



КРИПТОСОФТ
НАУЧНО-ТЕХНИЧЕСКОЕ ПРЕДПРИЯТИЕ

РАЗВИТИЕ ЭКОСИСТЕМЫ КРОССПЛАТФОРМЕННОЙ РАЗРАБОТКИ ПО ДЛЯ QR ОС

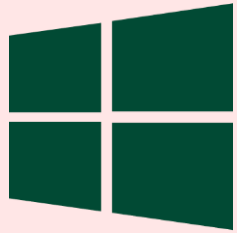
Орлов Михаил Алексеевич
ООО «НТП Криптософт»

АКТУАЛЬНОСТЬ РАБОТЫ

- Создание эффективной и безопасной экосистемы разработки для отечественной операционной системы QR ОС:
 - Повышение качества и безопасности продуктов компании
 - Соблюдение нормативно-правовых и нормативно-технических требований в области РБПО
 - Противодействие угрозам информационной безопасности
- Обеспечение процессов портирования ПО с открытым исходным кодом



КРОССПЛАТФОРМЕННАЯ РАЗРАБОТКА ПО



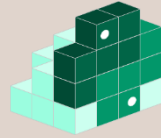
Windows



CMake



Git



QPyPi



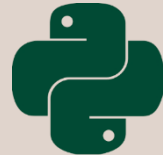
Meson



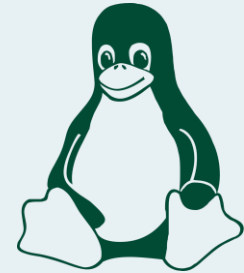
LLVM



Ninja



QPython



Linux



QP OC

ИНФРАСТРУКТУРА LLVM

- Набор компиляторов для языков высокого уровня
- Платформонезависимая система кодирования машинных инструкций (байткод)
- Оптимизаторы и санитайзеры кода
- Статический анализатор кода
- Сборщик исполняемых файлов
- Интерактивный отладчик
- Реализация стандартных библиотек для языков C и C++
- Расширяемая модульная структура



ОСОБЕННОСТИ LLVM ДЛЯ QP ОС

- Внесено свыше 25 тысяч изменений в исходном коде LLVM
- Создан собственный драйвер clang-qp, учитывающий особенности экосистемы QP ОС и позволяющий осуществлять компиляцию в QP ОС и кросс-компиляцию из и в QP ОС
- Разработаны инструментальные средства для кросс-компиляции из ОС Linux, ОС Windows в QP ОС
- Разработаны инструментальные средства для кросс-компиляции из QP ОС в ОС Linux
- Существенно расширен набор разработчика QP ОС для обеспечения совместимости с LLVM
- Реализованы сценарии для автоматизированной сборки и подготовки к поставке пакетов LLVM



СРЕДСТВО АВТОМАТИЗАЦИИ СБОРКИ СМАКЕ

- Кроссплатформенное средство автоматизации процессов сборки и установки программного обеспечения
- Относительно простой и функциональный язык описания сценариев
- Тестирование среды сборки и гибкая настройка под имеющиеся возможности
- Генерация сценариев для различных интегрированных сред разработки, целевых систем сборки, операционных систем, библиотек и инструментов



ОСОБЕННОСТИ СМАКЕ ДЛЯ QP ОС

- Внесено порядка 8 тысяч изменений в исходном коде CMake
- Обеспечена поддержка целевой платформы QP ОС
- Реализована поддержка нового драйвера clang-qp
- Разработан сценарий конфигурации для автоматизации процессов кросс-компиляции из различных операционных систем в QP ОС
- Обеспечено выполнение части требований национальных стандартов по РБПО

