>Причина запуска</b>	Работает совместно с панелью редактора временной шкалы системного профайльера и при наведении курсора на какой-либо поток и нажатии кнопки мыши отвечает на вопрос "Почему этот поток запущен?".

## Анализатор ОЗУ

Анализатор ОЗУ совмещает в себе расширенные возможности визуализации с инновационным методом записи данных об использовании памяти.

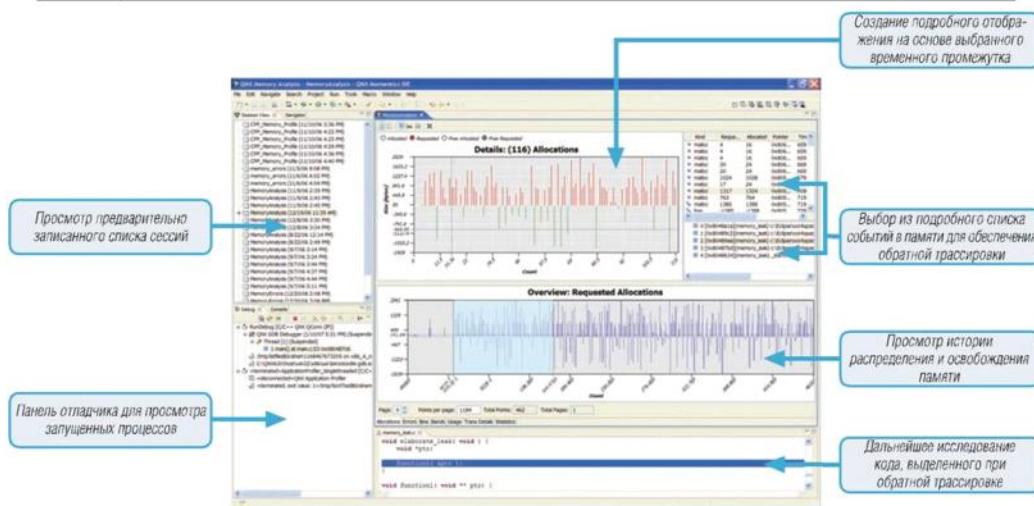
Анализатор ОЗУ помогает визуально отобразить использование памяти вашими программами и может быстро выявлять переполнения буферов, некорректные освобождения памяти и множество других типовых ошибочных ситуаций. Анализатор ОЗУ позволяет получить:

- уникальные инструменты для визуализации, представляющие графический вид профилей памяти;
- информацию на уровне процесса, с помощью которой можно быстро оценить карту памяти программы; закрашенные области представляют собой стек, код, библиотеки и "кучу";
- специализированную статистику распределения памяти для выявления возможных проблем утечки памяти; статистика содержит суммарное количество свободных, распределенных и используемых байтов и блоков;
- динамический журнал использования памяти для оценки изменений.

### Выявление ошибок распределения памяти при помощи штатной библиотеки

Комплект разработчика QNX Momentics содержит библиотеку распределения памяти, в которой реализовано большинство типовых операций над строками и памятью. Эти функции перед выполнением операции проверяют корректность использования указанной области памяти, позволяя разработчикам выявлять ошибки переполнения, выборки из пустого буфера, некорректного использования памяти и повторного освобождения одной и той же области. При обнаружении ошибки разработчик автоматически переводится при помощи обратной трассировки событий в редактор кода на нужную строку кода. Эти ошибки можно быстро исправить и провести тестирование заново.

#### Анализатор ОЗУ



Анализатор ОЗУ обеспечивает визуализацию использования памяти для понимания того, где и как в системе используется память.

## Выбирайте наиболее подходящий способ обработки ошибок

Анализатор ОЗУ обладает "интеллектуальным" механизмом отслеживания ошибок работы с памятью. При возникновении ошибки, соответствующий фрагмент исходного текста автоматически помечается предупреждением. В этом случае разработчик может:

- продолжить выполнение программы;
- завершить программу и сохранить её образ в дамп-файле;
- остановить программу и сразу переключиться в отладчик, где в вашем распоряжении будут все необходимые средства для локализации проблемы.

В любом случае, анализатор ОЗУ обнаруживает проблему и отслеживает по истории стека вызовов путь к причине возникновения ошибки. Производится быстрое исправление ошибок и повторное тестирование.

## Профиль использования памяти для более глубокого понимания

Понимание профиля памяти системы является ключевым моментом в оптимизации ресурсов памяти. Анализатор ОЗУ трассирует все распределения и освобождения памяти и отображает их упорядоченными по времени, чтобы помочь понять где, когда и как используется память.

## Оптимизация ресурсов памяти

При помощи технологии профилирования памяти анализатор ОЗУ помогает оценить возможности оптимизации использования памяти. Например, профилирование памяти может обнаружить интенсивное использование памяти в определённом объёме в конкретном времянном промежутке. Профиль памяти отображает время распределения памяти, место в коде этого распределения, как часто это происходит и в каких временных промежутках. Нежелательные распределения памяти можно легко обнаружить в коде, где разработчики могут применить более эффективную схему.

