



Технологии QNX и КПДА в России

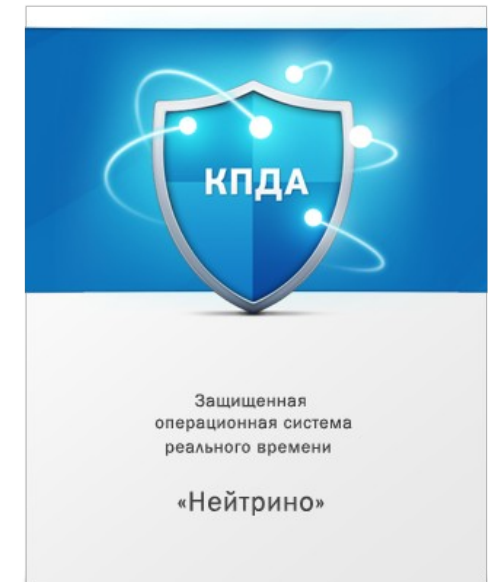
Москва, 25 апреля 2018

Инструментальные средства ЗОСРВ «Нейтрино»: актуальное состояние и тенденции развития

Игорь Рондарев, ООО «СВД Встраиваемые Системы»

ЗОСРВ «Нейтрино» (КПДА.10964-01)

- **Микроядерная** модульная архитектура
- Возможность использования в системах т.н. **“жёсткого” реального времени**
- Удовлетворяет требованиям к СВТ по **3 классу защиты информации от НСД, 2 уровню контроля отсутствия НДВ**, что позволяет использовать изделие КПДА.10964-01 в АС класса защищённости **до 1Б включительно**
- Широкий перечень поддерживаемых вычислительных платформ - **как зарубежных, так и отечественных**
- Бинарная и технологическая совместимость с **ОСРВ QNX 6.5.0**



- Построен на базе программного обеспечения **QNX Software Development Platform 6.5.0**
 - Поддерживаемые инструментальные платформы:
 - **Windows**
 - **GNU/Linux**
 - Поддерживаемые целевые платформы
 - **x86, ARMv7, PowerPC, MIPS, ARMv5/v6, SH**
 - Объём:
 - около **2 ГБ**
 - Состав:
 - Инструментальные средства
 - инструменты командной строки (кросс-компилятор, линкер, отладчик и т.д.)
 - графическая среда разработки **QNX Momentics IDE 4.7** + документация
 - **Двоичные файлы среды исполнения** + вспомогательные КОМПОНЕНТЫ

- **Базовый инструментарий (Core Development Tools)**
 - Компилятор и вспомогательное ПО
- IDE (интергрированная среда разработки)
 - QNX Momentics IDE
 - Qt Creator
- **Дополнительные инструменты и утилиты**
 - Инструменты статического анализа
 - Инструменты динамического анализа
 - Инструменты модульного тестирования
 - Средства оптимизации процесса сборки ПО
 - Прочее

- **До 2018 года**
 - Стандартный инструментарий (**QNX SDP**)
 - Обновления с ресурса <https://community.qnx.com>
- **С 2018 года**
 - **Обновления инструментария от компании «СВД ВС»**
 - Примеры инструментальных компонентов:
 - **kpda-gcc-5.5.0-linux64-r233-b27-20180315.tar.gz** - компилятор **gcc 5.5** для 64-разрядных инструментальных систем под управлением Linux
 - **kpda-binutils-2.24-win32-r187-b37-20180315.zip** - компоненты **binutils 2.24** (ассемблер, линковщик и т. д.) для 32- и 64-разрядных систем под управлением ОС Windows:
 - и т.д.
 - Стандартный инструментарий (**QNX SDP**)
 - Сборки с ресурса <https://community.qnx.com>

- Базовый инструментарий (в составе QNX SDP 6.5.0)
 - gcc 4.4.2 (октябрь 2009), gdb 6.8, binutils 2.19
- **Собственные « сборки » («КПДА»)**
 - **Актуальные версии**
 - **gcc 5.5** (октябрь 2017) и **gcc 4.8.3** (июль 2014)
 - Поддержка C++: **libstdc++.so.6.0.21** и **libstdc++.so.6.0.19**
 - **gdb 7.6**
 - **binutils 2.24**
 - Сравнение с **QNX SDP 7.0**
 - **gcc 5.4**
 - Поддержка C++: **libstdc++.so.6.0.21** и **libc++ (LLVM C++ Library)**
 - **gdb 7.10**
 - **binutils 2.25.1**

- **Основные отличия обновлённых версий gcc**
 - Поддержка актуальных стандартов
 - **C++11 (4.8.3), C++14 (5.5), OpenMP 3.1/4.0**
 - Поддержка новых функций и технологий
 - **Address Sanitizer (gcc >= 4.8)**
 - **Link Time Optimization (gcc >= 4.8)**
 - Оптимизация алгоритмов компиляции
 - Исправления архитектурно-специфичных ошибок
 - <https://gcc.gnu.org/releases.html>
- **Рекомендации при переходе со стандартного компилятора (gcc 4.4.2):**
 - на **gcc 4.8**: https://gcc.gnu.org/gcc-4.8/porting_to.html
 - на **gcc 5.5**: https://gcc.gnu.org/gcc-5/porting_to.html

- **Основные преимущества «сборок» «КПДА»**
 - Технологическая независимость
 - Поддержание инструментария в актуальном состоянии
 - Возможность оперативного исправления ошибок и адаптации под различные вычислительные платформы, в т.ч. отечественные
 - Примеры (по состоянию на 2018 год)
 - поддержка архитектуры **SPARCv8 (LEON)**
 - восстановление работоспособности некоторых компонентов для целевых архитектур **PowerPC и ARMv7**
 - адаптация ряда инструментов для поддержки системы **МС-24** (процессор **1892ВМ2Я**)
 - и т.д.
 - Инструментарий и сопутствующие библиотеки проходят автоматизированное тестирование на поддерживаемых целевых платформах

- **Как получить?**

- Доступны по запросу (support@kpda.ru или **Форма обратной связи**) + предполагаются к включению в состав инструментария разработчика ПО для ЗОСРВ «Нейтрино»

- Базовый инструментарий (Core Development Tools)
 - Компилятор и вспомогательное ПО
- **IDE (интергрированная среда разработки)**
 - QNX Momentics IDE
 - Qt Creator
- **Дополнительные инструменты и утилиты**
 - Инструменты статического анализа
 - Инструменты динамического анализа
 - Инструменты модульного тестирования
 - Средства оптимизации процесса сборки ПО
 - Прочее

Развитие QNX Momentics IDE (история версий)

- Впервые была выпущена компанией QSS в 2002 году
- QNX Momentics IDE = платформа Eclipse + модуль CDT + инструменты QNX
- Новые версии и обновления:
 - **Форум «СВД Встраиваемые системы»**
 - <http://community.qnx.com/sf/frs/do/viewSummary/projects.ide/frs>
- **Все** версии QNX Momentics IDE могут использоваться для разработки ПО для ЗОСРВ «Нейтрино»

Версия	Дата выпуска	Версия Eclipse	Версия CDT	Примечания
4.7	июнь 2010	3.5.2 (Galileo)	6.0.2	Базовая (SDP 6.5.0)
4.x	в течение 2010 года	3.6 (Helios)	7.0	Обновления к 4.7
5.0	май 2014	3.7 (Indigo)	8.x	Экспериментальная
5.0.1 SR5	октябрь 2015	4.2.1 (Juno)	8.1.2	GA (General Availability)
6.0 M1 / 7.0 M1	март 2016	4.6 M (Neon)	9.0	Экспериментальная
7.0 Alpha	июль 2016	4.6 (Neon)	9.x	Экспериментальная
7.0 GA	март 2017	4.6.2 (Neon)	9.2	GA (General Availability)
7.0.2	октябрь 2017	4.6.3 (Neon)	9.2.1	Актуальная

