

Автоматизация портирования ОС «Альт» на новые аппаратные платформы

О портировании Альта на x86

Г. И. Фотенгауэр-Малиновский

ООО «Базальт СПО» / ALT Linux Team

23 сентября 2017 г. / OSSDEVCONF-2017, Калуга



Альт на разных платформах

- x86_64
- ppc64
- arm
- armh
- aarch64
- e2k



Стадии портирования

- 1 минимальная сборочная среда (toolchain, shell, make);
- 2 сборщик пакетов (rpmbuild);
- 3 сборочная среда в виде пакетов (для сборки в пересоздаваемом изолированном сборочном окружении);
- 4 массовая сборка пакетов;
- 5 пересборка всех собранных пакетов;
- 6 интеграция новой архитектуры в общую систему сборки пакетов (возможно, в «догоняющем» режиме);



Сборка на основе другого дистрибутива

Плюсы:

- всё, что нужно уже собрано
- можно пропустить стадию 1 и большую часть стадии 2

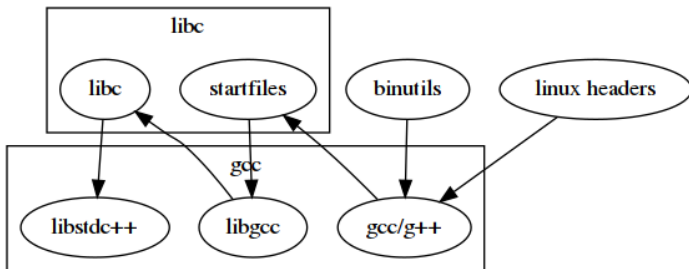
Минусы:

- приходится решать **непредсказуемые** проблемы, связанные с разницей между дистрибутивами
- плохо автоматизируемо



Стадия 1 (кросс-сборка)

Можно делать на любой поддерживаемой архитектуре, нужно только собрать среду для кросс-сборки.



Проблемы

Программы при сборке иногда запускают только что собранные исполняемые файлы, которые не обязаны вообще работать.



Стадии 2 и 3 (rpmbuild и сборочная среда)

- Удобно делать с помощью `qemu-user`, если он поддерживает целевую архитектуру.
- Можно использовать и любой другой дистрибутив, если есть возможность создать на нём изолированное окружение.



Стадии 4-6

Требуется оборудование, на котором уже загружен Альт и работает hasher.



Всё делалось вручную

А можно было бы автоматизировать большую часть работы.



bootstrap > arch support

Большая часть проблем портирования не зависит от архитектуры, потому что чаще всего проблема в наличии циклов зависимостей.



Изменяемость пакетов

- Любое изменение в пакетах, сборка которых автоматизируется может сломать всю систему автоматического портирования.
- Имеет смысл следить за работоспособностью автоматического портирования по мере изменения пакетов в Сизифе.



Выбор целевой архитектуры для периодической проверки

Стадию 1 можно тестировать вообще на любой целевой архитектуре, но лучше не x86.

Остальные стадии, наоборот, лучше делать на x86.



Что сделано

На данный момент сборка кросс-среды, стадии 1 и 2 полностью автоматизированы.



Планы

- сборка множества компиляторов и библиотек под разные архитектуры в Сизиф
- автоматизация сборки стадии 3 и отдельных сложных частей стадии 4.



Спасибо за внимание!