Тип ошибки памяти	Описание
<b>Нелегальное освобождение памяти</b>	Происходит при попытке вызова функции <code>free()</code> библиотеки С с пустым указателем, указателем на стек или статическую память, или с указателем на тип памяти, который не указывает на начало распределенного блока, или при попытке произвести повторное освобождение.
<b>Разыменование пустых указателей</b>	Регистрирует ошибки работы символьской памяти и строк, вызываемых в программе.
<b>Переполнение буфера</b>	Выдаётся при нечаянной записи программой в область памяти, которой не хватает для указанного буфера.
<b>Использование освобождённой памяти</b>	Выдаёт ошибку памяти, когда производится попытка прочитать или записать в память, которая была только что освобождена. Например, программа А вызывает функцию <code>free()</code> для одного блока, но продолжает после этого его использовать, тем самым создавая проблему повторного использования при произведении запроса <code>malloc()</code> .
<b>Чтение неинициализированной памяти</b>	Выдаёт ошибку памяти при попытке прочитать или записать в память, которая была освобождена по причине отсутствия инициализации.
<b>Утечка памяти</b>	Происходит, когда программа распределяет память, но не освобождает её в последствии. Например, функция А возвращает, когда локальная переменная является указателем на распределённую память, которая не была освобождена.

## Профайлер приложений

Профайлер приложений комплекта разработчика QNX Momentics® позволяет оценивать общую производительность программ независимо от их сложности или объема и без необходимости построчной обработки.

С помощью профайлера приложений вы можете быстро обнаруживать чрезмерно интенсивно используемые секции кода и применять к ним отладку, анализ производительности и оптимизацию. Например, профайлер приложений позволяет выполнять следующие задачи:

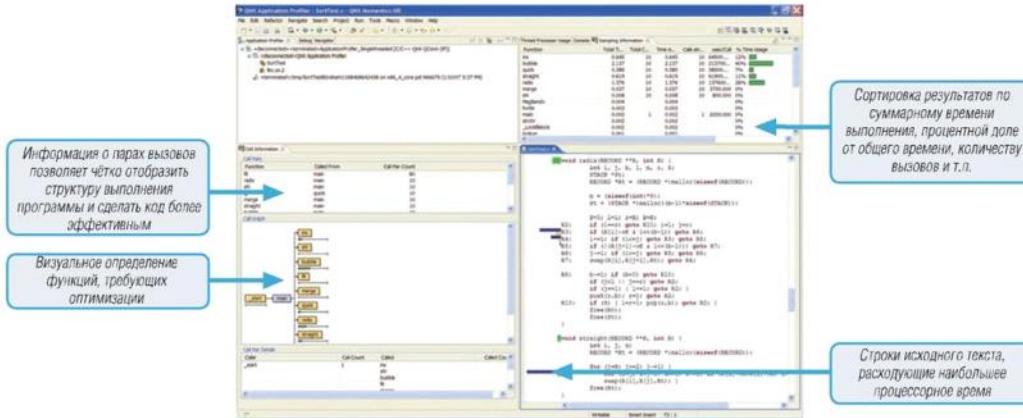
- в реальном времени собирать информацию, подключив профайлер к программе, выполняющейся на целевой системе;
- на уровне исходного текста выявлять строки, потребляющие наибольшие процессорные ресурсы;
- анализировать использование процессорного времени по множеству процессов и целевых систем, а также разделяемых библиотек;
- получать полную информацию о выполнении с помощью компилирования и запуска новой копии программы на целевой системе;
- производить "посмертное" профилирование и анализ посредством загрузки файлов статистики в профайлер.

### Статистическое профилирование для неаггрессивного контроля

Сканируя выполнение программы через фиксированные временные интервалы, профайлер приложений строит точную картину того, в каких участках кода система тратит больше всего времени. Для этого не требуется специальный инструментарий, изменение кода или особый режим компиляции. Неаггрессивное профилирование также гарантирует, что профайлер не исказит собираемую информацию.

Профайлер приложений предоставляет информацию об использовании процессорного времени каждым потоком и отображает её как в абсолютных значениях, так и в процентах от общего времени, после чего её можно сортировать удобным образом. Профайлер приложений также позволяет начать процесс профилирования заново без перезапуска самого приложения. В итоге, вы можете сравнить результаты для одного и того же приложения при различной нагрузке. Более того, профайлер может анализировать динамически загружаемые разделяемые библиотеки, что позволяет определить, где кроется причина снижения производительности – в коде приложения или в вызываемой им библиотеке.

### Профайлер приложений



Профайлер приложений позволяет анализировать использование приложением процессорных ресурсов и разделяемых библиотек на различных целевых системах. Профилирование может быть как "посмертным", так и в реальном времени.

### Внедрение диагностических тегов для точных измерений

Чтобы получить дополнительную информацию о выполнении кода, вы можете дать указание компилятору внедрить в результирующий код диагностические теги для профилирования. Код с диагностическими тегами будет выдавать информацию о вызовах функций и с парах вызовов (т. е. откуда и куда был вызов). С помощью информации о вызовах и статистики выполнения вы можете локализовать "узкие места", а затем перейти к визуальной диаграмме вызовов, чтобы определить место функции в цепочке вызовов. Это весьма удобный способ поиска фрагментов кода, требующих оптимизации.

### Выявление неэффективных фрагментов с точностью до строки исходного текста

Для оптимизации производительности часто недостаточно знать, какие именно функции потребляют наибольшее процессорное время. Чтобы получить более детальные сведения, профайлер приложений может спуститься до уровня исходного текста и показать, какие именно строки исходного текста потребляют наибольший процессорный ресурс. Это происходит с помощью графического аннотирования исходного текста в редакторе кода на C/C++ и отображения потребляемой доли процессорного времени – как для отдельных функций, так и для отдельных строчек кода. Вы можете визуально сравнить эффективность различных стратегий оптимизации, выявить неэффективные алгоритмы и сосредоточиться на оптимизации нужных фрагментов кода, просто просматривая исходный текст.

