

QNX® Software Development Platform 6.6: Описание изменений

Дата редакции: 7 марта 2014

QNX Software Development Platform (SDP) – среда кроссплатформенной разработки и отладки, включающая IDE и командно-строчные инструменты, предназначенные для построения двоичных образов ОСПВ QNX Neutrino 6.6 и программ для целевых платформ ARM v7 и x86.

Комплект программных инструментов QNX SDP может быть установлен в следующих ОС:

- Microsoft Windows 8 Professional 64-bit, Windows 7 Professional 64-bit или XP Professional SP3
- Linux Red Hat Enterprise 6 64-bit или Ubuntu Workstation LTS 12.04 32- и 64-bit

Функционирование ОСПВ QNX Neutrino проверено на следующих ARM-платформах:

- Freescale i.MX6x SABRE AI/ARD
- Freescale i.MX6Q SABRE Lite
- TI Jacinto 4/5 EVM
- TI Jacinto 5 ECO EVM
- TI AM335x Beaglebone/Beaglebone Black
- TI AM335x EVM
- TI OMAP4430/OMAP4460
- TI OMAP5432 uEVM ES2.0

ОСПВ QNX Neutrino поддерживает как обычные PC, так и следующие чипсеты x86:

- Intel BloomField Hills (также известный как Bay Trail - Atom E3800)
- Intel CedarView (также известный как Cedar Trail - Atom N2x00, D2700, D2550)
- Intel Haswell (Q87, QM87)
- Intel Ivybridge (Q77, QM77)
- Intel Pineview (Atom N4xx, D510)
- Intel SandyBridge (Q67, QM67)
- Intel Tunnel Creek (Atom Z6xx)

Внимание!

Если на компьютере установлена предварительная версия этого ПО, то ее необходимо удалить перед установкой официальной версии.

Перед запуском QNX Neutrino на целевой платформе необходимо убедиться, что в BIOS отключен параметр Plug and Play OS.

В тексте этого документа будут встречаться номера ошибок и изменений. При взаимодействии с отделом технической поддержки QSS, указывайте эти номера для определения проблемы. Вы также обнаружите, что эти номера удобны для отслеживания исправлений известных ошибок.

Для доступа к последней версии документа, необходимо зайти на сайт www.qnx.com, авторизоваться с помощью учетной записи myQNX и перейти в раздел Download. Там же вы найдете ISO образ среды исполнения QNX Neutrino, который можно установить в виртуальную машину VMware Workstation, VMware Player, VMware Fusion или Oracle VirtualBox.

Содержание

Каковы новые функции и возможности ОСРВ QNX Neutrino?

- Ядро
- Сетевая конфигурация
- Библиотеки и заголовочные файлы
- Драйверы
- Устройства ввода/вывода
- Безопасность

Каковы новые возможности комплекта разработчика QNX Momentics?

- Компилятор, инструменты и утилиты
- Интегрированная среда разработки IDE
- Отладочная информация по двоичным файлам

Устаревший функционал

Экспериментальное ПО и функционал

Каковы новые функции и возможности ОСРВ QNX Neutrino?

Основные изменения в данной версии включают:

Screen

Новая подсистема Screen Graphics замещает microGUI, включая PhAB, Phindows, и QNX Neutrino Advanced Graphics. Обычно называемая просто Screen, подсистема Screen Graphics предоставляет возможность внеэкранного рендеринга и способна формировать изображение, используя различные технологии отрисовки, включая HTML5, Elektrobit GUIDE, Crank Storyboard, Qt, и функции операционной системы (например, OpenGL ES). Более подробная информация содержится в руководстве «Screen Graphics Subsystem Developer's Guide».

Списки управления доступом (ACLs)

Некоторые файловые системы, такие как Power-Safe (fs-qnx6.so) расширяют доступ к файлам используя списки управления доступом (ACLs), соответствующие стандартам IEEE POSIX 1003.1e и 1003.2c.

Библиотека C включает в себя различные `acl_*`() функции, взаимодействующие с ACLs; в командной строке можно использовать утилиты `getfacl` и `setfacl` для определения или получения списка управления доступом к тому или иному файлу/файлам. Более подробная информация содержится в руководствах «QNX Neutrino User's Guide», «QNX Neutrino Programmer's Guide», «C Library Reference» и «Utilities Reference».

