

# Проблемы кросс-сборки: исполнение бинарных файлов

Михаил Шалаев  
Москва  
АО МЦСТ

## Аннотация

За время существования свободного программного обеспечения создано огромное количество программных пакетов, сборка каждого из которых может иметь свои особенности. Разработано несколько популярных и широко распространенных систем конфигурации, большинство из которых предоставляет вариативность выбора целевой операционной системы. Понятие кросс-компиляции и необходимость сборки для целевой аппаратной платформы появились достаточно давно, но до сих пор многие разработчики программного обеспечения предполагают, что их продукты будут собираться только на нативных системах. К сборке таких продуктов нужно подходить индивидуально. В докладе раскрыты наиболее часто возникающие проблемы и возможные пути их решения.

При создании вычислительной машины новой архитектуры неизбежно возникает проблема её программного обеспечения. Работа посвящена проблемам создания пользовательского окружения в кросс-режиме для широко используемых и новых аппаратных платформ.

Исходные требования к системе создания пользовательского окружения следующие:

- наименьшие затраты на адаптацию программного кода к новой платформе (максимальная совместимость с существующей кодовой базой);
- ограниченный доступ к аппаратуре;
- возможность быстрого масштабирования процесса разработки;
- поддержка нескольких архитектур.

За время существования свободного программного обеспечения создано огромное количество программных пакетов, сборка каждого из которых может иметь свои особенности. Разработано несколько популярных и широко распространенных систем конфигурации, большинство из которых предоставляет вариативность выбора целевой операционной системы.

При постановке задачи сборки какого-либо из существующих linux дистрибутивов в кросс-режиме возникают дополнительные проблемы, связанные с особенностями сборочных сред этих дистрибутивов и де-факто отсутствием стандартов написания сборочных скриптов при добавлении пакетов. Наиболее часто встречаются следующие проблемы:

- отсутствие опций для определения целевой платформы сборки при конфигурации;
- явное определение переменных, используемых при сборке, например компилятора, без возможности переопределения извне, несмотря на имеющиеся механизмы динамического определения значений для этих переменных;
- явное указание путей поиска исполняемых библиотек и заголовочных файлов.

Эти и подобные проблемы можно решить как правками «на лету», так и доработкой исходных текстов.

Еще одна проблема, возникающая при кросс-сборке некоторых пакетов заключается в необходимости запускать только что собранные исполняемые файлы. Это возникает в ряде случаев:

- запуск собранных исполняемых программ в процессе проведения внутренних тестов (make check);
- запуск программ для получения документации, например при использовании утилиты help2man;
- запуск промежуточных программ, используемых при генерации файлов, необходимых для дальнейшей сборки программного пакета.

Если решить первую проблему можно отключением внутреннего тестирования пакета, то для решения остальных можно предложить следующие подходы:

- сборка на вычислительном комплексе с подменой исполнения целевых бинарных файлов локальными;
- сборка на вычислительном комплексе с исполнением целевых бинарных файлов в виртуальной машине, например, qemu;
- сборка на вычислительном комплексе с исполнением целевых бинарных файлов на удаленной машине (использование remote shell);
- сборка на вычислительном комплексе с целевой архитектурой, компиляция проводится кросс-компилятором с использованием distcc.

При использовании данных подходов появляется возможность удовлетворить требованиям, поставленным к конкретному конечному продукту. Выбирать из них следует в зависимости от доступных программных решений и имеющегося количества аппаратных комплексов.