### Широкие возможности реализации параллелизма

Новейшая многоядерная технология QNX в сочетании с поддержкой симметричной многопроцессорности (SMP) позволяет реализовать истинный параллелизм во встраиваемых системах. Профайлер приложений помогает быстро ответить на вопрос "В каких точках можно оптимизировать систему для работы в многоядерном режиме?" посредством выявления тех частей кода, которые могут подходить для реализации параллелизма. После первоначального переноса кода на многоядерную систему разработчики могут быстро выделить код для параллелизации на основе анализа, выполненного с помощью профайлера приложений. Например, высокоактивную подпрограмму обработки сигналов можно разбить на несколько многопотоковых подпрограмм для параллельного исполнения на SMP-ядре (например, OCPB QNX Neutrino).

## Анализ покрытия кода

Ускорьте этапы оптимизации, тестирования и контроля качества программного продукта в процессе разработки. Инструмент анализа покрытия кода, входящий в состав профессионального комплекта разработчика QNX Momentics®, реализует специальную методологию для обеспечения качества программного кода и целостности продуктов.

Данный инструмент позволяет выделить ветви исходного кода, не прошедшие тестирование, и при необходимости, модифицировать или удалить этот сегмент.

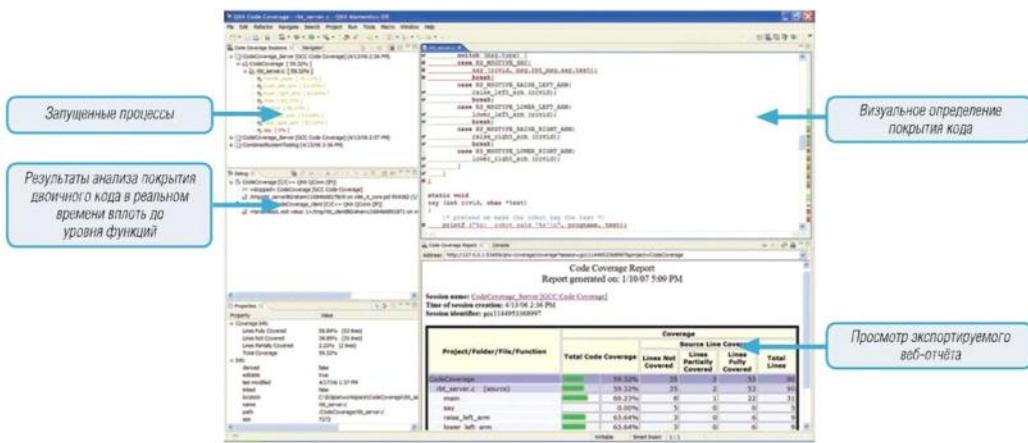
Благодаря полной интеграции этого инструмента со средой разработки, оптимизация, тестирование и контроль качества программного кода становятся значительно проще.

Анализ покрытия кода – важнейший инструмент в тех случаях, когда тестирование, исправление ошибок и сопровождение программного продукта осуществляется разными группами инженеров, которые могли и не участвовать в разработке кода.

Инструмент анализа покрытия кода позволяет:

- запускать сессию анализа покрытия кода и непосредственно наблюдать работу приложения в процессе его выполнения;
- получать результаты анализа покрытия двоичного кода в реальном времени вплоть до уровня основных блоков (ветвей кода);
- запускать редактор кода для быстрого определения, какие строки кода были покрыты и какие нет;
- использовать все инструменты навигации и комментирования, имеющиеся в среде разработки;
- наблюдать за покрытием кода одновременно нескольких приложений;
- генерировать отчеты для последующего анализа.

### Анализ покрытия кода



Анализатор кода QNX Momentics позволяет проводить точное и полное тестирование программного обеспечения.

## Средства работы с целевой системой

QNX Momentics® предоставляет полный набор инструментов для начальной загрузки и взаимодействия с целевым оборудованием. В этот инструментарий входят пакеты поддержки для широкого спектра процессорных плат, построитель встраиваемых систем, позволяющий быстро формировать и настраивать целевые образы, и уникальный расширяемый целевой агент, обеспечивающий одновременное выполнение множества задач для настройки целевой системы, в том числе, отладки, профилирования и сбора системной информации.

### Постройтель встраиваемых систем

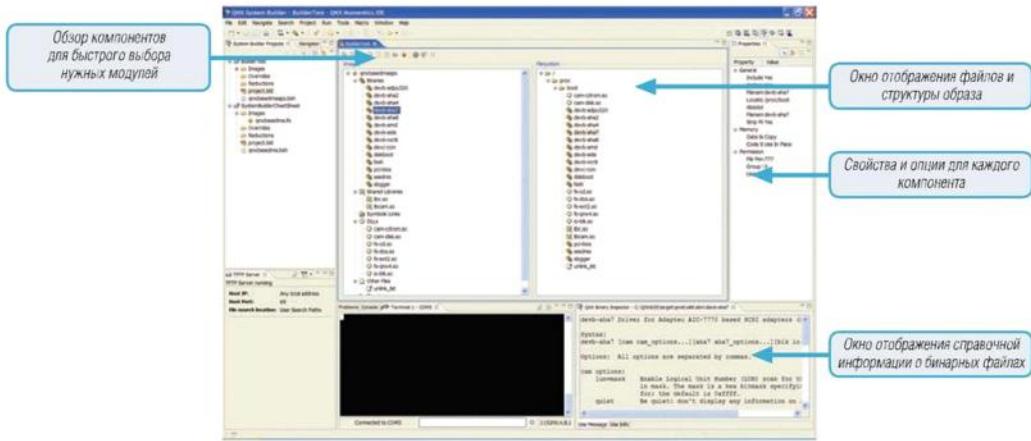
Постройтель встраиваемых систем позволяет существенно сэкономить время при создании загрузочных образов и флеш-образов встраиваемых файловых систем для ваших целевых систем. Для каждого создаваемого образа построитель встраиваемых систем позволяет выполнять следующие задачи:

- автоматизировать создание новых BSP-проектов в среде разработки;
- импортировать существующее описание образа из BSP-пакета или создать свой собственный файл описания;
- использовать браузер компонентов образа для быстрого выбора нужных двоичных модулей, динамически подключаемых и других библиотек;

- проверять взаимные зависимости библиотек и получать предупреждения об отсутствующих компонентах;
- гарантировать применение последней версии проекта в образе;
- сокращать объём памяти, занимаемый приложением, с помощью исключения ненужных функций из разделяемых библиотек.