Возможности менеджера процессов

Новые возможности менеджера процессов позволяют запускать приложения, которым требуется некоторые права суперпользователя, от имени обычного пользователя, предоставив им только необходимую часть привилегий суперпользователя. Более подробная информация содержится в главе "Process privileges" руководства «QNX Neutrino Programmer's Guide», в главе "Security" руководства «Writing a Resource Manager», также в описании функции `procmgr_ability()` в руководстве «C Library Reference» и утилиты `ability` в руководстве «Utilities Reference»

Режим «Tickless»

Теперь ядро может функционировать в так называемом режиме «tickless», позволяющем сокращать энергопотребление. Системные часы по-прежнему активны, и все работает в штатном режиме до тех пор, пока система не переходит в состояние ожидания. Только когда система полностью простаивает, ядро «отключает» системные часы. Фактически они замедляются таким образом, что очередное прерывание от системного таймера происходит сразу после того, как должен произойти ближайшее срабатывание активного таймера, поэтому таймер сработает немедленно.

Если таймеру не требуется высокая точность, для него можно определить значение толерантности, что даст ядру возможность быстрее перейти в режим «Tickless». Более подробная информация содержится в главе "Clocks, timers, and power management" руководства «QNX Neutrino Programmer's Guide», и в описаниях функций `TimerSettime()`, `TimerTimeout()`, `timer_settime()`, `timer_timeout()` и `procmgr_timer_tolerance()` в руководстве «C Library Reference».

Отложенные прерывания (Lazy interrupts)

Для того чтобы ядро потребляло меньше энергии, можно ввести функцию отложенного прерывания (`lazy interrupt`) и назначить для нее подходящее время задержки. Перед тем, как перевести центральный процессор в режим сна, ядро проверяет значения всех задержек прерываний и контролирует, способно ли оно обеспечить обработку другого прерывания (например, от системного таймера) раньше истечения времени задержки. Если ядро вычисляет, что другое прерывание может быть обработано раньше, оно маскирует отложенное прерывание перед переходом в режим сна. Когда процессор получает прерывание, все отложенные прерывания демаскируются. Более подробная информация содержится в главе «Interrupts and power management» руководства «QNX Neutrino Programmer's Guide».

DHCP

ОСРВ QNX 6.6 поддерживает протокол DHCP версии 6. Более подробная информация содержится в описаниях утилит `dhclient`, `dhclient-script`, `dhclient.conf` и `dhclient.leases` в руководстве «Utilities Reference», и в описании функций `dhcpcd*`() в руководстве «C Library Reference».

Возможности файловой системы Power-Safe (fs-qnx6.so)

- Теперь появилась возможность полного или частичного шифрования контента файловой системы Power-Safe путем разделения его на шифровальные домены. Более подробная информация содержится в пункте "Encryption" главы «Filesystems» руководства «System Architecture», а также в описании утилиты fsencrypt в руководстве «Utilities Reference».
- Файловая система Power-Safe поддерживает команду TRIM, которая, при использовании NAND-флэш, позволяет указывать файловой системе, что определенные секторы сбояли и могут быть исключены.

Доверенные файловые системы

Если любой процесс пытается задать для какого-либо региона памяти флаг PROT_EXEC, то файлы, отображаемые в этот регион должны быть либо доверенными, либо получены из доверенной файловой системы. Более подробная информация содержится в описании утилиты pathtrust в руководстве «Utilities Reference» и в описании функции fdistrusted() в руководстве «C Library Reference».

Библиотеки Dinkum

Теперь в состав операционной системы входит библиотека Dinkum C++11 в виде файла libcxx.so.5. Для обратной совместимости поставляются следующие библиотеки:

- libcxx.so.4 — библиотека C++
- libecxx.so.4 — библиотека Embedded C++

Образы файловой системы

Теперь можно создавать образы файловых систем Power-Safe (fs-qnx6.so) и FAT; см. описания утилит mkqnx6fsimg и mkfatfsimg в руководстве «Utilities Reference».

slogger2

Системный журнал, устраняющий различные ограничения системы slogger. Подробнее см. описание утилит slogger2 и slog2info в руководстве «Utilities Reference», и описание функций slog2*() в руководстве «C Library Reference».

