Zagreb, FER, DORS/CLUC 2022

How we developed a custom build platform for the Enterprise

Jakov Petrina

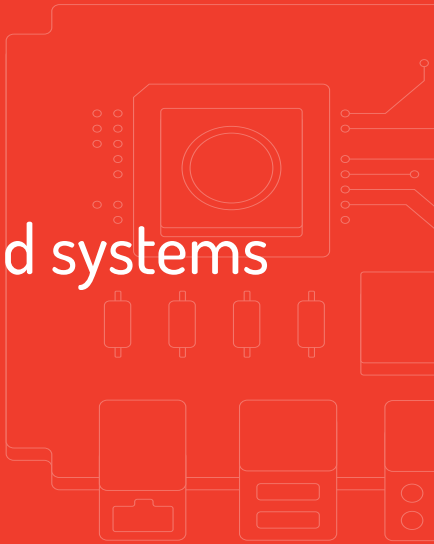
sartura

May 20th, 2022

About

- Leading Replica.one development team
- Embedded Linux development and integration
- Continuous participation in Open Source projects
  - Upstream contributions for Gentoo, OpenWrt, the Linux kernel, systemd, etc.

Introduction to Linux build systems



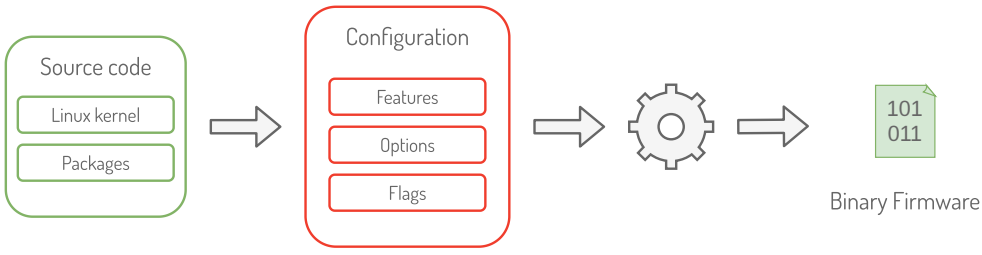
sartura

Linux on embedded systems

- We are mostly focused on network embedded (e.g. APs, CPEs, routers, switches)
- Architecture-specific considerations (e.g. x86, ARM, MIPS)
  - Cross-compilation to target ISA from other platforms
  - Booting process and bootloaders
  - Describing or discovering hardware (e.g `devicetree` vs. ACPI)

Build systems for embedded Linux

- What do Linux build systems do?
  1. Generate toolchain for cross-compilation
  2. Fetch, prepare, configure, build Linux kernel and user-space packages
  3. Integrate results into a ”firmware” image
- Examples of existing open source build tools
  - **Buildroot** — a set of makefiles
  - **OpenEmbedded / BitBake** — inspired by Gentoo’s Portage
  - **Open Network Linux (ONL) build system**


Source code
Linux kernel
Packages
Configuration
Features
Options
Flags
101
011
Binary Firmware

Obvious requirements

- Open Source and Linux-based
  - Consume benefits from the *community* activity
- Flexible configuration and build system
  - Extending base features with *value added* improvements
- Active project and community
  - Guarantee for *future* success

Overlooked aspects

- Championed by the community
  - Upstream *maintained* and *documented*
  - Downstream *economical maintenance*
- Empowering by providing a number of *alternative* software packages
  - Leverage *technology immunity*

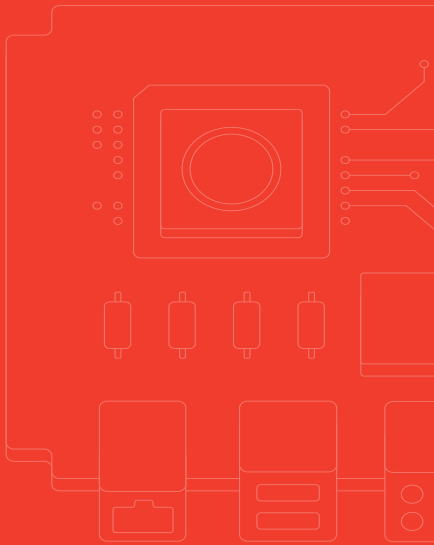
Challenges with existing platforms

- Limited scope of targeted devices
  - e.g. no support for bare-metal switches
- Non-standard or purpose-built solutions
  - e.g. system libraries and/or configuration frameworks
- Maintenance and/or integration burden
  - e.g. large amount of out-of-tree patches
  - e.g. distro-specific compile-time defaults