Для переноса полученного образа на целевую систему в построителе встраиваемых систем имеется встроенный последовательный терминал, который может взаимодействовать с удалёнными мониторами ПЗУ. Он также может пересыпать образы по TFTP или BOOTP. Запустив образ на целевой системе, вы сможете внести дополнительные изменения и пересыпать файлы при помощи любого из доступных механизмов. Например, вы можете редактировать файлы прямо на целевой системе, используя редактор среды разработки.

### Постройтель встраиваемых систем



Постройтель встраиваемых систем значительно сокращает время создания, оптимизации и передачи целевых образов.

## Агент целевой системы

Агент целевой системы представляет среду разработки с расширяемым механизмом взаимодействия с одной или несколькими целевыми системами. Каждый инструмент в составе среды разработки может взаимодействовать с целевой системой через агента, запуская на ней нужный модуль при запросе соответствующей службы с инструментальной машины. Когда служба больше не требуется, модуль можно выгрузить, чтобы освободить ресурсы. Как и любой другой драйвер в QNX, агент целевой системы может быть при необходимости запущен или выгружен в любой момент.

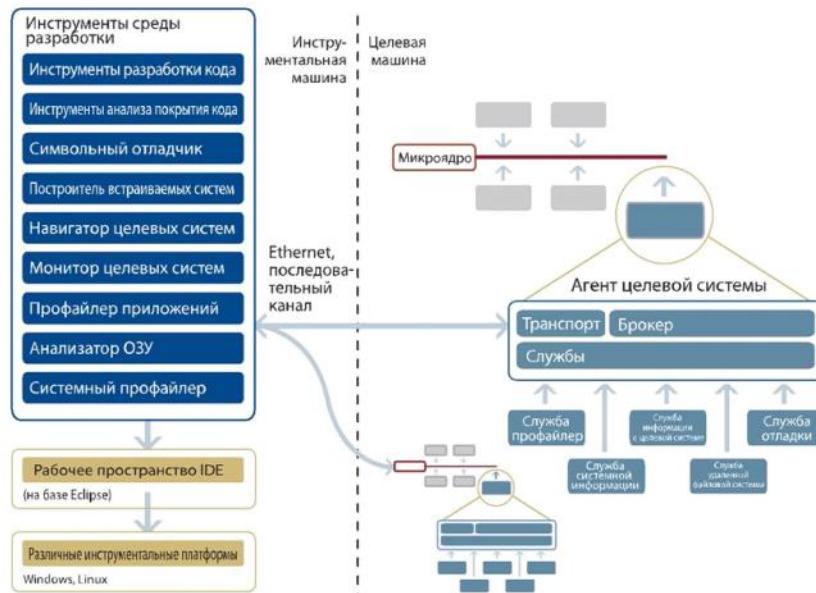
## Навигатор целевых систем

Навигатор целевых систем позволяет всем инструментам среды разработки согласованно взаимодействовать с целевыми системами. Вы можете использовать этот навигатор для определения целевых систем и подключения к ним. С помощью навигатора целевых систем вы также можете связать текущий проект (например, образ системы) с той или иной целевой системой. Кроме того, навигатор целевых систем позволяет отобразить доступные аппаратные компоненты и устройства целевой системы.

После определения целевой системы навигатор целевых систем позволяет выбрать целевой компонент для применения последующих операций, например запуска telnet-сессии.

Новые инструменты могут использовать отображаемый целевой компонент как точку расширения, чтобы добавить свои операции в контекстное меню целевых систем. Например, такими операциями могут быть отправка сигнала процессу, выполняемого на целевой системе, подключение отладчика к выполняемому процессу.

### Агент целевой системы



Агент целевой системы работает как брокер соединений, позволяя навигатору целевых систем (и другим инструментам среды разработки) взаимодействовать с одной или несколькими целевыми системами.

