

Using GNOME OS on real hardware

Valentin David



GUADEC 2020

July 22nd

Table of Contents

- 1 What is GNOME OS?
- 2 ARM 64
- 3 x86_64
- 4 Experience using GNOME OS
- 5 How to get it

What is GNOME OS?

Bootable image deliverable of GNOME releases and continuous builds

- Application developers: test their applications before release
- Release process itself
- UX testing
- Hardware testing

With some ideas going further

Preinstalled on hardware. *A bright future for GNOME* Juan José Sánchez and Xan López (GUADEC 2012)

GNOME is not a platform until it is not also an OS. See *There is No “Linux” Platform* Tobias Bernard and Jordan Petridis (LAS 2019)

First implementation: GNOME Continuous

- Initiated by Colin Walters
- Both a tool to build, and the manifest describing the modules
- Based on top of an image built with Yocto
- Updates with OSTree (atomic updates). Nice!

Many manifests, many builds

- Release using JHBuild
- Flatpak's GNOME SDK built with Flatpak Builder
- GNOME Continuous

Replaced by BuildStream and GNOME Build Metadata

- *GNOME Build Strategies and BuildStream* Tristan Van Berkom (GUADEC 2017)
- *Migrating from JHBuild to BuildStream* Michael Catanzaro (GUADEC 2018)
- *GNOME and Buildstream, two (three?) years later* Abderrahim Kitouni (2020), see him on Friday 24th, 15:30 UTC

What is in GNOME OS?

- Boots on UEFI using systemd-boot
- Initramfs is generated with dracut
- Plymouth (graphical boot) with bgrt theme.
- Systemd init (also in initramfs)
- GNOME core
- Flatpak for the other applications
- Wayland + XWayland
- Mesa drivers
- Starts with GNOME Initial Setup
- Root partition resized at first boot
- OSTree atomic update, with eos-udpater and GNOME Software

Table of Contents

- 1 What is GNOME OS?
- 2 ARM 64**
- 3 x86_64
- 4 Experience using GNOME OS
- 5 How to get it

ARM boards tested

- Pinebook Pro (laptop)
- Rock 64
- Raspberry Pi 4 (by Tom Pollard)

UEFI on ARM?

Booting ARM boards is awful. Weird bootloaders.

Rarely support UEFI out of the box. But some do.

When EDK II is not ported, U-Boot can be used to boot UEFI.

Bootloader is firmware, and is not part of OSTree.

- Each board has a different initial image
- However the OSTree repository can be the same

The device tree mess

- Device trees come with the kernel. They are specific to hardware.
- ACPI/SMBIOS comes from hardware on x86_64
- OSTree does not really properly support
- Thankfully UEFI provides device tree
- How to update the device tree? fwupd?

Non upstream kernels

Pinebook Pro and Raspberry Pi 4: kernel not upstreamed

Different kernel \implies different OSTree

We should not support board with no upstream Linux support

How well does it run?

- Range of ARM products is wide. From microcontrollers to powerful workstations
- Pinebook Pro (laptop), Rock 64, Raspberry Pi 4 are cheap. But runs well.
- Raspberry Pi 3 A had too little memory: 512MB. Most of it went to video memory
- All had some video acceleration

Table of Contents

- 1 What is GNOME OS?
- 2 ARM 64
- 3 x86_64**
- 4 Experience using GNOME OS
- 5 How to get it

Running all my desktops/laptops on GNOME OS

I do not use those aarch64 devices everyday. So...



Installer on ISO

First, we need an installer

- An ISO booting image with Endless OS installer as a proof of concept
- Endless OS installer is based on GNOME Initial Setup

Note: not merged yet

Table of Contents

- 1 What is GNOME OS?
- 2 ARM 64
- 3 x86_64
- 4 Experience using GNOME OS**
- 5 How to get it

Does GNOME core work?

Most core applications worked.

Fixed

- Builder: needed Flatpak Builder
- Boxes: libvirt was missing many dependencies
- Photos, Music: Tracker broken

Still broken

- Orca not working (does it work on Wayland?)
- Web and Videos are missing h264 codec
- Web missing microphone/webcam support

Other things that were fixed

Power management: was missing integration

Printing: was missing lots of dependencies. Only tested postscript printers. Please test.

Flatpak portals for GTK+ were missing

Hardware:

- bluetooth
- wifi
- sound, microphones
- touchscreens, touchpads
- webcams

To develop GNOME OS on GNOME OS, I needed BuildStream.

So BuildStream on Flatpak? Yes. But bad user experience.

objdump? gdb? valgrind? Flatpak too.

git? flatpak-builder? Dependencies of GNOME Builder.

Solution: dual trees

Two trees:

- For users (Core minus GNOME Builder)
- GNOME Builder + SDK (as in Flatpak)

(Like GNOME Continuous “runtime” and “debug-devel”)

What about debuginfo? We should use debuginfod.

Steam devices udev files, Gamemode

VR works, but Wayland does not yet have drm lease.

Still missing fan and voltage control, cpu frequency, and sensor monitoring

I have used AMD and Intel graphics. What about NVidia proprietary drivers?

- We use glvnd
- How to deploy and EGL/vulkan/opengl backends?

How did I make slides?

Built TeXLive with BuildStream and run in a BuildStream shell.

Maybe using podman, or other OCI container system would have been easier.

One last thing

CCID support: sign and login with a hardware key

Table of Contents

- 1 What is GNOME OS?
- 2 ARM 64
- 3 x86_64
- 4 Experience using GNOME OS
- 5 How to get it

Should I use it?

Still in development, and needs documentation.

As a virtual machine for testing: yes, soon.

On hardware? Not yet. Unless your point is to contribute into making it work on hardware.

How to get it

OSTree repository begin set up

Master build available on GNOME Build Metadata

<https://gitlab.gnome.org/GNOME/gnome-build-meta>

Ask #gnome-os on GIMPNet

Aknowledgements

Special thanks to Abderrahim Kitouni and the rest of the release team.

This work was sponsored by:

