

approaches to building a distribution: OpenHarmony and OpenMandriva

Bernhard "bero" Rosenkränzer

<bero@lindev.ch>

FOSDEM 2021 -- February 6, 2021

MY CONTACT:

bernhard.rosenkraenzer.ext@huawei.com,
bero@lindev.ch

LINKEDIN:

<https://www.linkedin.com/in/berolinux/>

Bernhard "bero" Rosenkränzer

I don't usually do "About Me", but since it
may be relevant to the topic:

Principal Technologist at Open Source
Technology Center since November 2020

President of the OpenMandriva Association,
Contributor since 2012 - also a contributor to
Mandrake back in 1998/1999

Way more important
than that, he also feeds
dogs.



What is OpenHarmony?

- More than an operating system: Can use multiple different kernels (Linux, Zephyr, ...)
- Key goal: autonomous, cooperative devices -- multiple devices form a distributed virtual bus and can share resources
- Initial target devices: Avenger 96 (32-bit ARMv7 Cortex-A7+-M4), Nitrogen 96 (Cortex-M4)
- Built with OpenEmbedded/Yocto - one command builds the entire OS
- Fully open, developed as an Open Source project instead of an inhouse product from the start.
- For more information, visit [Stefan Schmidt's talk in the Embedded devroom, 17.30](#) and/or talk to us at the Huawei OSTC stand.

What is OpenMandriva?

- A more traditional Linux distribution - controlled by the community, continuing where Mandriva left off after the company behind it went out of business in 2012. Its roots go back to the first Mandrake Linux release in 1998.
- Originally targeting only x86 PCs - Support for additional architectures (aarch64, armv7hnl, RISC-V) added later
- Repositories contain 17618 packages, built and updated individually, assembled into an installable product with `omdv-build-iso` or [`os-image-builder`](#).
- For more information, visit the OpenMandriva stand.



Key fact about this distro:
I'm its official mascot!

How OpenHarmony is built

- Download the OS source:

```
mkdir ohos; cd ohos
```

```
repo init -u https://git.ostc-eu.org/OSTC/OHOS/manifest.git
```

```
repo sync
```

At this point, you have what is needed to build OpenHarmony: bitbake recipes, board description files and configuration files describing how to build everything from source.

How OpenHarmony is built

- Initialize the environment

```
cd poky
```

```
. oe-init-build-env
```

```
bitbake-layers add-layer ../meta-openembedded/meta-ohos
```

```
bitbake-layers add-layer ../meta-openembedded/meta-python
```

Now the build system is set up with what's needed.

- `DISTRO=poky MACHINE=qemux86-64 bitbake core-image-minimal`

This takes care of everything else: Source code for the actual components is downloaded (unless it's there from a previous build), built according to what's in the bitbake recipes, and assembled into an image that works for the \$MACHINE target.

How OpenMandriva is built

- Every upstream package (library/application) is packaged in a separate rpm package
- The package is sent to ABF (Advanced Build Farm), which sends it to builders for all supported architectures (at this time: x86_64, znver1, aarch64, armv7hnl, riscv64, i686. ppc64 may be added in the future, we got access to a builder...).

Builds are done as native builds, on a machine matching the architecture (or, in the case of RISC-V, in qemu on faster machines).

If the build succeeded everywhere, the package is added to the repositories. If necessary, packages depending on the package are rebuilt.

How OpenMandriva is built

- A script assembles the packages and builds an ISO image (traditionally) or, more recently (with the advent of non-x86 support, more on that later) whatever format a specific target needs.

Why build from source in one go? (OpenHarmony)

- Crosscompiling built into the OS builder by design (you DON'T want to build an OS on a Cortex-M4 powered device!)
- Makes it easier to build on top of different kernels (Linux/Zephyr/...) with different requirements (not every RTOS supports shared libraries)
- Makes it easier to work with boards that need custom kernels, custom bootloaders etc.
- No possibility of problems caused by ABI changes in a library - everything is guaranteed to be built against the library version actually in use
- Makes the use of license compliance tools like REUSE easier

I like this approach. We sometimes get to take a walk after our 2-legged pet types `"bitbake core-image-minimal"`.



Why use binary packages? (OpenMandriva)

- Building all 17618 packages (some of them big -- LibreOffice and all) takes multiple days even on modern hardware. Try doing it in 1998.
- For a larger system, where not everyone is expected to update at the same time, updating individual packages is often preferable over an OTA approach
- On a larger system, not every user wants to use the same set of packages - e.g. few people need KDE Plasma and GNOME at the same time
- Supports packages that can't be crosscompiled (sadly, they still exist)

How OpenHarmony is updated

- Details are still being worked out, but it is all but certain that:
- Updates will come as an OTA image, updating the entire OS in one go
- Updates will be some variant of a filesystem image, or a delta to a previous filesystem image
- Updates are sent to users when a new version has been completed and tested.