Развитие QNX Momentics IDE (4.7 → 7.0)

- **Общие изменения**

- Поддержка 64-разрядных инструментальных ОС
 - Поддерживаемые инструментальные ОС: **GNU/Linux, Windows и MacOS X**
 - Поддержка MacOS X актуальная при использовании QNX SDP 7.0 и выше.
 - Актуальная редакция Java-машины в составе дистрибутива IDE
 - Нет необходимости «подгонять» системную JVM под требования Momentics
- Переработан интерфейс; новые инструменты создания и запуска проектов
- Новые наборы инструментов; перспектива **QNX Analysis**
- Новый интерфейс редактирования **build**-файлов инструмента **System Builder**
 - Импорт файлов построения образов из QNX Momentics IDE 4.7
- Поддержка инструментария **Valgrind**
 - Взаимодействие с целевой машиной + визуализация результатов
- Повышение стабильности, дополнительная функциональность некоторых компонентов
- и т. д.

Развитие QNX Momentics IDE (4.7 → 7.0)

- **Модуль C/C++ Development Toolkit (CDT)**

- Поддержка синтаксиса проектов, построенных на **платформе Qt**
- Поддержка инструментов **модульного тестирования** (в т.ч. с визуализацией результатов)
 - платформы **Qt Test**, **Boost Test** и **Google Test**
- Поддержка **GDB Debugger Services Framework (DSF)**
- Дополнительная функциональность инструментов работы с кодом
 - Расширенное автодополнение, навигация через журналы («логи») компиляции и т.д.
- и т. д. (подробнее: <http://wiki.eclipse.org/CDT/User/NewIn70>, <http://wiki.eclipse.org/CDT/User/NewIn80> и <http://wiki.eclipse.org/CDT/User/NewIn90>)
- FYI: разработчик модуля CDT и его блог: <http://cdtdoug.ca> (**Doug Schaefer**)

- **Платформа Eclipse**

- **Переработано меню настройки перспектив** (Window → Perspective)
- Встроенный эмулятор локальной консоли (Linux)
- поддержка системы **контроля версий Git** (Eclipse EGit)
- **репозиторий** дополнительных инструментов **Eclipse Marketplace**
- Разворачивание среды разработки на полный экран (Alt+F11)
- Поиск в двоичных файлах
- и т. д. (см. статьи «Eclipse 4.x — New and Noteworthy» на официальном сайте Eclipse.org)

Установка обновлений для QNX Momentics IDE

- **Eclipse Marketplace**

- Онлайн-каталог дополнительных компонентов для IDE
 - В QNX Momentics IDE 7.0 клиент для доступа к Eclipse Marketplace установлен по умолчанию
 - В предыдущих версиях (4.7.1 — 5.0.1) устанавливается из соответствующего репозитория (см. выше)
- Использует технологию «Update Sites»

- **Update Sites** (репозитории обновлений)

- Базовый механизм обновления Eclipse IDE
- Позволяет устанавливать дополнительные компоненты и обновлять ранее установленные
 - Также позволяет обновлять компоненты, специфичные для QNX Momentics IDE, из специализированных репозиториях
- Примеры репозиториях:
 - Для Momentics 7.0
 - <http://download.eclipse.org/releases/neon/> (основной репозиторий Eclipse)
 - <https://www.qnx.com/swcenter/p2/ide/content.jar> (официальный репозиторий QNX)

Дополнительные инструменты

- **Mylyn**
 - Работа с системами совместной работы над исправлениями
- **Eclipse Runner**
 - Ускоренный запуск приложений (**Run Configurations**)
- **HEP (Hex Editor)**
 - Шестнадцатеричный редактор
- **wrapper**
 - поддержка функциональности vi/vim в стандартном редакторе
- **cppcheclipse / CodeCheckerEclipsePlugin**
 - Поддержка статических анализаторов кода (cppcheck / ClangSA)
- **bracketeer for C/C++ (CDT)**
 - Автокомментирование блоков кода C/C++
- **distcc4eclipse**
 - Мониторинг системы распределённой сборки (distcc)
- И Т.Д.

