

FOSS GNU/Linux ОС Apertis для встраиваемых устройств

Денис Пынькин, инженер-программист, Collabora Ltd, Великобритания

ОС Apertis[1] – это FOSS (Free and open source) GNU/Linux платформа, основанная на Debian/Ubuntu. Изначально Apertis разрабатывался для использования в информационно-развлекательных автомобильных системах, но с течением времени фокус разработки сместился на обеспечение базового функционала для широкого спектра встраиваемых устройств.

ОС Apertis

Цель проекта – предоставить базу и референсные примеры создания загрузочных образов дисков для встраиваемых устройств, с учетом специфики такого рода устройств.

Компоненты, которые используются в ОС Apertis либо в сборочной системе, открыты и доступны для использования как по отдельности, так и в составе других проектов, а большая часть изменений «уходят» в апстрим соответствующих проектов либо в ОС Debian.

На момент создания тезисов поддерживаются 3 целевые архитектуры:

- x86_64 (в том числе Qemu);
- arm64 (Renesas R-Car);
- armhf (i.mx 6, SabreLite).

Обеспечена поддержка различных вариантов ОС, различающихся как по функциональности, например с использованием минимального набора ПО либо предоставляющий графическое окружение на целевой системе, так и по внутренней организации – «классический» вариант с использованием пакетной базы и вариант, основывающийся на проекте libostree[2].

Одной из интересных и важных для производителей особенностей ОС Apertis является минимальный базовый набор ПО, который за редким исключением (например gcc), не включает в себя проекты, использующие лицензию GPLv3 и похожих, страхуя разработчиков конечных решений от неприятных лицензионных сюрпризов.

Подробнее с ключевыми особенностями ОС Apertis можно ознакомиться на портале с дизайн-документацией[3] и wiki[4], разумеется учитывая, что документация в открытых проектах может несколько отличаться от реального положения дел.

Обновления ОС с libostree

libostree[2] – это системная библиотека, а также CLI утилита `ostree`, обеспечивающие возможность атомарных обновлений операционной системы как с использованием сети Интернет (OTA-обновления), так и с использованием дисковых носителей (offline-обновления).

Проект libostree позволяет работать с деревьями ОС в стиле git, обеспечивая возможность держать на одной корневой файловой системе несколько версий ОС, в том числе различные варианты ОС и/или разные ОС (при обеспечении совместимости с libostree). Поскольку изменение «текущей» ОС производится в стиле git (checkout), то появляется возможность быстрого и простого отката к предыдущему состоянию при неудачном обновлении.

Отдельным пунктом хотелось бы упомянуть интеграцию libostree с различными загрузчиками для обеспечения автоматического определения невозможности полноценной загрузки ОС и отката к предыдущей «рабочей» версии. Для автоматического управления атомарными обновлениями и откатами в ОС Apertis, был создан специальный сервис Apertis Update Manager [5].

Краткий обзор достоинств и недостатков[6] libostree для встраиваемых устройств является частью доклада.

Создание загрузочных образов с помощью утилиты Debos

Утилита Debos[7][8] предназначена для создания кастомизированных версий операционных систем с использованием пакетной базы ОС Debian. Debos разработан в качестве инструмента для решения стандартных задач, возникающих при модификациях ОС.

Основная задача, которую решает Debos – максимальное упрощение для конечных пользователей описания создания образов систем, готовых к использованию на целевых устройствах, оставляя при этом достаточно возможностей для реализации любого нетривиального этапа процесса сборки.

Одной из ключевых особенностей утилиты Debos является интеграция с проектом libostree, позволяя создавать загрузочные образы дисков для ОС, использующих libostree.

Стандартная проблема, которая возникает при сборке загрузочного образа – это необходимость использования повышенных привилегий для некоторых шагов, таких как установка пакетов. Разными утилитами и дистрибутивами эта задача решается по-разному. Для Debos используется библиотека fakemachine[9], написанная Sjoerd Simons. Эта библиотека использует виртуальную машину Qemu, позволяя работать с повышенными привилегиями в текущей системе. Кроме того, такой подход позволяет без дополнительных затрат организовать сборку образа под любую архитектуру, поддерживаемую в Qemu.

В задачу утилиты не входит создание повторяемого сборочного окружения. Подразумевается, что для каждого проекта такое окружение уникально и должно создаваться другими средствами, например, Docker.

CI

Для обеспечения автоматической сборки и тестирования пакетов и готовых дисковых образов используются следующие системы и утилиты:

- Gitlab[10] – хранение и ревью исходного кода;
- Phabricator[11] – управление проектом и трекер ошибок;
- Jenkins[12] – управление сборочными процессами;
- Open Build Service[13] – сборка и публикация пакетной базы;
- Debos[7] – утилита для создания загрузочных образов дисков;
- LAVA[14] – автоматическое тестирование собранных дисковых образов;
- QA Apertis[15][16] – тесты и набор скриптов для анализа результатов тестования, а также генерации отчетов.

В документации[3] присутствует описание настройки ключевых компонентов для создания своей собственной инфраструктуры, при необходимости.

Использованные источники

1. <https://apertis.org>
2. <https://ostree.readthedocs.io>
3. <https://designs.apertis.org>
4. <https://wiki.apertis.org>
5. <https://gitlab.apertis.org/appfw/apertis-update-manager>
6. Пынькин Д.А., OSTree – атомарные обновления ОС в стиле git, 2018, <https://lvee.org/ru/abstracts/289>
7. <https://github.com/go-debos/debos>
8. Пынькин Д.А., Debos -- еще одна утилита для создания ОС, 2018, <https://lvee.org/en/abstracts/263>
9. <https://github.com/go-debos/fakemachine>
9. <https://about.gitlab.com/>
10. <https://phacility.com/phabricator/>
11. <https://jenkins.io>
12. <https://openbuildservice.org/>
13. <https://www.lavasoftware.org/>
14. <https://qa.apertis.org/>
15. <https://gitlab.apertis.org/tests/apertis-test-cases>