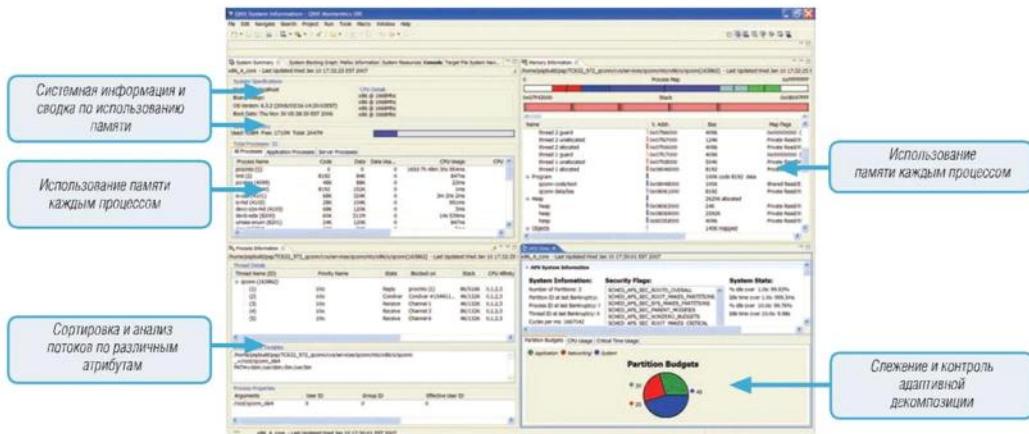
## Монитор целевых систем

Монитор целевых систем предоставляет огромный объём информации о системах и процессах и генерирует отчёты (как в реальном времени, так и в "посмертном" варианте) для выбранной целевой системы. При выборе целевой системы сразу же происходит обновление всей информации о текущих процессах в ней. Вы можете начать с общего обзора системы, просмотрев список выполняющихся процессов и их аргументы. Вы также можете наблюдать за использованием процессора и памяти, как для системы в целом, так и для каждого процесса в отдельности. Затем вы можете получить более детальный анализ с помощью инструментов, позволяющих отследить атрибуты потоков (например, состояние, дисциплину планирования, использование процессора, размер стека), состояния сигналов, карты памяти программы, файловые дескрипторы и т. д.

Монитор целевых систем позволяет:

- отслеживать интенсивность использования ресурсов (например, расход памяти и ресурсов процессора, количество открытых файлов), в реальном времени наблюдая за процессами и потоками;
- выявлять потенциальные ситуации взаимных блокировок с помощью графической диаграммы отношений блокирования между процессами;
- управлять файлами на удалённой целевой системе напрямую с инструментальной машины;
- легко копировать файлы на удалённую целевую систему и редактировать их при помощи редактора напрямую из среды разработки;
- расширять возможности загрузчика программ для запуска исполняемых модулей на целевой системе простым двойным щелчком мышью.

## Монитор целевых систем



Монитор целевых систем собирает полную информацию об атрибутиках потоков, использование процессора, открытых файловых дескрипторах, блокировках и множестве других параметров.

## Photon Application Builder (PhAB)

С помощью Photon Application Builder (графического инструмента разработки приложений для оконной системы QNX Photon® microGUI®) вы можете легко, одной мышью создавать полнофункциональные пользовательские интерфейсы. Фактически, вам не приходится писать никакого кода – построитель приложений может автоматически сгенерировать полностью работающий прототип графического интерфейса вашего приложения.

С помощью Photon Application Builder вы можете:

- быстро начать разработку, взяв за основу существующие шаблоны, либо строить пользовательский интерфейс из обширной палитры готовых элементов управления (виджетов);
- привязывать диалоговые окна и меню непосредственно к виджетам или добавлять вызовы функций;
- динамически генерировать копии ранее созданных виджетов в любом необходимом для приложения количестве;
- переводить пользовательский интерфейс на различные языки без перекомпиляции и перекомпоновки кода; версии для всех языков могут использовать один и тот же исполняемый модуль.

Photon Application Builder предоставляет компоненты пользовательского интерфейса для самых различных приложений: от карманных устройств до операторских терминалов.

### Базовые компоненты

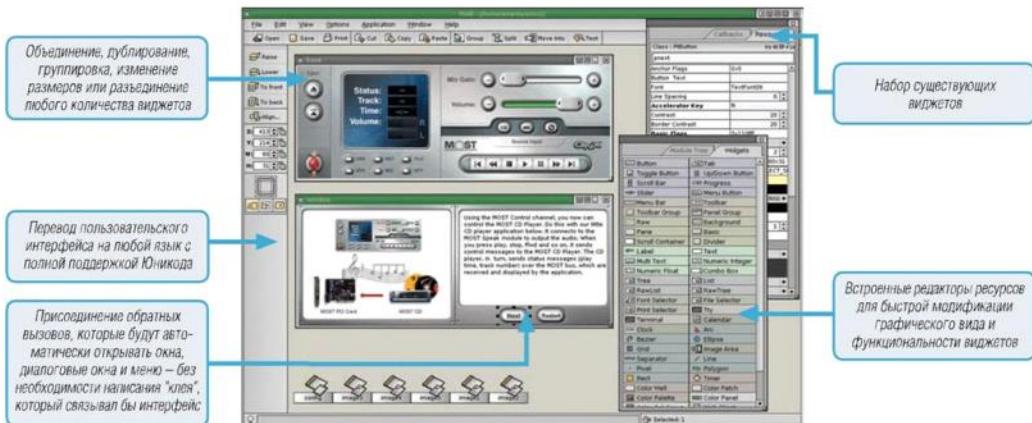
- Кнопки
- Контейнеры
- Измерительные шкалы
- Векторная графика
- Изображения
- Списки
- Деревья
- Меню
- Ярлыки и текст

### Дополнительные компоненты

- HTML
- Терминал
- Выбор шрифта
- Выбор принтера
- Flash
- Контейнер видео

*Примечание: построитель приложений поддерживается только на инструментальной платформе QNX Neutrino.*

### Постройтель приложений



## Комплекты разработки драйверов

С помощью комплектов разработки драйверов (DDK) вы можете быстро создавать драйверы для нестандартного оборудования – аудио-, графических и сетевых адаптеров, устройств ввода, принтеров, символьных и USB-устройств.

Комплекты содержат готовый программный каркас для написания администраторов ресурсов и классов драйверов устройств, а также включают в себя детальную документацию и исходные тексты. Программный каркас драйверов реализует весь высокуровневый аппаратно-независимый код в виде библиотек, поэтому вам остаётся сосредоточиться только на аппаратно-зависимом коде для микросхемы, используемой в вашем устройстве. Даже если вам понадобится разработать драйвер для устройства нового типа, для которого не существует DDK, вы сможете использовать этот каркас как основу для быстрого старта.