The screenshot displays the Eclipse IDE interface. At the top, the 'Runner' window shows a project structure with 'test.g (with -a argument)' selected. Below it, the 'test.c' editor shows the following code:

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int main(int argc, char *argv[]) {
5     printf("Welcome to the QNX Momentics IDE\n");
6     return EXIT_SUCCESS;
7 }
8
```

At the bottom, the 'Distcc Status' window shows a table of compilation tasks:

Host	Slot	File	Status	Time
10.20.24.87	0	test-inserter.cc	COMPILE	19964
10.20.24.87	6	test-bench-lte.cc	CPP	19990
10.20.24.87	7	cylindrify-cdma.cc	CPP	20229
10.20.25.163	0	AluNLabelOptions.cpp	COMPILE	20169
10.20.25.163	0	cylindrify-sam-lte.cc	CPP	20243
10.20.25.163	1	test-bench-inserter.cc	COMPILE	19907
10.20.25.163	1	test-cylinder-repair-umts.cc	CPP	20048
10.20.25.163	4	cylindrify-evdo.cc	CPP	20203
10.20.25.163	4	test-rtr-lte.cc	COMPILE	19934
10.20.25.163	7	test-bench-evdo.cc	COMPILE	19979
10.20.25.66	1	BookmarkAction.cpp	CPP	20172

An overlay window titled 'Conditions' shows a code snippet with annotations:

```
if( a == 1 && b == 2 && c == 3)
{
    /* if(a == ... & c == 3) */
    else if (a == 0)
    {
        /* if(a == 0) */
        else if (a == 1)
        {
            /* if(a == 1) */
        }
        /* else of if(a == 1) */
    }
}
```

Arrows point from the text 'Autocomments (hints) of which condition ends' to the highlighted code blocks.

- Способы установки инструментов и обновлений
 - **Сетевой**
 - Из каталога **Eclipse Marketplace**
 - В **QNX Momentics IDE 7.0** клиент **Eclipse Marketplace** присутствует по умолчанию (в версии 5.0 и ранее требуется установка из репозитория)
 - Из репозитория разработчиков соответствующего инструмента
 - Пример:
 - Инструмент: **cppcheclipse**
 - Репозиторий: <https://dl.bintray.com/cppcheclipse/p2/updates/>
 - **Локальный**
 - Из предварительно созданной локальной копии репозитория соответствующего инструмента
 - методика: http://wiki.eclipse.org/Equinox_p2_Repository_Mirroring
 - скрипт: <http://forum.kpda.ru>

Совместное использование IDE разных версий

- Пример конфигурации, включающей в себя основной инструментарий и две дополнительные версии IDE:
 - **Базовый QNX SDP**
 - **QNX Momentics 4.7 (в составе QNX SDP 6.5.0)**
 - Каталог продукта: `/opt/qnx650/host/linux/x86/usr/qde/eclipse`
 - Рабочий каталог (Workspace): `/home/user/ide-4.7-workspace`
 - **Новые версии Momentics IDE**
 - **QNX Momentics 7.0**
 - Каталог продукта: `/opt/momentics7`
 - Рабочий каталог (Workspace): `/home/user/ide-7.0-workspace`
 - Способы обмена проектами между IDE:
 - Механизм **Import/Export**
 - Общие **репозитории (SVN и т. д.)**
 - Все версии IDE работают в рамках единой лицензии на комплект разработчика (какое-либо дополнительное лицензирование не требуется)

Среда разработки Qt Creator IDE

- Основная среда разработки приложений на базе платформ **Qt 4.x/5.x**
- Возможности:
 - Создание и отладка приложений с учётом особенностей платформы **Qt (сигналы, слоты, визуальные компоненты и т.д.)**
 - **Связь с целевой системой (QNX/ЗОСРВ «Нейтрино»)** для удалённого запуска приложений и автоматизированной загрузки библиотек Qt
 - Поддержка **Valgrind** - инструмента анализа работы с памятью и профилирования
 - Поддержка визуальной разработки приложений с помощью подсистемы **Qt Designer**
 - Поддержка модулей **статического анализа** (cppcheck)
 - Поддержка фреймворков **модульного тестирования** (qtest, googletest)
- В качестве инструментальной платформы поддерживаются ОС семейств **Windows и GNU/Linux**



Среда разработки Qt Creator IDE — что нового?

- Краткий обзор изменений

- Развитие: **3.6 (март 2016)** → **4.0 (июль 2016)** → **4.2 (апрель 2017)** → **4.6 (март 2018)** →
- Улучшена поддержка актуальных стандартов (**C++14, C++17**)
- Добавлена поддержка **статического анализатора clang** на базе **clang-tidy**
- Исправления и доработки в части взаимодействия с **целевыми машинами QNX**
 - Поддержка подключение отладчика к ранее запущенным процессам на целевых системах
 - Убрано сообщение «**sh: slog2info: cannot execute - No such file or directory**»
- и т.д.

- «**What's New**»:

- **RSS-канал**: <http://blog.qt.io/feed/>
- <http://blog.qt.io/blog/category/qtcreator/>
- <https://code.qt.io/cgit/qt-creator/qt-creator.git/plain/dist/>



- Базовый инструментарий (Core Development Tools)
 - Компилятор и вспомогательное ПО
- IDE (интергрированная среда разработки)
 - QNX Momentics IDE
 - Qt Creator
- **Дополнительные инструменты и утилиты**
 - Инструменты динамического анализа
 - Инструменты статического анализа
 - Инструменты модульного тестирования
 - Средства оптимизации процесса сборки ПО
 - Прочее

- Стандартные средства
 - **QNX Application Profiler, QNX System Profiler, QNX Code Coverage, QNX Memory Analysis, mudflap (gcc <= 4.8)**
- Дополнительные средства