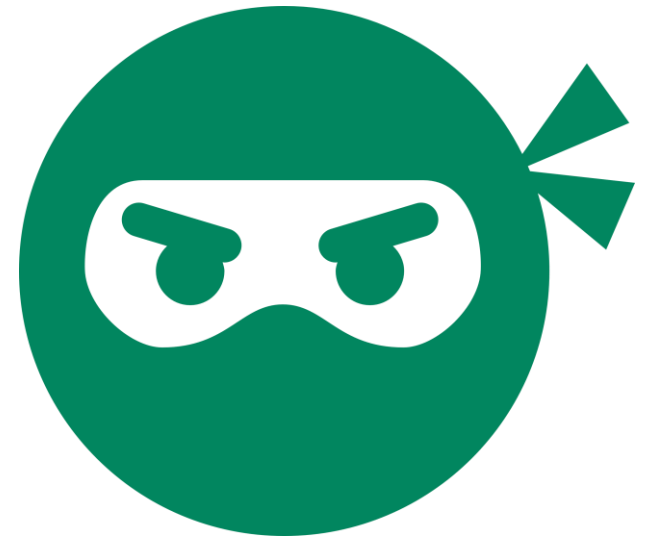
QPOS.cmake

```
...
# Проверка наличия необходимых параметров
CMake.
if (NOT LLVM_BIN_DIR OR NOT
QPOS_DEVKIT_DIR)
    message (SEND_ERROR "Необходимо задать
параметры CMake: " "LLVM_BIN_DIR и
QPOS_DEVKIT_DIR.")
endif ()
# Параметры смаче для целевой платформы.
set (CMAKE_SYSTEM_NAME QPOS CACHE STRING
"Целевая система")
# Проверка на наличие средств сборки (LLVM)
в системе.
if (NOT EXISTS "${LLVM_BIN_DIR}/clang-
qp${SYSTEM_EXE_EXTENSION}")
...

```

СИСТЕМА СБОРКИ NINJA

- Кроссплатформенная система сборки
- Существенное ускорение процессов сборки и пересборки
- Эффективное использование системных ресурсов
- Поддержка очень больших проектов
- Основная система сборки проектов CMake и Meson в QP ОС



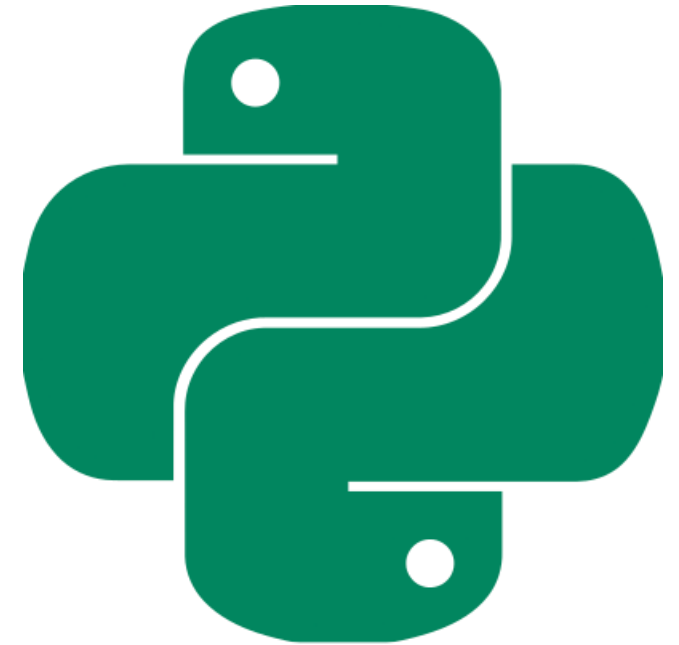
СИСТЕМА УПРАВЛЕНИЯ ВЕРСИЯМИ GIT

- Обеспечение целостности и необходимого уровня конфиденциальности кода программного обеспечения
- Основа для безопасной системы сборки программного обеспечения
- Гарантированный доступ к хранимым документам в изолированной сети компании
- Надёжный контроль авторства и истории изменения документов
- Возможность независимого параллельного редактирования документов несколькими разработчиками



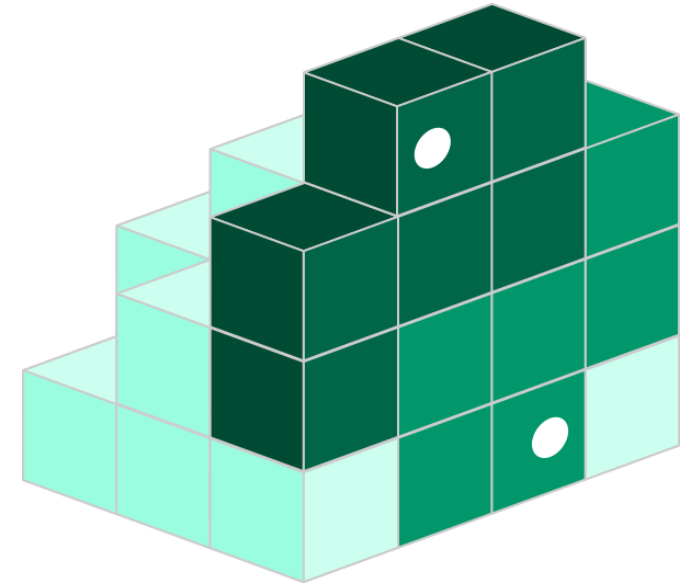
ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON

- Современный высокоуровневый язык программирования общего назначения
- Более 200 пакетов для QP ОС
- Эмуляция среды разработки QP ОС в среде Windows
- Полноценная интерактивная отладка непосредственно в QP ОС
- Портирование выполнено с помощью инфраструктуры LLVM



ИНДЕКС ПАКЕТОВ QPURI

- Локальное зеркало индекса пакетов из международного индекса PyPi
- Автономная работа в изолированной сети компании, с возможностью модификации существующих и разработки новых пакетов
- Хранение дистрибутивов с исходным кодом используемых пакетов
- Автономная сборка и кэширование бинарных дистрибутивов используемых пакетов
- Взаимодействие через штатные интерфейсы языка Python и с помощью веб-интерфейса



СРЕДСТВО АВТОМАТИЗАЦИИ СБОРКИ MESON

- Современная система автоматизации сборки и конфигурирования программного обеспечения
- Простой и читаемый синтаксис на базе Python-подобного DSL
- Высокая производительность сборки
- Встроенная поддержка кросс-компиляции
- В результате портирования:
 - Обеспечена поддержка триплетов целевой системы QP ОС, а также компонентов экосистемы разработки
 - Успешно реализована сборка СУБД PostgreSQL 17 для QP ОС



