

Система сборки для минимальных POSIX-совместимых ОС и окружений
Аникин Антон Андреевич; МЦСТ; 119049, Москва, Ленинский проспект, д. 51;
anikin_a@mcst.ru

Всякий разработчик прикладного ПО при желании поддержать свой продукт в мире POSIX неизбежно сталкивается с высокой фрагментацией экосистем, выстроенных вокруг платформ данного семейства. На основе Linux, Illumos и BSD создан ряд дистрибутивов ПО, которые несовместимы между собой по множеству характеристик, из которых в рамках темы мы выделим нас интересующую: в дистрибутивах по-разному реализованы одни и те же базовые функции, необходимые для развёртки и сопровождения прикладных программных продуктов.

В случае решений для встраиваемых систем фрагментация носит ещё более укоренённый характер, т.к. в подобных программных системах изначально востребована максимальная специализация и эффективность в ущерб какой бы то ни было унификации.

Проблему совместимости среды этапа исполнения (чаще всего, совместимости ABI в том или ином виде) регулярно поднимают, и периодически появляются предложения на эту тему (стандарт LSB, унифицированные образы AppImage\Flatpack, контейнеры). Однако этап сборки и дальнейшего формирования конечной поставки программы остаётся максимально разрозненным.

Для унифицированных сборок в различных POSIX-окружениях разработан сборочный инструмент `apu`.

Архитектура инструмента `apu` выглядит следующим образом:

Непосредственная сборка одной единицы (программного пакета) выполняется прямолинейным кодом на shell, в котором описаны уникальные для пакета действия.

Система формирует скрипт, в который помещаются:

- типовые вспомогательные действия (одинаковые для каждого пакета в серии);
- уникальный для каждого пакета код, заданный пользователем.

Процесс сборки происходит через простой запуск данного скрипта, который можно затем отлаживать и анализировать классическим образом.

Вспомогательные действия реализованы через простые библиотеки и включают в себя разворот исходников, наложение патчей, проверка зависимостей этапа сборки, формирование установочного архива.

Формат бинарного результата максимально абстрагирован от описания сборки. Из единого сборочного файла возможно получать установочные пакеты различных форматов.

Все действия возможно выполнять над единственным пакетом или над серией пакетов.

Проект сборки ведётся от простого пользователя и находится в автономной директории по аналогии с проектом IDE или репозиторием `git`.

Конфигурация через командную строку и настройки формата `.ini` (секция данных, опция=значение).

Минимализм требований и совместимость

Система сборки написана на shell и утилитах стандарта POSIX (без расширений GNU, возможностей `bash` или другой специфики какой-либо ОС). Поэтому она не требует компиляции и процесса бутстрапа, и готова к применению даже в минималистичных окружениях на широком наборе ОС и аппаратных архитектур.

Не требуется какой-либо специальной совместимой версии тулчейна для компиляции: используются существующие системные инструменты «как есть».

Система является чистым инструментом этапа сборки. Она не реализует собственный формат установочных пакетов и не ведёт параллельный независимый репозиторий (что делают `pkgsrc` или T2). Результат сборки совместим с «общим» дистрибутивом или компактной средой исполнения.

Общие возможности

Для пользователя-программиста реализован ряд функций, которые позволяют в компактном читаемом виде поддерживать в едином сборочном файле работу с разными ОС или тулчейнами, отключать или выключать опции по динамически проверяемым условиям.

Функции для частных сущностей вынесены в модули (таких, как поддержка установочных форматов deb или rpm). Модули работают независимо друг от друга, и добавление одного модуля не влияет на остальные. Отсюда возможна поддержка большого числа конфигураций без переусложнения основной функциональности. Модули подключаются из командной строки и из конфигурационных файлов.

Поддержаны широкие возможности для создания автономных исполнимых образов: функции кросс-компиляции, средства для разрезания кольцевых зависимостей, сборка с чистого листа без зависимостей от предыдущего состояния.

На пользователя-программиста накладывается относительно немного ограничений при написании своих сборочных файлов: он может дополнять типовые действия, полностью их менять, назначать свою структуру сборки.

Возможности для встроенных систем

Сочетание следующих факторов в any открывает широкие возможности для сборки прошивок, установочных образов, образов контейнеров и других видов самодостаточных сред, необходимых для компактных и встроенных устройств:

- полное разделение окружения для исполнения any (“хоста”) и компонент, используемых при сборке приложения (целевых зависимостей). В сочетании с кросс-компиляцией это обстоятельство позволяет поддерживать сборку поставки одновременно под целый ряд специализированных операционных систем. Даёт возможность масштабирования сборок, когда есть ограничения для целевых встраиваемых устройств (в том числе, когда устройства разрабатываются и существуют только в виде эмуляторов).

- сборка поставки с нуля в виде полного списка пакетов. Убирает критичность предыдущего репозитория в бинарном виде. Позволяет в любой момент получить полный объём поставки со всеми пакетами/компонентами, которые согласованы по ABI между собой.

Примеры и ссылки

Пример сборки образовательного дистрибутива Linux From Scratch с помощью any:

<https://opendistro.org/lfs/11.2/build.html>

Примеры простых сборочных файлов:

<https://opendistro.org/lfs/11.2/listing.html>

<https://opendistro.org/lfs/11.2/listing-pass1-binutils-2.39.html>

Начальное руководство по any:

<https://opendistro.org/ru/any/any-guide.7.html>