 - **Valgrind`s Tool Suite**
 - профилирование памяти, поиск утечек, анализ эффективности работы с кэш-памятью
 - анализ процесса синхронизации в многопоточных приложениях («race conditions»)
 - построение графов вызовов и прикладное профилирование
 - **GCC Address Sanitizer (gcc >= 4.8)**
 - Анализатор ошибок работы с памятью
 - «Use after free», «Use after return» и т.д.
 - Переполнение «кучи», стэка и т.д.
 - Утечки памяти

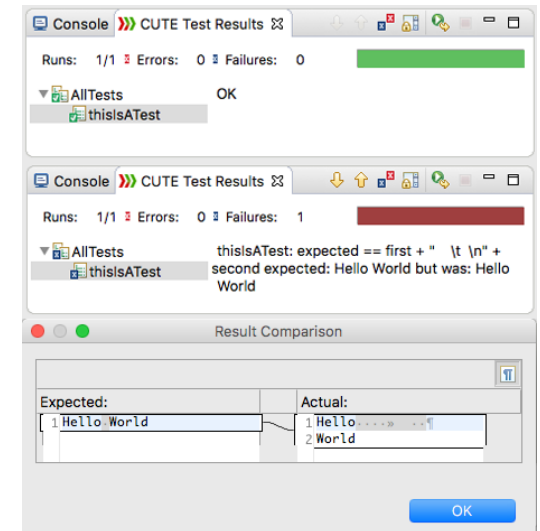
Инструменты статического анализа

- Основные задачи:
 - выявление ошибок еще до непосредственной компиляции приложений, в т.ч. ошибок, трудно поддающиеся локализации с помощью других методик и инструментов (отладчиков, динамических анализаторов и т. д.)
 - **Синтаксические ошибки:** `memset (dst, val, len) <=> memset (dst, len, val)`
 - **Смысловые ошибки:** `if(obj.x == x || obj.y == y || obj.z == y) {...}`
 - проверка на соответствие стандартам безопасного программирования
 - **NASA JPL, MISRA C/C++, CERT C** и т.д.
- Использование при разработке ПО для QNX и ЗОСРВ «Нейтрино»
 - Различные инструменты статического анализа ПО, разрабатываемого на языках C и C++:
 - **cppcheck, Clang Static Analyzer, LDRA Tool Suite, splint, cpplint, PC-lint, PVS-Studio** и т. д.
 - Инструменты статического анализа сценариев командного интерпретатора («скриптов»):
 - **shellcheck**

Инструменты модульного тестирования

- Решаемые задачи
 - Автоматизация процесса тестирования ПО (**unit testing**)
 - Оптимизация процесса анализа покрытия кода (**code coverage**)
 - Применение альтернативных методик разработки
 - TDD — **Test-Driven Development**
- Платформы модульного тестирования (C/C++):
 - **CUnit, CppUnit, CUTE, Qt Tests, Google Tests** и т. д.
- Пример интеграции:
 - **CUnit**
 - `ntox86-gcc AutomatedTest.c ../src/AppSourceFiles.c -lcunit -I ../src/ -I $QNX_TARGET/usr/include/CUnit/`
 - **CUTE (C++ Unit Testing Easier), Google Tests**
 - Полноценная интеграция в QNX Momentics IDE*

* рекомендуемая версия QNX Momentics IDE - 5.0 и выше



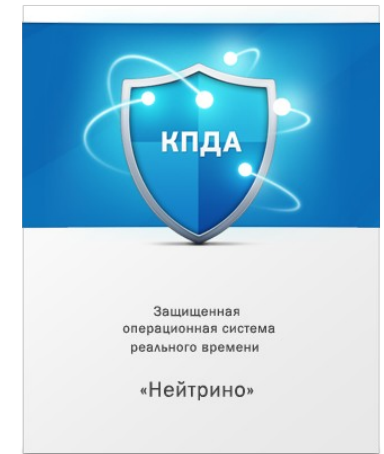
Оптимизация процесса сборки ПО

- **distcc** - система распределённой сборки
 - Позволяет выполнять сборку приложений с использованием распределённой вычислительной архитектуры, т.е. для компиляции файлов исходных кодов задействуются вычислительные ресурсы в локальной сети
 - Часто используется в связке с инструментом кэширования результатов компиляции (**ccache**)
- **ccache** - система кэширования результатов компиляции
 - сохраняет промежуточные результаты компиляции для снижения нагрузки на препроцессор
- **Jenkins CI** - система автоматизированной сборки и т. н. «непрерывной интеграции»
 - Позволяет осуществлять анализ, сборку и тестирование разрабатываемых программных продуктов в автоматизированном режиме
 - Интегрируется с QNX SDP с помощью сценариев командного интерпретатора («скриптов»)
- И т.д.



Основные тенденции и направления развития

- Инструментарий разработчика приложений для **ЗОСРВ «Нейтрино»** и **QNX** развивается в соответствии с актуальными тенденциями и технологиями разработки, отладки и тестирования программного обеспечения
 - стек **gcc/binutils/gdb (gcc 5.x/4.x + наработки для портирования следующих версий)**
 - расширяемая **IDE** на базе **Eclipse CDT + Qt Creator**
 - возможно совместное использование как базовых, так и дополнительных (внешних) инструментов «из мира» **Unix/Linux**
- Осуществляется техническая поддержка и адаптация инструментальных средств, в т.ч. под отечественные вычислительные платформы



<http://forum.kpda.ru> | support@kpda.ru

- Вопросы?

Спасибо за внимание

Игорь Рондарев

инженер-программист

ООО «СВД Встраиваемые Системы»

тел.: +7 (812) 346-8956

факс: +7 (812) 346-8953

<http://www.kpda.ru> | <http://forum.kpda.ru> |

<http://www.swd.ru>