QNX SDK for Apps and Media

Среда разработки, поставляемая как виде отдельного продукта вместо «QNX Aviage Multimedia Suite». Предоставляет поддержку мультимедиа, фреймворк управления приложениями, браузер "chromeless", и встраиваемый браузер.

Для получения информации об остальных изменениях, см. следующие разделы:

- [Ядро](#)
- [Сетевая конфигурация](#)
- [Библиотеки и заголовочные файлы](#)
- [Драйверы](#)
- [Устройства ввода/вывода](#)
- [Безопасность](#)

Новые возможности: Ядро

- Поддержка `-ae` в `procnto` на процессорах ARMv7 изменена так, что бы позволять осуществлять доступ к данным в аппаратуре без выравнивания. Это значит, что на этих целевых системах функция `ThreadCtl(_NTO_TCTL_ALIGN_FAULT, ...)` больше не изменяет поведение потока, что раньше требовалось во избежание ошибки выравнивания. (ref# 77775, 77873, J169734)
- По умолчанию, когда память освобождается для последующего использования, ее содержимое остается нетронутым до тех пор, пока эта память не будет выделена другому процессу. К тому моменту, когда память становится доступна другому процессу, она зануляется. Новая опция `-mc` позволяет указывать ядру, чтобы оно зануляло память при освобождении, не дожидаясь того, когда она будет выделена в следующий раз. Например:

Примечание: при реальных файлах построения нельзя использовать обратный слэш (`\`) для разделения длинной строки на фрагменты. Это сделано в примере для удобства чтения.

```
[image=0x8c000000]
```

```
[virtual=shle,srec +compress] .boot = {  
  
    ../../startup-msrp1 -v -Nrp1    -Dscif..38400  
  
    PATH=/proc/boot:/bin:/sbin:/usr/bin:/usr/sbin: \  
  
    LD_LIBRARY_PATH=/proc/boot:/usr/lib:/usr/nto/shle/lib:/usr/nto/shle/lib/dll:\br/>  
    /usr/nto/shle/usr/lib:/tmp \  
  
    procnto -v -mc  
  
}
```

Чтобы явно задать поведение по умолчанию можно использовать опцию `-m~c`. (Ref# 77509, 77351, J169691)

- При создании элементов в каталогах `/proc/pid/` вместо маски создания файлов `0022` используется значение `0066`.
Примечание: файлы в каталогах `/proc/pid/` успешно открываются в режиме просмотра, даже если права доступа должны возбуждать ошибку `EACCESS`. Вместо этого ядро сделает в ОСB пометку, что к файлу допустимо применять только функцию `devctl()`. Это предотвратит сканирование вычислительной памяти непривилегированными процессами, однако позволит запущенной от имени обычного пользователя утилите `pidin` отображать некоторую полезную информацию. (Ref# 78922, 78924, 169797)

Новые возможности: сетевая конфигурация

lsm-autoip.so

По умолчанию протокол AutoIP больше не выполняет маршрутизацию для того, направлять линк-локальную сеть на использование маршрутизируемых IP-адресов интерфейса, и не выполняет маршрутизацию на линк-локальные IP-адреса. Если необходимы старое поведение маршрутизации, используйте опцию *old*. Такое использование маршрутизации сопровождалось определенными проблемами в зависимости от конфигурации пиринговых точек, а также при наличии нескольких интерфейсов. Теперь опция *force* не действует, если не задать также опцию *old*. (Ref# 77436, 77649, J169717)

pfctl

Чтобы обеспечить непрерывную переадресацию, нужно установить значение переменной `net.inet.ip.forwarding` и/или `net.inet6.ip6.forwarding` на единицу в файлах типа `/etc/sysctl.conf` и затем запустить команду `sysctl`, используя `f`-опцию в ваших системных файлах типа `/etc/rc.d/rc.local`. (Ref# 78736, J382535)

Разница между реализациями DHCPv4 и v6

Существует несколько ключевых различий между старой и новой реализациями DHCP, о которых нужно знать:

- Для `dhclient` добавлено 2 новые опции командной строки:

-m

Запись конфигурации обнаружителя в память (используя `confstr`) вместо `/etc/resolv.conf`.

--no-resolve

Не устанавливать конфигурацию обнаружителя.

- `dhclient`: Вся конфигурация локального узла обеспечиваются командным сценарием `dhclient`, то есть существует возможность модифицировать и адаптировать его по мере необходимости, что обеспечивает высокую гибкость. Базовая версия сценария размещена в каталоге `/sbin`.
- Вышеупомянутые опции командной строки `-m` и `--no-resolve` реализуются путем простой передачи сценарию переменных окружения `RESOLV_TO_MEMORY` и `MODIFY_RESOLV` соответственно, и базовый сценарий их обрабатывает.
- Базовый сценарий `dhclient` переписывает конфигурацию локального обнаружителя, не объединяя ее. Это преимущество: устаревшая конфигурация обнаружителя не хранится, но у этого подхода есть свои недостатки (см. ниже).
- Устаревший `dhcpcd` обслуживал один интерфейс, новый `dhclient` может управлять несколькими интерфейсами в рамках одного процесса (но конечные автоматы будут разные).
- Новые `client`, `relay` и `server` поддерживают IPv6 (старые не поддерживали).
- Новые `client`, `relay` и `server` функционировать либо в режиме IPv4 (задано по умолчанию, опция `-4` командной строки), либо IPv6 (соответственно, `-6`), но не в обоих сразу.

- При одновременном запуске двух и более копий клиента `dhclient`, есть потенциальная возможность гонки общих параметров конфигурации, которые они могут принимать одновременно, и это должно учитываться сценарием `dhclient`. Одним из примеров таких параметров конфигурации является конфигурация обнаружителя, в данном случае проблема решается запуском одной копии клиента с опцией `--no-resolve` или модификация базового сценария `dhclient`.
- Возможность ARP-зондирования, которая была реализована в старой версии `dhcpd`, не поддерживается в новом `dhclient`; такой функционал должен быть в сценарии `dhclient`.
- Ошибка: отказы при выполнении операции «Duplicate Address Detection» в протоколе IPv6 не регистрируются `dhclient`. Очевидно, они должны обрабатываться сценарием `dhclient`, однако в настоящее время возвращаемые значения сценария не обрабатываются, поэтому это не функционирует.
- Мы добавили поддержку DHCPv6 через интерфейсы PPP (и мы надеемся, что ISC будет официально поддерживать данный патч).
- `dhclient` не выполняет очистку при принудительном завершении. Вместо этого используйте опции `-r` и `-x`.
- Для обычных интерфейсов отсчет порядковых номеров IAIDs основан на MAC-адресах, но начинается с нуля для интерфейсов PPP.
- Протокол DHCPv6 использует уникальные идентификаторы DHCP (DUIDs), которые уникальны для целых узлов, а не для отдельных интерфейсов, как можно подумать. DUID – скрытый идентификатор для однорангового узла сети и узел для проверки соответствия может использовать только его (==).
Если DUID находится в лизинговой базе, он будет извлекаться оттуда перед попыткой создания интерфейса. Его нельзя сформировать из интерфейсов PPP, вместо этого наш патч будет пытаться использовать другие интерфейсы (если они заданы в командной строке). DUID также может быть установлен вручную, как это описано в документации в статье `dhclient.conf`.
- Протокол DHCPv4 не рассчитан для работы на PPP, при попытке такого запуска выдается ошибка следующего вида:

```
Unsupported device type 23 for "ppp0"
```

У утилит DHCP имеются следующие зависимости:

dhcpd

Зависит от следующих библиотек/двоичных файлов:

- `libcrypto.so`
- `libsocket.so`
- `libdhcpctl.so` (как часть `io-pkt/services/dhcp`)
- `io-pkt-v4`, `io-pkt-v4-hc`, или `io-pkt-v6-hc` (в зависимости от того, используется IPv4 или IPv6)

Файлы конфигурации:

- `/etc/dhcpd6.conf` (обязательно требуемый файл конфигурации DHCPv6; при запуске можно переопределить его имя `-cf config_file`)

- /etc/dhcpd.conf для IPv4 (DHCPv4)
- /var/db/dhcpd6.leases (обязательно требуемая база данных и идентификатор сервера, должна быть доступна по чтению и записи; при запуске можно заменить его на -lf leases_file)
- /var/db/dhcpd.leases для IPv4

В общем случае необходимо создать пустой файл лизинговой базы данных.

dhclient

Зависит от следующих библиотек/двоичных файлов:

- libcrypto.so
- libsocket.so
- io-pkt-v4, io-pkt-v4-hc, или io-pkt-v6-hc (в зависимости от того, используете ли вы IPv4 или IPv6)
- /sbin/dhclient-script (необходим обязательно; при запуске можно заменить его на -sf script-file)

