

Развитие экосистемы кроссплатформенной разработки ПО для QR ОС

Орлов Михаил Алексеевич

ООО «НТП Криптософт»

Ул. Лермонтова, 3, Пенза, 440026, Россия

m.orlov@cryptosoft.ru

Актуальность

Актуальность проводимой работы обусловлена необходимостью создания экосистемы для эффективной разработки программного обеспечения для отечественной операционной системы QR ОС и портирования существующего программного обеспечения с открытым исходным кодом. К такой экосистеме предъявляется множество нормативно-правовых и нормативно-технических требований в области разработки безопасного программного обеспечения (РБПО). Реализация данных требований необходима для противодействия угрозам информационной безопасности и обеспечения соответствия национальным стандартам при разработке ПО для QR ОС.

Состав инфраструктуры разработки

Для обеспечения полного цикла разработки программного обеспечения в среде QR ОС сформирована специализированная инфраструктура, включающая инструменты для настройки, сборки, контроля версий и отладки программных проектов.

В качестве основы инфраструктуры выбрана связка LLVM, CMake и Ninja. Дополнительно в экосистему включена система автоматизации сборки Meson, которая применяется в ряде таких актуальных проектов, как PostgreSQL, не поддерживающих CMake.

Для работы системы Meson был портирован интерпретатор Python и обеспечена полная инфраструктура для разработки и отладки ПО на данном языке программирования. Для изоляции среды разработки также был реализован локальный репозиторий пакетов – QPyPi, обеспечивающий автономную разработку и обеспечение жизненного цикла ПО на языке Python.

Такой подход обеспечивает высокую степень переносимости решений и позволяет как осуществлять сборку непосредственно в QR ОС, так и организовывать кросс-компиляцию между QR ОС и другими операционными системами. Это расширяет возможности интеграции и ускоряет процесс адаптации существующего ПО для работы в защищенной среде QR ОС.

Для реализации системы контроля версий в QR ОС был адаптирован инструмент git, основанный на библиотеке libgit2, что обеспечивает стабильную работу с репозиториями.

Инфраструктура LLVM

Инфраструктура LLVM представляет собой мощный и универсальный набор инструментов, предназначенный как для разработки программного обеспечения, так и для расширения самой платформы путём создания новых компиляторов, оптимизаторов, анализаторов и других инструментальных средств. В её состав входят компиляторы для популярных языков программирования высокого уровня, ассемблер, интерактивный отладчик, а также ряд других полезных подсистем, обеспечивающих всестороннюю поддержку разработки и оптимизации программного кода.

В ходе портирования инфраструктуры LLVM в операционную систему QR ОС было внесено свыше 25 тысяч изменений в исходном коде LLVM; разработан специализированный драйвер компилятора clang – clang-qr, учитывающий особенности QR ОС в процессе компиляции ПО; разработаны инструментальные средства для кросс-компиляции из ОС Linux и Windows в QR ОС, а также из QR ОС в Linux; существенно расширен набор инструментов разработчика QR ОС для обеспечения совместимости с LLVM, а также реализованы сценарии для автоматизированной сборки и подготовки пакетов LLVM к поставке.

Средство автоматизации сборки CMake

Для автоматизации процессов сборки и установки программного обеспечения предлагается использовать инструмент CMake, который обладает относительно простым и функциональным языком описания сценариев, позволяет тестировать имеющиеся среды сборки и гибко адаптироваться к имеющимся возможностям. CMake не выполняет сборку, а подготавливает сценарии для широкого спектра систем сборки, обеспечивая тем самым гибкость и универсальность возможных подходов.

В ходе портирования CMake было внесено около 8 тысяч изменений в исходный код CMake; обеспечена поддержка платформы QP ОС; разработаны сценарии кросс-компиляции из различных систем в QP ОС; обеспечено частичное выполнение требований национальных стандартов по разработке безопасного программного обеспечения.

Система сборки Ninja

В качестве основной системы сборки, поддерживаемой CMake, выбрана система Ninja, которая является наиболее универсальной, эффективной и переносимой.

В ходе портирования системы Ninja было внесено порядка 300 изменений в базовую версию системы и подготовлена рабочая версия программы, функционирующая в QP ОС.

Интерпретатор Python в QP ОС

Особое внимание в нашей инфраструктуре разработки уделено поддержке языка программирования Python как одного из наиболее востребованных в современных разработках. Под QP ОС доступно более 200 пакетов Python, обеспечивающих широкие функциональные возможности для создания, тестирования и отладки программных решений. Реализована возможность полноценной интерактивной отладки непосредственно в среде QP ОС, а также эмуляция среды разработки QP ОС в Windows, что значительно упрощает начальный этап разработки.

Поддержка Python в QP ОС также позволила внедрить альтернативный механизм автоматизации сборки и конфигурирования программного обеспечения — систему Meson. Среди ключевых преимуществ Meson — простота синтаксиса, высокая производительность и встроенная поддержка кросс-компиляции.

В рамках доработки Meson обеспечена поддержка триплетов QP ОС, что позволило использовать его как часть кросс-компиляционного конвейера. В частности, с применением Meson успешно реализована сборка СУБД PostgreSQL 17 для QP ОС, что демонстрирует практическую эффективность данного подхода.

Драйвер компилятора clang-qp

Ключевой доработкой платформы LLVM стал новый драйвер компилятора Clang. Драйвер в архитектуре LLVM играет роль связующего звена, обеспечивающего взаимодействие между компонентами инфраструктуры LLVM и операционной системой или средой разработки. Параметры, передаваемые на вход драйвера, интерпретируются в соответствии с требованиями конкретной целевой платформы или среды. Таким образом обеспечивается независимость решения от сторонних компонентов.

Clang-qp обладает расширенным набором параметров, специфичных для QP ОС, выполняет определение и настройку окружения сборки и обеспечивает простую кросс-компиляцию итоговых исполняемых файлов, готовых к переносу и работе в QP ОС.

Кроссплатформенная разработка для QP ОС

Для обеспечения кроссплатформенной разработки программного обеспечения для защищённой операционной системы QP ОС компоненты инфраструктуры разработки были доработаны в части поддержки новых триплетов целевых систем, которые определяют особенности целевой платформы при кросс-компиляции и содержат информацию об аппаратной платформе, производителе, операционной системе и окружении. Таким образом

была реализована поддержка новых триплетов для основных аппаратных платформ, поддерживаемых QP ОС: x86, x64, Arm32, Arm64.

Также нами был разработан специальный toolchain-файл, который обеспечивает создание виртуальной файловой системы для использования заголовочных файлов из регистронезависимых файловых систем в регистрозависимых, подключение и настройку набора средств разработки под QP ОС, а также автоматическую настройку параметров компиляции и сборки.

Таким образом, с использованием доработанной экосистемы разработки можно обеспечить портирование практически любого ПО, написанного на языках «Си» и C++ под сторонние платформы, в QP ОС, что позволит конечному пользователю работать со знакомыми инструментами в новой среде.

Аналогичным образом, за счет гибкой системы триплетов, реализованной в инфраструктуре LLVM, возможна кроссплатформенная сборка из QP ОС в ОС семейства Linux.

Следующие шаги

- портирование менеджера зависимостей для C/C++ Conan,
- реализация интегрированной среды разработки непосредственно в защищенной отечественной операционной системе QP ОС,
- доработка основных компонентов LLVM с целью обеспечения полного соответствия новейшим стандартам по РБПО,
- интегрирование обновлений в компоненты экосистемы разработки и внедрение новых функциональных возможностей.