### Отлаживайте драйверы в исходном тексте с помощью обычных инструментов IDE

Поскольку в QNX драйверы выполняются как обычные пользовательские процессы, их можно отлаживать и оптимизировать при помощи того же интегрированного инструментария, который в QNX Momentics служит для отладки обычных приложений. Нет никакой необходимости применять отладчики на уровне ядра, так как это может застопорить работу всей целевой системы и, в результате, скрыть ошибки в коде.

Более того, микроядерная архитектура QNX Neutrino позволяет тестировать изменения в коде драйверов без перезагрузки системы и даже без перезапуска сеанса отладки – просто перекомпилируйте и перезапустите драйвер.

### Отлаживайте драйверы прямо на своём компьютере

Если вы предпочитаете резидентную модель разработки и программируете непосредственно в среде QNX Neutrino, вы можете спокойно тестировать и отлаживать драйверы прямо на своей инструментальной машине. Драйверы выполняются в защищенной области памяти, поэтому вы можете применять стандартные инструменты отладки исходного кода на той же самой машине. Более того, исходный код драйверов является совместимым, поэтому в него не требуется вносить изменения, чтобы компилировать для разных процессорных архитектур.

### Архитектура DDK-комплектов



Комплекты разработки драйверов сокращают процедуру написания драйверов до минимума – во многих случаях большинство вашей работы уже сделано.

## Операционная система реального времени QNX Neutrino

Основой профессионального комплекта разработчика QNX Momentics® является ОС реального времени QNX Neutrino, которая обладает уникальной репутацией, подтверждённой годами бесперебойной работы – 24 часа в сутки, 365 дней в году, без остановки. Что делает ОС QNX Neutrino столь надёжной? Ответ прост: это настоящая операционная система на основе микроядра.

В QNX Neutrino ядро обрабатывает только базовые примитивы ОС. Все остальные компоненты – драйверы, файловые системы, стеки протоколов, пользовательские приложения – выполняются вне пределов ядра как отдельные процессы, каждый в своём защищённом адресном пространстве. Такой подход автоматически обеспечивает системам на основе QNX Neutrino «встроенную» отказоустойчивость.

Не менее важной особенностью является и то, что все компоненты QNX Neutrino взаимодействуют друг с другом через единый, чётко детерминированный механизм – синхронный обмен сообщениями. Этот механизм образует между компонентами системы виртуальную "программную шину", позволяющую подключать к ней или, наоборот, отключать любой компонент "на лету". Более того, сообщения могут свободно передаваться между узлами вычислительной сети, предоставляя прозрачный доступ к любому ресурсу, где бы он ни находился.

Используя QNX Neutrino, вы можете осуществлять следующие задачи.

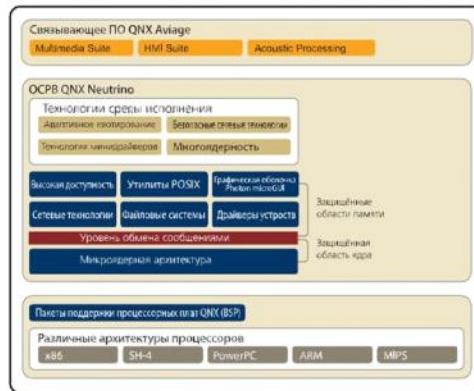
- **Создавать самовосстанавливающиеся системы.** В QNX Neutrino любой компонент в случае отказа может быть перезагружен динамически, не нарушая работу микроядра и других компонентов. Например, если драйвер пытается обратиться к памяти за

пределами своего адресного пространства (что для большинства ОС является фатальной ошибкой), QNX Neutrino корректно завершит этот драйвер и освободит все занятые им ресурсы. Вы можете даже автоматически перезапустить этот драйвер, используя монитор ключевых процессов QNX Neutrino.

■ **Применять одну и ту же ОС во всей своей линейке продуктов.** Благодаря исключительной модульности QNX Neutrino, любые уже испытанные и проверенные компоненты – драйверы, приложения, дополнительные сервисы ОС – можно использовать повторно в других продуктах. Фактически, вы можете создать универсальный набор бинарных модулей и затем применять его либо в однопроцессорном устройстве, либо в SMP-системе, либо в вычислительном кластере. Независимо от масштаба и сложности разрабатываемой системы вы сможете использовать одну и ту же ОС, один и тот же программный интерфейс приложения (API) и один и тот же инструментарий разработчика.

■ **Производить модернизацию систем, не останавливая их.** Поскольку практически любой компонент в QNX Neutrino может быть добавлен или удалён динамически, ваша система может продолжать работу даже во время замены или добавления в неё новых приложений, драйверов или стеков протоколов.

### ОСРВ QNX Neutrino



Благодаря микроподдернистичной архитектуре с защитой памяти, ОСРВ QNX Neutrino обеспечивает надёжную основу для создания систем, способных автоматически обнаруживать ошибки и самовосстанавливаться, выполнять динамическую модернизацию и обладать широкой масштабируемостью.