Файлы конфигурации:

- /var/db/dhclient6.leases (не обязательно; если файл отсутствует, то при каждом запуске будет создаваться новый идентификатор клиента; можно заменить его при запуске)
- /etc/dhclient6.conf (не обязательно; если файл отсутствует, применяются настройки по умолчанию; можно заменить его при запуске)
- /etc/dhclient.conf для IPv4
- dhclient6.leases
- dhclient.leases для IPv4

Скрипт dhclient

Зависит от следующих библиотек/двоичных файлов:

- ifconfig (которому нужен io-pkt-* и libsocket.so)
- sh
- route
- hostname
- getconf
- setconf
- cat
- mv
- Если присутствуют файлы /etc/dhclient-enter-hooks и /or /etc/dhclient-exit-hooks, сценарий dhclient запустит и их.

Конфигурационные файлы:

- /etc/resolv.conf (не обязательно, однако требуется обновление DNS-сервера; файл должен быть доступен по чтению и записи)

dhcrelay

Зависит от следующих библиотек:

- libcrypto.so
- libsocket.so
- io-pkt-v4, io-pkt-v4-hc, или io-pkt-v6-hc (в зависимости от того, используете ли вы IPv4 или IPv6)

Все настройки выполняются при запуске в командной строке.

Новые возможности: библиотеки и заголовочные файлы

ClientInfoExtFree()

Очищает структуры, используемые для получения расширенной информации о клиентском соединении (Ref# 77731, 79485, J169855)

ConnectClientInfoExt()

Хранит расширенную информацию о клиентском соединении (Ref# 77731, 79485, J169855)

InterruptCharacteristic(), InterruptCharacteristic_r()

Устанавливает характеристики прерывания (Ref# 75524, 78945)

inotify_add_watch()

Добавляет или обновляет часы для событий в файловых системах (Ref# 104112, 114210)

inotify_event

Структура, описывающая события в файловых системах (Ref# 104112, 114210)

inotify_init()

Инициализирует работу оповещающей системы индексных дескрипторов (Ref# 104112, 114210)

inotify_rm_watch()

Удаляет часы, сопряженные с данным дескриптором часов (Ref# 104112, 114210)

iofunc_client_info_ext()

Хранит информацию о клиентском соединении. Это расширенная версия более не поддерживаемой функции `iofunc_client_info()`, которая может поддерживать больше дополнительных групп NGROUPS_MAX. (Ref# 77731, 79485, J169855)

iofunc_client_info_ext_free()

Структура, предназначенная для получения информации о клиентском соединении (Ref# 77731, 79485, J169855)

iofunc_devctl_verify()

Выполняет проверку, общую для всех обработчиков сообщений _IO_DEVCTL (Ref# 81734, 263259, J194933)

memset_s()

Безопасно устанавливает заданное значение памяти (Ref# 116600, 163197, J173676, J181738)

OMAPI

Программный интерфейс управления объектами (Ref# 101106, J172134)

procmgr_ability()

Управляет способностью процесса выполнять определенные операции (Ref# 90654, 90953, J171128)

procmgr_timer_tolerance()

Устанавливает временные рамки для таймера по умолчанию для определенного процесса (Ref# 290293, 294234, J336228)

set_lowest_fd()

Определяет минимально приемлемый дескриптор файла (Ref# 122362, 170742, J174315, J183120)

slog2*()

Интерфейсы APIs работают с подпрограммой slogger2, которая является системой протоколирования, снимающей различные ограничения с унаследованной slogger.

strlen()

Высчитывает длину строки до максимального числа байт (Ref# 98975, 242696, J171922, J192427)

sysmgr_runstate_burst()

Запрашивает активацию динамически автономных ЦП (Ref# 172224, 173649)

vslog2f()

Записывает строку в формате printf-style в буфере slog2 (функция varargs) (Ref# 155346, 175311, J180160, J333708)

Среди прочих изменений можно отметить следующие:

ChannelCreate()

Новый флаг `_NTO_CHF_PRIVATE`, с помощью которого можно пометить канал как приватный. Если сторонний процесс вызывает функцию `ConnectAttach()` для соединения с приватным каналом, то она завершается с кодом ошибки `EPERM`. (Ref# 76018, 78892)

errno

Ошибка `EALREADY` теперь называется `EALREADY_NEW`, таким образом, `EALREADY` и `EBUSY` имеют разные значения, как это задано в POSIX. Для более подробной информации см. "Изменения в ошибке `EALREADY`" в описании `errno` руководства «C Library Reference».

etfs_devio

Структура `etfs_devio` (заданная в `<fs/etfs.h>`) теперь расширена, ее размер увеличен за счет размера интервала.