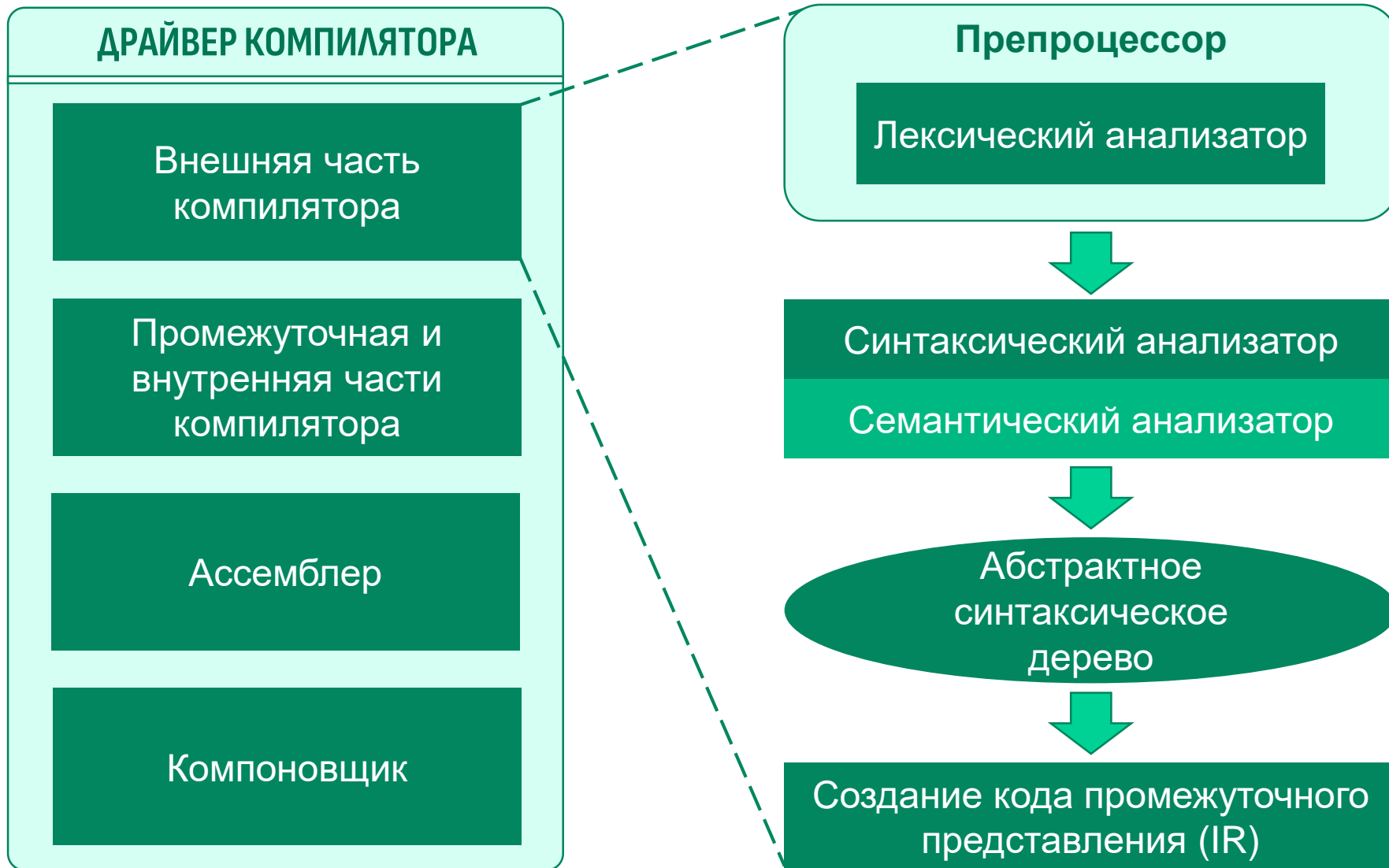
ОСОБЕННОСТИ MESON ДЛЯ QP ОС

- Внесено более тысячи изменений в исходном коде Meson
- Обеспечена поддержка целевой платформы QP ОС
- Реализована поддержка нового драйвера clang-qp
- Разработаны сценарии конфигурации для автоматизации процессов кросс-компиляции из различных операционных систем в QP ОС

x86_64-win-cryptosoft-qpos-llvm.txt

```
# Константы
[constants]
path_to_qpsdk = 'C:\qpsdk'
path_to_devkit = path_to_qpsdk + '\devkit'
devkit_libs = 'WinCrt WinSdk'
...
# Инструменты сборки
[binaries]
cmake = path_to_qpsdk + '\bin\cmake.exe'
ar = path_to_qpsdk + '\bin\llvm-lib.exe'
c_ld = path_to_qpsdk + '\bin\lld-link.exe'
...
# Целевая машина
[host_machine]
system = 'qpos'
cpu_family = 'x86_64'
cpu = 'x64'
endian = 'little'
```

ДРАЙВЕР КОМПИЛЯТОРА SLANG-QP



ДРАЙВЕР КОМПИЛЯТОРА CLANG-QP

Для учёта специфики операционной системы QP ОС реализован новый драйвер компилятора **CLANG-QP**, который обеспечивает:

- Упрощение процессов кросс-компиляции для QP ОС
- Расширенный набор параметров, специфичных для QP ОС
- Совместимость с QP ОС DevKit
- Поддержку специфики API и ABI QP ОС
- Специальную настройку препроцессора, компилятора и сборщика
- Расширенную диагностику процессов компиляции и сборки

```
~/llvm18_qp/bin/clang-qp -help
```

OVERVIEW: clang LLVM compiler

USAGE: clang-qp [options] file...

