

# Creating custom Debian images for your embedded device

Embedded Recipes  
May 2022, Paris

Christopher Obbard  
twitter: @obbardc  
e-mail: [chris.obbard@collabora.com](mailto:chris.obbard@collabora.com)

# Who am I?

- Building Debian-based systems since ~2010
  - <https://lwn.net/Articles/484477/>
- Electronics Engineer, Embedded devices/IoT
- Senior Engineer at Collabora
- Integration: image creation, packaging, maintenance
- Board farm & automation
- Device bring-up



# Agenda

- Why use Debian over Buildroot/Yocto?
- Introduction to Debos
  - build generic tarball of Debian user-space
  - demo: build disk images for rockpi-e
- Build custom Debian packages
  - bootloader and kernel
- Plans for Debos
- Questions





# Why use distro as base?

# Buildroot/Yocto?

- Buildroot
  - cross-compile the world into a generic image which “just works”
  - simple, familiar and powerful (one repo! Kconfig! Makefiles!)
- Yocto
  - rolls your own distro from lots of recipes, easy to trip up when modifying things
  - can be complex, many moving parts to keep track of
- Distro (e.g. Debian/Fedora/Arch)
  - collection of prebuilt packages
  - not typically used for embedded systems...



# Distro (e.g. Debian) Pros

- Collection of prebuilt packages, updated often
  - e.g. debug or tracing tooling with apt-get
- Security updates
- Many architectures
- Less components to maintain yourself
- Development environment similar to final system
- Packages have most generic features enabled
- Community decide the direction
- Bug tracker, patches welcome

# Distro (e.g. Debian) Cons

- Harder to customise - “desktop” assumptions
- Steep learning curve
- Process (e.g. bugs) complicated
- Default kernel/bootloader/toolchain is more generic
- Not many examples for Embedded/IoT
- Rely on community support
  - <https://www.debian.org/consultants/>



# Raspbian/Armbian

- Debian base packages with custom kernel/bootloader
- Setup/customisation scripts
- Package manager still works
- Large disk image
- General purpose images for tinkerers: ROFS, Verity, Signing, Image upgrades ...
- Not ready for the field



# Apertis

- Debian derivative for in-car multimedia, IoT
- No GPLv3 code: patched packages, utils instead of coreutils
- Software BoM
- Predictable release dates & backporting policies
- Follows Linux LTS
- System upgrades: ostree and A/B upgrades
- User-space components: Flatpak application manager, update manager
- Reference hardware images tested regularly



# Summary

- Using a distro lets you focus on the most important part of your development:  
YOUR application!
- No single good solution
- All solutions require:
  - learning new tooling
  - maintenance
  - process change?



COLLABORA

# Debos

# Introducing Debos!

- Created originally to build Raspberry Pi Debian images
- Now used for Apertis
- Golang, Apache-2.0 license
- Parses YAML recipes into actions
- Fakemachine library creates VM
- Packaged in Debian, Arch, Dockerhub
- <https://github.com/go-debos/debos> (bugs, PRs welcome!)

# What is Debos?

- Recipe written in YAML
- Preprocessed with Go templating (variables, control flow)
- Recipe is an array of actions
- Action has one specific function
  - Debootstrap, Apt, Download, ImagePartition, Raw, Overlay, Pack, Unpack, Run, Ostream
  - Recipe Action (recursion?)
- Actions are written in Go

# Why not just run shell scripts?

- <https://github.com/drtyhlpr/rpi23-gen-image>
- Mounting images
- Root permissions
- Clean environment
- Fakemachine creates a vm (KVM or User-Mode Linux) and attaches disks/mounts parts of host filesystem
- Reuses kernel & programs from your system (docker?)

# Fakemachine

- <https://github.com/go-debos/fakemachine>
- For when a docker container will not do
- Go library and standalone executable
- Create a VM based off the running system - /bin
- Mount volumes from host
- Mount images as disk drives



# Who is using Debos?

- **Mobian** Debian for Mobile Phones
- **Apertis** Debian Derivative for Multimedia/IoT
- **Radxa** Development Board vendor
- **Meta** Debian images for resctl-demo benchmark tool
- You?





# What is Debos not useful for?

- Building Debian packages (OBS)
- Hosting Debian repositories (reprepro, aptly)
  
- ...it just creates images!



# Why not use X tool?

- Many other Debian image building tools
- Other tools focus on specific task
- Debos is a toolchain which can be used for many different tasks



# Debos development stages

- One simple image pipeline for (most?) cases:
  - Early device bring-up
  - Application development, driver development
  - Production: in-field upgrades, security & signing, security fixes
  - Further: cost-down variant, new features
- Store your recipe in one git repository (no more meta-\*, no more repo pull)
- Integrates with CI (GitLab, GitHub)
- Lava/Labgrid pre-merge/nightly image testing



# Recipe structure: ospack/rootfs

- Output is a rootfs tarball of a pre-configured system
- Include:
  - Packages required
  - System configuration
  - Install your application
- Supposed to be generic
- One tarball per supported architecture
- Use debootstrap, apt, run, pack actions



# Recipe structure: hwpack/image

- Unpacks generic rootfs tarball into a hardware-specific image
- Partition image
- Install kernel, bootloader
- Can have multiple hwpacks per project
- Transform kernel and dtb into boot.img
- Sign binaries
- Create upgrade image



COLLABORA

# Demo

<https://gitlab.collabora.com/obbardc/debos-recipes>

# Build your own packages

- Debian packaging is confusing. [New Maintainer's Guide](#)
- `dh_make`
- Store sources in git and use `git-buildpackage`
- Import source from Debian into git using `import-debian-package` helper
- Build in OBS, upload using [GitLab pipeline](#)
- OBS handles downloading dependencies, rebuilds, hosting repository



# Custom kernel

- No package: Build kernel manually and copy binaries into image
- Fork Debian package & add patches/config on-top
- Fork Debian package & use own original tarball







# Demo 2

[https://gitlab.collabora.com/obbardc/debian\\_u-boot](https://gitlab.collabora.com/obbardc/debian_u-boot)

[https://gitlab.collabora.com/obbardc/debian\\_linux](https://gitlab.collabora.com/obbardc/debian_linux)



COLLABORA

# Debos future

# Recent changes

- Go modules (library security fixes!)
- Unit tests & test recipes in CI
- Build docker images on GitHub
- Bug squashing

# Next steps

- Release v1.1.0 (nearly!)
- Merge Arch image building (pacstrap, pacman actions)
- Fedora actions?
- Rewrite some Debian shell scripts for parallelism?
- Improve test coverage
- Improve documentation & test recipes
- Bugs! Maintenance!
- Invest time into the tool



**Thank you!**



COLLABORA

**Open First**



**We are hiring**  
**[col.la/careers](https://col.la/careers)**



COLLABORA

**Open First**