MsgDeliverEvent()

Для удобства можно присвоить переменной `rcvid` значение 0, чтобы пометить событие в активном потоке. (Ref# 76137, 77519, J169706)

munmap_flags()

Эта функция поддерживает следующие флаги:

- `UNMAP_CLEAN` — при возможности очищает область памяти при отключении от адресного пространства.
- `UNMAP_NOCLEAN` — не очищает память при отключении от адресного пространства, даже если это установлено по умолчанию.

Эти флаги меняют поведение по умолчанию, определенное опциями `-mcs` или `-m~c` в менеджере процессов `procnto`. (Ref# 77509, 77351, J169691)

shm_ctl(), shm_ctl_special()

Эти функции устанавливают для `errno` более понятные значения. Для более подробной информации см. описания функций в руководстве «C Library Reference». (Ref# 26440, J166020, J170273)

ThreadCtl()

- На целевых платформах ARMv7, команда `_NTO_TCTL_ALIGN_FAULT` теперь игнорируется, невозможно управлять поведением, связанным с ошибкой выравнивания. Можно установить поведение «global», задействовав опции `-ad` или `-ae` в менеджере процессов `procnto*`. (Ref# 77775, 77873, J169734)
- На целевых платформах ARMv7, команда `_NTO_TCTL_IO` больше не переводит потоки в (привилегированный) режим выполнения. Драйверы, для запуска которых необходим этот режим, могут вместо этого использовать новую команду `_NTO_TCTL_IO_PRIV`. Эти изменения сделаны для поддержания двоичной совместимости с устаревшими драйверами.

Более подробная информация содержится в приложении «What's New» руководства «C Library Reference».

Новые возможности: драйверы

Звуковые драйверы (deva-*)

deva-ctrl-via8233.so

Звуковой драйвер для аудио-контроллера VIA 8233

Символьные драйверы (devc-*)

devc-serusb_dcd

Драйвер для адаптеров USB-to-serial

Сетевой драйвер (devn-*, devnp-*)

Среди новых драйверов можно отметить следующие:

devn-smsc9500.so

Драйвер для USB-Ethernet конверторов SMSC9500

devnp-asix.so

Драйвер для USB-Ethernet конверторов ASIX AX88172, AX88172A, AX88178, AX88772, AX88772A, AX88772B; замещает devn-asix.so и devnp-axe.so

devnp-ecmplus.so

Драйвер для USB Ethernet устройств класса CDC ECM/RMNET

devnp-ixgbe.so

Драйвер для контроллеров Intel 10 Gigabit Ethernet

devnp-speedo.so

Драйвер Fast Ethernet для адаптеров Intel 82557, 82558, и 82559; замещает **devn-speedo.so**

devnp-usbdnet.so

Драйвер для USB-устройств класса USBDNET (USB Device Network Driver)

Теперь драйверы io-pkt-* подключаются к libnetdrv, а не к libdrv. (Ref# 328930, J352515)

USB-драйверы (devu-*)

devu-umass_client-block

Драйвер для съемных USB-накопителей

devu-xhci.so

Драйвер интерфейса XHCI для USB 3.0. Память драйвера devu-xhci.so на данный момент не поддерживает высокие скорости.

Новые возможности: устройства ввода/вывода

io-blk.so

Чтобы уменьшить размер io-blk.so, мы переместили набор символов японского и традиционного и стандартного китайского языков (кодированные страницы 932, 936, и 950) в отдельную библиотеку, названную charset.so, которая загружается только при записи тома FAT, CD- или DVD-дисков, форматируемых в одной из этих языковых настроек. Если системе не требуются наборы этих символов, можно удалить charset.so из образа системы.

Новые возможности: безопасность

- Утилита pprojectl больше не запускается с параметром setuid root. (Ref# 275244, J475184)

Каковы новые возможности комплекта разработчика QNX Momentics?