Our ideals

- A single, unifying platform for a wide range of devices
- Mainline or at least soon-to-be mainline platform support
- High degree of flexibility and integrability
- Manageable supply chain inspection

Replica.one



sartura

Replica.one

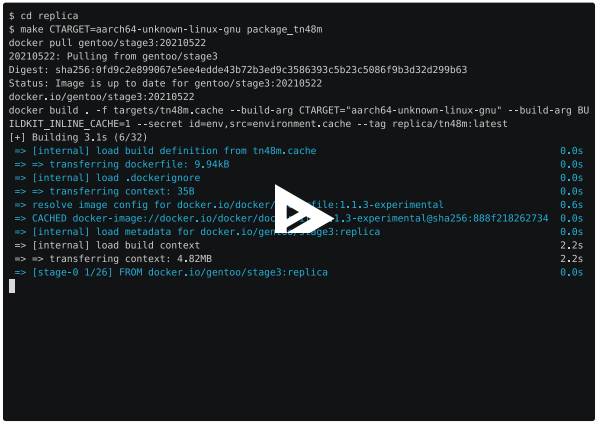
- An open source Gentoo-based firmware build system
  - Highly customized, stripped-down Gentoo based firmware builder
  - A unified stack for wide range of devices (e.g. APs, routers, switches, servers)
  - Leveraging latest advancements from upstream (Linux kernel, U-Boot, etc.)
- *Like LEGO™ bricks, modular and maintenance free*

Architecture overview

- Platform-agnostic firmware for *Enterprise* applications
- Builds are driven by top-level Makefile, GNU m4, and Docker definitions
- Toolchain generation using Gentoo `crossdev` mechanism
- Tight integration with Gentoo repositories and profiles
- Target-specific packaging process

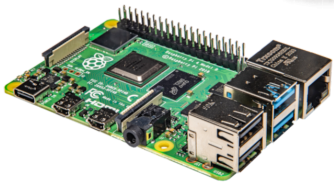
Best practices

- How do we solve the maintenance issue?
  - In-house board-bringup, driver support
  - Continuous upstreaming effort in coordination with the community
- How do we ensure a flexible, but easy-to-integrate system?
  - Make use of the Gentoo's package manager — **Portage**
  - Leverage **Gentoo** community resources
- How do we target wide range of devices?
  - Mechanism vs. policy


$ cd replica
$ make CTARGET=aarch64-unknown-linux-gnu package_tn48m
docker pull gentoo/stage3:20210522
20210522: Pulling from gentoo/stage3
Digest: sha256:0fd9c2e899067e5ee4edde43b72b3ed9c3586393c5b23c5086f9b3d32d299b63
Status: Image is up to date for gentoo/stage3:20210522
docker.io/gentoo/stage3:20210522
docker build . -f targets/tn48m.cache --build-arg CTARGET="aarch64-unknown-linux-gnu" --build-arg BU
ILDKIT_INLINE_CACHE=1 --secret id=env,src=environment.cache --tag replica/tn48m:latest
[+] Building 3.1s (6/32)
=> [internal] load build definition from tn48m.cache 0.0s
=> => transferring dockerfile: 9.94kB 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 35B 0.0s
=> resolve image config for docker.io/docker/ file:1.1.3-experimental 0.6s
=> CACHED docker-image://docker.io/docker/do 1.3-experimental@sha256:888f218262734 0.0s
=> [internal] load metadata for docker.io/ge age3:replica 0.0s
=> [internal] load build context 2.2s
=> => transferring context: 4.82MB 2.2s
=> [stage-0 1/26] FROM docker.io/gentoo/stage3:replica 0.0s

Already in production

- WiFi AP and CPE
- Switch
- Core router
- Server and Compute infrastructure
- Service isolation
  - Containerization (LXC, LXD, systemd-nspawn, Docker)
  - Kernel-based Virtual Machine (KVM)




IEI
Puzzle 00

Takeaways & Highlights

- Virtually no maintenance requirements
- Aligned with the upstream communities
- Lightweight system fulfilling multiple use cases
- Captures business of today and unlocks the service market of tomorrow

Zagreb, FER, DORS/CLUC 2022

How we developed a custom build platform for the Enterprise

jakov.petrina@sartura.hr

info@sartura.hr · www.sartura.hr