## Краткий обзор QNX Momentics

От инструментов встраивания на процессорную плату до инструментов системного анализа – QNX Momentics включает в себя всё, что вам необходимо:



### ОС реального времени QNX Neutrino

- Высоконадёжная архитектура на основе микроядра
- Прозрачные распределённые вычисления
- Симметричная многопроцессорность
- Отладочная версия микроядра
- Окна система QNX Photon microGUI
- Сотни утилит POSIX, UNIX и QNX

### Полностью интегрированная Среда разработки

- Платформа Eclipse как основа
- Средства разработки кода на C, C++, embedded C++
- Средства управления версиями
- Анализ покрытия кода
- Символьный отладчик
- Построитель встраиваемых систем
- Монитор целевых систем
- Агент целевой системы
- Профайлер приложений
- Анализатор ОЗУ
- Системный профайлер

### Различные инструментальные платформы

- QNX Neutrino 6.5
- Windows 2000, XP, Vista, 7
- Red Hat Enterprise Linux Workstation 5.4
- Red Hat Fedora 12
- Ubuntu Workstation 9.10
- openSUSE 11.2

### Библиотеки и инструментарий GNU

- ANSI C
- Dinkum C++, Embedded C++
- Оптимизирующие компиляторы GCC v4.4
- Компилятор Intel C++ Compiler
- GDB 6.7

### Построитель приложений

- Инструмент графической разработки для создания полнофункциональных пользовательских интерфейсов
- Готовые шаблоны и виджеты для разработки пользовательских интерфейсов
- Перевод пользовательских интерфейсов на различные языки без перекомпиляции и перекомпоновки кода

### Пакеты поддержки процессорных плат (двоичная форма)

- Для популярных процессорных плат на основе ARM, MIPS, PowerPC, SH-4, x86

### Комплекты разработки драйверов

- Для аудио-, графических и сетевых адаптеров, устройств ввода, принтеров, символьных и USB-устройств

### Документация

- Контекстно-ориентированная справка и тысячи страниц онлайновой документации

### Системные требования

- Pentium IV 2ГГц и выше
- не менее 512 Мб ОЗУ
- не менее 3 Гб дискового пространства
- разрешение экрана не менее 1024x768

## **О компании QNX Software Systems**

QNX Software Systems является одной из дочерних компаний Research in Motion занимает лидирующую позицию в области технологий операционных систем реального времени. Такие ведущие компании мира, как Cisco, DaimlerChrysler, General Electric, Lockheed Martin и Logitech, применяют технологии QNX для создания комплексов сетевой маршрутизации, медицинского оборудования, транспортных телематических систем, систем безопасности и обороны, промышленной робототехники и других приложений критического назначения и жизнеобеспечения. Компания QNX Software Systems была основана в 1980 году. Её штаб-квартира расположена в Оттаве (Канада), а продукция компании распространяется более чем в 100 странах мира.



**SWD Software Ltd.**  
Платиновый дистрибутор компании QNX Software Systems  
196210, Санкт-Петербург, ул. Внуковская, 2,  
БЦ "Пулково Скай", офис С-507.  
Тел: (812) 611-07-51, 611-07-59, факс: (812) 611-07-58  
[info@swd.ru](mailto:info@swd.ru) • [www.swd.ru](http://www.swd.ru)



## **QNX® Momentics®**

### КОМПЛЕКТ РАЗРАБОТЧИКА



• [English](#)

• [Contact](#)

• [myQNX](#)

- Products & Services

**[PRODUCTS & SERVICES](#)**

[Acoustics Middleware](#)

[Hypervisor](#)

[Operating Systems](#)

[QNX Advanced Driver Assistance Systems Platform](#)

[QNX CAR Platform](#)

[QNX Loosely Coupled Lock Step](#)

[QNX Momentics Tools Suite](#)

[QNX Platform for Instrument Clusters](#)

[QNX SDK for Apps and Media](#)

[QNX SDK for Bluetooth Connectivity](#)

[Wireless Framework](#)

**[SERVICES](#)**

[Professional Services](#)

[Training & Education](#)

[Consulting & Programs](#)

[Automotive Services](#)

- Markets

**[MARKETS](#)**

[Automotive](#)

[Industrial](#)

[Medical](#)

[Security & Defense](#)

[Rail Safety](#)

- Developers

**[DEVELOPERS](#)**

[QNX Software Center](#)

[Developer community](#)

[Product documentation](#)

[BSP directory](#)

[Reference design + demos](#)

**[DOWNLOADS](#)**

[QNX Software Development Platform 7.0\\*](#)

[QNX CAR 2.1](#)

[SDK for Apps and Media 1.1](#)

[QNX Wireless Framework 1.0](#)

[QNX Download Centre](#)

[Free 30 day Evaluation](#)

- Partners

**[PARTNERS](#)**

[Partner Directory](#)

- Support

**[SUPPORT](#)**

[Overview](#)