Комплект разработчика QNX Momentics 6.6 содержит инструменты следующих версий:

- GCC 4.7, включая поддержку расширений Intel Advanced Vector (AVX)
- GDB 7.5
- Binutils 2.23
- Python 2.7.5, как инструмент для хоста

Для более подробной информации о новых возможностях GCC 4.7 см.:

- <http://gcc.gnu.org/gcc-4.7/changes.html>
- http://gcc.gnu.org/gcc-4.7/porting_to.html

Для получения подробной информации о других изменениях смотрите следующие разделы:

- [Компилятор, инструменты и утилиты](#)
- [Интегрированная среда разработки IDE](#)
- [Отладочная информация по двоичным файлам](#)

Новые возможности: компилятор, инструменты и утилиты

Inflator

Добавлена опция `-t`, с помощью которой можно определить максимальное число рабочих потоков. 4 по умолчанию. (Ref# 77417, 78036, J169746)

ls

Если запустить `ls -l` на файле, который не является исполняемым, и у которого выставлен флаг `setgid`, то права доступа будут содержать `S` вместо `L` в соответствии с POSIX. (Ref# 103631, J381132)

mkefs, mketfs, mkifs

- Теперь нельзя определить задать атрибут `type=link` и устанавливать только один путь. (Ref# J587290)
- Эти утилиты теперь поддерживают строки ввода произвольной длины. (Ссылки 200543, J400635)

QCC, qcc

Новая опция `-save-temps` сохраняет промежуточные файлы, созданные в процессе компиляции. (Ref# 54630, 77819, J143234, J381127)

slogger

Новые опции:

- `-m` — использует источник времени `CLOCK_MONOTONIC` вместо заданного по умолчанию `CLOCK_REALTIME`.
- `-u event_id` — для всех получаемых сообщений генерирует пользовательские события строкового типа с заданными идентификатором события. (Ref# 77106, 77332, J169690)

Более подробная информация содержится в приложении «What's New» руководства «Utilities Reference».

Новые возможности: интегрированная среда разработки (IDE)

Комплект разработчика QNX Momentics 6.6 содержит версию IDE 5.0.1. со следующими новыми возможностями:

- Eclipse 4.2.1
- CDT 8.2.0

Веб-сайт с обновлениями, доступ к которым выполняется через меню «Help > Check for Updates», доступен по адресу <https://www.qnx.com/account/updates/ide/4.7/>

Эта версия IDE содержит следующие исправления и новые возможности:

- Текст в разделе «About» содержит лицензионную информацию. Зайдите в раздел «Help» > «About QNX Momentics IDE», затем «Details», «Configuration tab» и «License Information». (Ref# 98011, J467224)
- Команда «make clean», заданная в `makefile`-ах проектов «System Builder», при выборе опции «Clean Project» теперь вызывается корректно. (Ref# 45628, J329085)
- Проекты типа «Make C++», компилируемые как C++ GNU с исключениями больше при запуске не завершаются с ошибкой. (Ref# 259430, J400928)

Для более подробной информации об изменениях в IDE, см. приложение «What's New in the IDE» руководства пользователя «IDE User's Guide».

Отладочная информация по двоичным файлам

Двоичные файлы QNX SDP снабжены отладочной информацией (`-g`) и файлами распределения памяти. За некоторыми исключениями все двоичные файлы снабжены отладочной информацией, но она, как и все данные типа `.ident`, вырезана (`stripped`) и хранится в отдельном файле

binaryName.sym. Эти файлы связаны между собой, поэтому gdb знает, где искать символьные данные.

Теперь более не нужны отдельные отладочные версии для всех двоичных файлов. Есть отдельный tar-файл, содержащий все файлы типа .sym, которые будут обработаны наряду с двоичными файлами. Этот файл можно загрузить, он находится в разделе «Download» нашего веб-сайта, а также в директории debugging_info на DVD-диске QNX SDP.

Что это значит:

- Двоичные целевые файлы теперь стрипованы.
- Двоичные целевые файлы теперь не содержат в себе информацию SRCVERSION (то есть функция -s работать не будет).
- Все двоичные файлы в QNX Neutrino собираются с опцией -g.
- Теперь файлы карт распределения компоновщиком памяти создаются для всех двоичных файлов QNX Neutrino.
- Отладочные символы для двоичного файла some_binary (вместе с информацией SRCVERSION) хранятся в файле some_binary-buildid.sym.
- Двоичный файл связан sym-файлом, поэтому отладчик gdb знает, где искать символьную информацию.

