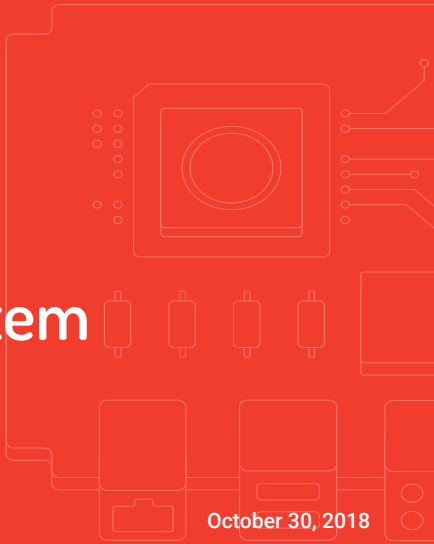
Lisbon, OpenWrt Summit 2018

Porting any service to any Linux-based OS using OpenWrt Build System

Marko Ratkaj

sartura



October 30, 2018

Why would you do that?

- Replacing the original firmware is not an option
- You want your service to be
  - OS-independent
  - Available for deployment on wide range of devices

Our case:

- No original SDK
- No hardware specification
- No information about software running on the device
- No OpenWrt support
- Read-only file system
- Access to shell

Lost cause?

Assumptions

- Must keep the currently running kernel
- Must not reuse libraries already on the file system
- Must install all libraries and dependencies under a non-standard path or use static binaries
  - Static vs. dynamic
- Picking the least invasive route

Procedure

1. Understanding the system
2. Generating the toolchain
3. Static or dynamic approach
4. Integrating with the rest of the system

1. Understanding the system

- Architecture
- Free storage, free memory
- Kernel configuration
- Available tools
- Running services

- Useful tools:
  - uname, df, free, file, readelf, objdump, ...
- Useful files:
  - `/proc/cpuinfo`
  - `/proc/meminfo`
  - `/proc/mtd`
  - `/proc/config.gz`
  - ...

```
$ file busybox
busybox: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV),
    dynamically linked, interpreter /lib/ld-uClibc.so.0, stripped
```

toolchain hint

```
$ readelf --arch-specific busybox
Attribute Section: aeabi
File Attributes
  Tag_CPU_name: "7-A"
  Tag_CPU_arch: v7
  Tag_CPU_arch_profile: Application
  Tag_ARM_ISA_use: Yes
  Tag_THUMB_ISA_use: Thumb-2
  Tag_FP_arch: VFPv2
  Tag_ABI_PCS_wchar_t: 4
  Tag_ABI_FP_denormal: Needed
  Tag_ABI_FP_exceptions: Needed
```

-march=armv7-a

hard float

2. Generating the toolchain

- Adding a simplified new target to OpenWrt
  - No need to work with image, go with ”tar.gz”
  - No need to work with target-specific patches
- Files of interest:
  - `./target/linux/<mytarget>/Makefile`
  - `./include/target.mk`

3. Static or dynamic approach

- Static
  - musl is awesome
- Dynamic
  - Standard locations (Host OS files)
    - `/lib`, `/usr/lib`
  - Non-standard locations (e.g. usb mount point)
    - `/mnt/usb1_1/lib`, `/mnt/usb1_1/usr/lib`
    - elf magic needed (patchelf)

Dynamic approach

- Non-standard locations
  - patchelf to set interpreter location
  - patchelf or exporting `LD_LIBRARY_PATH` to set non-standard library search paths
- How does OpenWrt handle binaries?

4. Integrating with the rest of the system

- Init systems
  - Different init systems (busybox init, procd, systemd, etc.)
  - Sartura - packaged a lightweight Open Source process monitor to start and monitor our services
- Firewall
  - Focus must be placed not to interfere with other services on the host (e.g. IPTV, VoIP, etc.)

Real world example

Miyagi - remote management system

- www.miyagi.io

Porting any service to any Linux-based OS using OpenWrt Build System

marko.ratkaj@sartura.hr · info@sartura.hr · www.sartura.hr