OPTIONS:

-### Print (but do not run) the commands to run for this compilation

--amdgpu-arch-tool=<value> Tool used for detecting AMD GPU arch in the system.

--analyze Run the static analyzer

--config=<file> Specify configuration file

...

Опции совместимости с QP ОС:

-fqp-compatibility Параметр позволяет сообщить компилятору о совместимости с QP ОС

/qpos-devkit-dir=<dir> Путь к набору разработчика QP ОС

/qpos-devkit-libs=<lib1;lib2;...> Список используемых библиотек из набора разработчика QP ОС

КРОССПЛАТФОРМЕННАЯ РАЗРАБОТКА ДЛЯ QP ОС

Процесс кросс-компиляции проекта под QP ОС в ОС Linux включает:

- Определение и настройку окружения сборки
- Эмуляцию целевой файловой системы
- Выбор целевого бинарного интерфейса
- Выбор и настройку необходимого инструментария
- Генерацию сценариев сборки
- Формирование целевых исполняемых файлов, готовых к переносу и работе в QP ОС

```
dev@dev:~/git/llvm-qp/examples/hello_clang++/build$ cmake -G Ninja
..
-DMAKE_TOOLCHAIN_FILE=$HOME/git/llvm-project.dev/llvm/cmake/platforms/QPOS.cmake -DHOST_ARCH=x86_64
-DLLVM_NATIVE_TOOL_DIR=$HOME/llvm18/bin
-DQPOS_DEVKIT_DIR=$HOME/DevKit
-DQPOS_DEVKIT_LIBS="Ucrt;Crt;SdkUm;SdkShared"
-DMAKE_BUILD_TYPE=Debug
-- Целевая система: QP ОС Мастер 4.579
-- Целевая платформа: x64
-- Пути поиска библиотек: /home/dev/DevKit/Sdk/Libs/x64
-- Текущая файловая система чувствительна к регистрам. Запущена эмуляция целевой файловой системы.
-- Целевая платформа в CMake: QPOS
-- The CXX compiler identification is Clang 18.0.0 with MSVC-like command-line
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /home/dev/llvm18/bin/clang-qp - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done (3.5s)
-- Generating done (0.0s)
-- Build files have been written to:
/home/dev/git/llvm-qp/examples/hello_clang++/build
dev@dev:~/git/llvm-qp/examples/hello_clang++/build$ ninja install
[2/3] Linking CXX executable hello_clang++.qpx
[2/3] Install the project...
-- Install configuration: "Debug"
-- Installing:
/home/dev/git/llvm-qp/examples/hello_clang++/bin/hello_clang++.qpx
dev@dev:~/git/llvm-qp/examples/hello_clang++/build$
```

СЛЕДУЮЩИЕ ШАГИ



ПОРТИРОВАНИЕ МЕНЕДЖЕРА ЗАВИСИМОСТЕЙ ДЛЯ C/C++ CONAN

Доработка современного менеджера зависимостей, автоматизирующего процессы сборки и сопровождения проектов, разрабатываемых на языках C/C++, а также обеспечивающего реализацию требований по РБПО



ПОДДЕРЖКА ЭКОСИСТЕМЫ

Интегрирование обновлений в компоненты экосистемы разработки и внедрение новых функциональных возможностей.



РЕАЛИЗАЦИЯ ИНТЕГРИРОВАННОЙ СРЕДЫ РАЗРАБОТКИ

Обеспечение замкнутого цикла разработки в защищенной отечественной операционной системе QP ОС.



ОБЕСПЕЧЕНИЕ ПОЛНОГО СООТВЕТСТВИЯ СТАНДАРТАМ ПО РБПО

Доработка и расширение инфраструктуры разработки для QP ОС с целью обеспечения полного соответствия новейшим стандартам по РБПО.



КРИПТОСОФТ
НАУЧНО-ТЕХНИЧЕСКОЕ ПРЕДПРИЯТИЕ

СПАСИБО ЗА ВНИМАНИЕ!



(8412) 99-10-91, 56-00-77
mails@cryptosoft.ru

www.cryptosoft.ru