Файлы типа *-buildid.sym являются платформозависимыми tar-файлами. Использовать их очень просто. Предположим, необходимо отладить утилиту ls для x86. Для этого достаточно просто скопировать на целевую систему весь набор отладочных файлов (разумеется, необходимы права суперпользователя):

1. Возьмите файл -x86-debug-date.tgz.
2. cd \$QNX_TARGET
3. tar -zxf path/target-x86-debug-date.tgz

Затем команда ntox86-gdb \$QNX_TARGET/bin/ls автоматически загрузит отладочные символы из файла типа .sym. Поскольку он содержит полную отладочную информацию, то можно указать отладчику gdb на расположение исходных текстов утилиты ls.

Нет необходимости полностью извлекать содержимое tar-файла. По сути, достаточно того, чтобы файл типа .sym находился в том же каталоге, что и двоичный файл. Так что достаточно скопировать только файлы \$QNX_TARGET/x86/bin/ls и the x86/bin/ls-*.sym (из отладочного tar-файла) в текущий каталог и запустить там отладчик gdb.

Чтобы получить список файлов исходного текста, использованных для построения двоичного файла (например, чтобы выяснить связанные с ним лицензионные соглашения), вместо двоичного файла используется .sym-файл. Если использовать приведенный выше пример, для того, чтобы получить список файлов с исходными текстами, использованными для построения утилиты ls, необходимо выполнить команду:

```
use -s $QNX_TARGET/x86/bin/ls-*.sym
```

Устаревший функционал

Более не поддерживается следующий функционал:

- Целевые платформы: ARM9, ARM11, MIPS, PowerPC или SH4
- asynchronous messaging (функции *asynmsg_**())
- Photon microGUI, включая PhAB, Phindows, и QNX Neutrino Advanced Graphics—заменено на Screen Graphics Subsystem
- QNX Aviage Multimedia Suite
- библиотеки Dinkum C99, C++, и Embedded C++—заменено на библиотеки Dinkum C++11
- cvs
- devn-asix.so — заменено на devnp-asix.so
- devn-speedo.so — заменено на devnp-speedo.so
- devnp-ath.so
- devnp-axe.so — заменено на devnp-asix.so
- devnp-bcm43xx.so
- devnp-ral.so
- devnp-rum.so
- devnp-ural.so
- dhcprelay — заменено на dhcrelay
- diff3
- diskboot
- enum-usb — заменено на usblauncher (подробнее см. в руководстве «Device Publishers Developer's Guide»)
- finstall
- mksbp
- patch
- qbinaudit
- qde — заменено на run-qde
- qed
- QWinCfg — run-qde задает необходимую среду разработки перед запуском IDE
- setupbsp
- SNMP:
- /etc/acl.conf
- /etc/context.conf
- /etc/mib.txt
- mstrip
- /etc/party.conf
- smic
- snmpbulkwalk
- /etc/snmpd.conf
- snmpd
- snmpgetnext
- snmpget
- snmpnetstat
- snmpset

- snmpstatus
- snmpstest
- snmptranslate
- snmptrapd
- snmptrap
- snmpwalk
- /etc/view.conf
- svn
- sysinfo
- who

Более не поставляется резидентная версия QNX SDP (это значит, что теперь нет возможности использовать саму ОСРВ QNX Neutrino для разработки программ). Также больше не поставляется загрузочный CD для x86.

Поддержка Mudflap была исключена из новейших версий FSF gcc, поэтому будущие версии QNX Neutrino также не будут ее поддерживать.

Экспериментальное ПО и функционал

ВНИМАНИЕ!

Экспериментальное ПО предоставляется нашим клиентам и сообществу в первую очередь для апробирования и оценки возможных нововведений в будущем. Для подробной информации об использовании экспериментального ПО и функционала прочтите лицензионные соглашения Commercial Software License Agreement (CSLA) или Partner Software License Agreement (PSLA) в разделе «Licensing» нашего сайта, <http://www.qnx.com/legal/licensing/>

Экспериментальный функционал в QNX SDP 6.6:

- Асинхронная очередь сообщений (управляется mq)