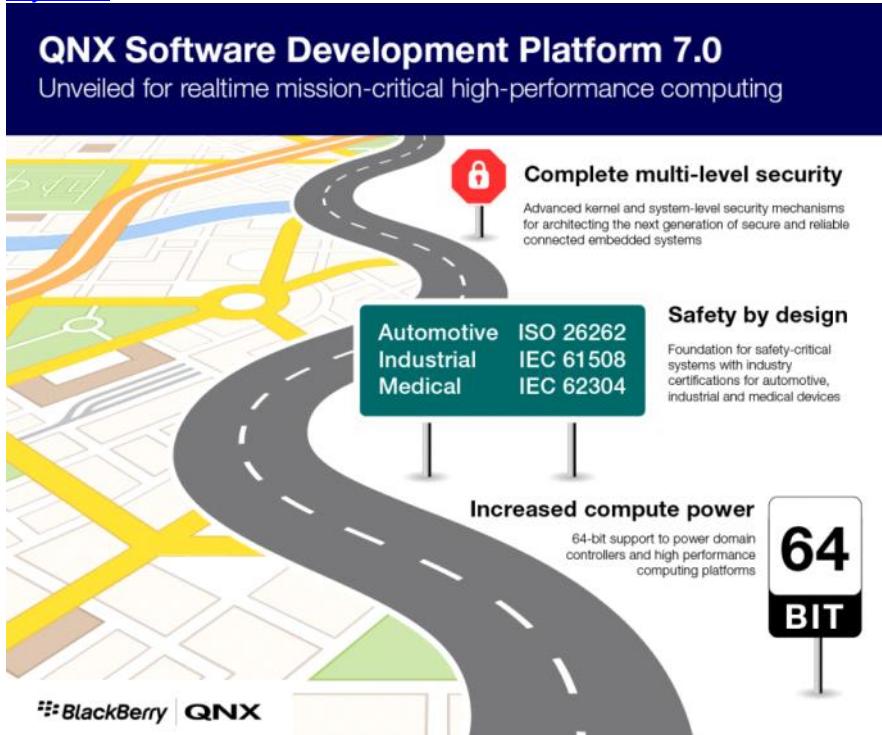
[Support options](#)

[Support portal](#)

[Knowledge base](#)

[Product documentation](#)

- [Home](#)
- [Contact](#)
- [myQNX](#)



## Overview

QNX Software Development Platform (SDP 7.0) includes the next generation 64-bit QNX® Neutrino® RTOS and the award-winning QNX® Momentics® Tool Suite. It provides a comprehensive, multi-level, policy-driven security model incorporating best-in-class security technologies from BlackBerry, which help guard against system malfunctions, malware and cyber security breaches. Building on existing certifications including ISO 26262, IEC 61508 and IEC 62304, QNX SDP 7.0 also brings a proven safety pedigree. Various features, including: microkernel architecture, file encryption, adaptive time partitioning, and high availability framework, make QNX SDP 7.0 the most advanced and secure embedded OS developed for use in all safety and mission critical applications.

## QNX Software Development Platform 7.0 highlights:

### Complete multi-level security

- **Configurable:** Optimal security levels can be specified by the system-wide policy-driven security model.
- **Runtime protection:** Highly secure systems can leverage security features such as address space layout randomization, secure boot, chain of trust establishment, integrity measurement, mandatory access control, path space control, rootless execution and anomaly detection.
- **Secure software delivery:** Software supply chain integrity management enabled through digitally signed package delivery and software update alerts.

### Safety by design

- **Safety certified pedigree:** ISO 26262 ASIL D for automotive, IEC 61508 SIL3 for industrial and IEC 62304 for medical.
- **Component isolation:** Separation between user applications, system services and device drivers enabled by the QNX Neutrino RTOS' proven microkernel architecture.
- **Selectable scheduling algorithms:** Priority based, sporadic and time-triggered deterministic algorithms for system flexibility.
- **Guaranteed CPU allocation:** Minimum CPU allocation at the thread or process

level.

## Increased compute power

- **High performance:** Full 64-bit and 32-bit support for ARMv7, ARMv8 and Intel x86 Architecture.
- **Complete hardware optimization:** AMD, Intel, Nvidia, NXP/Freescale, Qualcomm, Renesas, Samsung, Texas Instruments and Xilinx SoCs, Comprehensive board support packages.
- **GPU integration:** ARM, Imagination, Intel, Nvidia, Qualcomm and Vivante.

## Award winning development tools

- **Development lifecycle tools:** QNX Momentics Tool Suite provides time-saving tools for the entire development cycle in a single, unified standards-based environment.
- **Standards-based:** Eclipse-based Integrated Development Environment (IDE), GCC command line tools, Valgrind integration and C++14 support
- **Debug and optimize:** Complete system-level debugging and optimization through best-in-class application profiler, code coverage, memory and runtime analysis tools.

BlackBerry QNX Launches Software Platform for Autonomous Drive and Connected Cars

### [View Press Release](#)

QNX Neutrino® Realtime Operating System

### [Download Product Brief](#)

QNX Momentics Tool Suite

### [Download Product Brief](#)

Download QNX SDP 7.0

### [Learn more](#)

QNX SDP 7.0 BSPs

### [View BSPs](#)

Ready to purchase?

### [Contact sales team](#)

- Company
  - [About QNX](#)
  - [Autonomous Vehicle Innovation Centre](#)
  - [Careers](#)
  - [Certifications](#)
  - [Events](#)
  - [News Releases](#)
  - [QNX-in-Education](#)
  - [Webinars](#)
- Community
  - [Developer community](#)
  - [Product documentation](#)
  - [Products & Updates](#)
  - [Forums](#)
  - [Newsletter archive](#)
  - [Social @ QNX](#)
- Try QNX Now
  - [QNX Evaluation Kits](#)
  - [30-day free evaluation](#)
  - [Reference designs & demos](#)
- Collateral
  - [QNX Download Center](#)
  - [Benchmarks](#)
  - [Company briefs](#)
  - [Market briefs](#)
  - [Product briefs](#)
  - [Video library](#)
  - [Whitepapers](#)
- Headquarters
  - 1001 Farrar Road
  - Ottawa, ON K2K 0B3
  - Canada

Phone: +1-613-591-0931  
Email: [info@qnx.com](mailto:info@qnx.com)

[Worldwide locations](#)

Copyright @ 2017 QNX Software Systems Limited, a subsidiary of BlackBerry.

- 
- 
- 
- 
- 

[Accessibility](#)  
[Privacy](#)  
[Licensing](#)  
[Legal](#)  
[Sitemap](#)

Источник <<http://blackberry.qnx.com/en/sdp7>>