How OpenMandriva is updated

- Every package is updated individually
- On the "cooker" (development) and "rolling" (rolling release), people usually receive updated packages every day
- Users of the "rock" (stable) tree get only tested packages, only once in a while

Why do OTA updates (OpenHarmony)?

- Guaranteed consistent system - people will always get a build that has been tested in its entirety (no possibility of e.g. a user getting an updated library without a corresponding application rebuild against the new library)
- No need for temporary storage (updated packages being downloaded on running old system) on systems that are extremely short on storage (a Nitrogen 96 board has 512 kB of storage, and is not necessarily the smallest target for OpenHarmony)
- It's what people have come to expect in the consumer device space

Why do package updates (OpenMandriva)?

- Deal with the fact that not every system has the same packages installed - no point in sending a GTK update to a purist Qt user or vice versa
- Give users control over what exactly they're updating, allow users to blacklist an individual update
- It's what people have come to expect in a Linux desktop/server



I can't wait for that
dogfood-2021.02.07-1
update package!

Workarounds for deficiencies in either system

- When starting to target non-x86 devices, OpenMandriva ran into 2 problems:
 - There is no defined standard format in which to distribute images for non-PC devices (like ISO files for traditional PCs).
While aarch64 servers these days tend to support UEFI and ISO files, many interesting devices, from Raspberry Pi to Rock Pi to Pinebook Pro to PinePhone to SiFive Unleashed don't.
 - Many SOC's require a custom kernel with many patches that aren't in the tree yet. They typically touch files all over the kernel and conflict with patches needed for other SOC's. It is quite a challenge to build a kernel that can support all needed features of e.g. a current Qualcomm SOC, a current Mediatek SOC and a current Allwinner SOC at the same time.

Workarounds for deficiencies in either system

- Many other traditional Linux distributions decided to "solve" this problem by not supporting any devices that don't support UEFI.
But that's not an option for OpenMandriva - we still care about Linux on the desktop (and the Pinebook Pro is a great way to get there), and about phones that are not under the control of either Google or Apple.
- So... Learn from OpenHarmony and rebuild the distribution the same way?

PARTIALLY!

- For the reasons mentioned before, OpenMandriva can't switch to a fully "build the distribution from source all the time" mode. But the new os-image-builder takes some of its ideas from OpenHarmony/OpenEmbedded.
 - Per-device config files indicate which extra packages should be included for the target device
 - It can also specify a custom kernel repository, a custom kernel config, a DeviceTree file name, a custom u-boot repository, etc.

os-image-builder

- **Parts of the device config for Pinebook Pro:**

```
ARCH="aarch64"
```

```
HWPACKAGES="kernel-firmware-extra"
```

```
HWPACKAGES_gui="lib64dri-drivers-panfrost"
```

```
KERNEL="https://github.com/megous/linux.git#orange-pi-5.11" # <-- orange-pi-5.11 is the branch
```

```
KERNEL_CONFIG="pinebook_pro_defconfig"
```

```
KERNEL_EXTRACONFIG="--enable NVM --module BINFORMAT_MISC"
```

```
DTB="rockchip/rk3399-pinebook-pro"
```

```
UBOOT="https://megous.com/git/u-boot"
```

```
UBOOT_CONFIG="pinebook-pro-rk3399_defconfig"
```

os-image-builder

- os-image-builder also allows the per-device config to replace parts of the build script, so any requirements like "copy u-boot to sector 1234 on the SD-card" or "create a FAT16 formatted MBR partition number 4 containing the kernel" can be accomodated.
- At build time, it builds rpm packages for the custom kernel (if no custom kernel is specified, the normal distribution kernel is used), and builds the custom u-boot (if specified) - everything else is still pulled from the binary package repositories
- The scripts then output the install image for the right target device in the expected format, much like bitbake does in OpenHarmony

Will OpenHarmony run into limits with the other approach?

- Yes.

As soon as we add a bigger target device that will handle custom application choice, we'll need some way to deal with packages - that could be something similar to DNF or Apt (possibly opkg), or something closer to the AppGallery that exists on de-Googleified Huawei phones (if its licensing can be adjusted).

Or something entirely new we haven't thought of yet. We'll deal with the problem when we get there.

Thank You!

... And get in touch with us

Both projects talked about can use an extra hand!

VISIT OUR GITLAB REPO:

<https://git.ostc-eu.org/OSTC>

**OUR WEBSITE WILL BE LIVE SOON, START
ADDING IT TO YOUR PREFERRED WEBSITES AND**



KEEP POSTED:

www.ostc-eu.org

Visit OpenMandriva at <https://openmandriva.org/>