SYNOPSYS®
Silicon to Software™

OpenWrt

The most affordable embedded Linux distribution

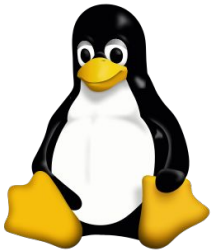
Alexey Brodkin

LinuxPiter 2016

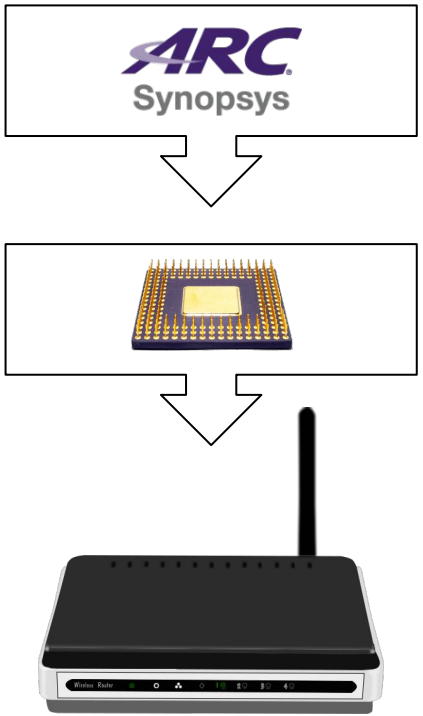
Contents

About the author

- Software engineer at Synopsys, St-Petersburg
- ~10 years experience in embedded software development
- Things I do:
  - Linux kernel
  - U-Boot bootloader
  - Buildroot
  - uClibc
  - OpenWrt



Why talking about OpenWrt?



ARC 700,
ARC HS38 cores

Custom SoCs

Wireless
routers

What is OpenWRT

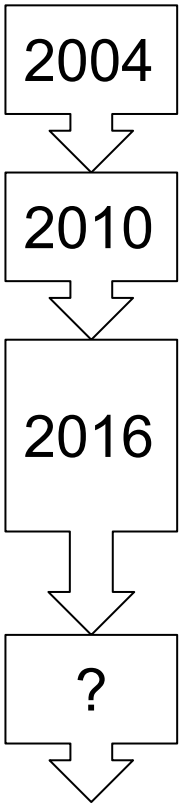
"OpenWrt is an embedded operating system based on the Linux kernel, primarily used on embedded devices to route network traffic. The main components are the Linux kernel, util-linux, uClibc or musl, and BusyBox"

says https://en.wikipedia.org/wiki/OpenWrt

Features:

- Supports 10+ CPU architectures, 40+ platforms, 1000+ devices
- Non resource hungry
- Package-based
- Flexible

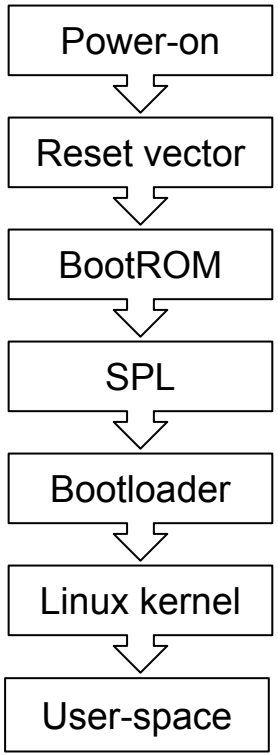
Historical overview



- Started in January 2004 based on Linksys GPL sources for WRT54G and Buildroot.
- First companies (Atheros and Netgear) start to use OpenWrt in their products.
- Broadcom, Qualcomm Atheros, Mediatek, Belkin, Linksys, Freescale/NXP, Marvell, etc use OpenWrt as a base for their firmwares and SDKs.
- LEDE got forked in May.
- LEDE merges with OpenWrt.
- New cool releases.

OpenWrt version	source packages	kernel version
0.9 "White Russian"	~150*	2.4.30
8.09 "Kamikaze"	~875*	2.6.26
12.09 "Attitude Adjustment"	~1150*	3.3
15.05 "Chaos Calmer"	~770**	3.18.y
current master	~980**	4.4.y

Boot process of a device running OpenWrt



0.1-1 kB,	RO,	ROM
1-10 kB,	RW,	Flash/SD/HDD
10-1000 kB,	RW,	Flash/SD/HDD
1-10 MB,	RW,	Flash/SD/HDD/USB/Network

Boot process of a device running OpenWrt (cont')

BootROM (Pre-bootloader or Primary Program Loader)

- Resides in ROM (built in silicon)
- Read-only (couldn't be changed)
- Extremely small and simple (hundreds of bytes)
- Loads SPL from flash or removable storage

Boot process of a device running OpenWrt (cont')

SPL (Secondary program loader)

- Larger than BootROM (Couple of kB)
- Could be modified
- Usually resides in flash or on removable storage
- Initializes hardware (caches, DDR, clocks)
- May load full-featured bootloader from more sources: Flash/SD/USB/Network/HDD

Boot process of a device running OpenWrt (cont')

Bootloader (usually GRUB on x86, U-Boot for others)

- Hundreds of kB
- Could be changed
- Usually resides in local flash or on removable storage
- Powerful: own scripting language; tons of peripherals; console
- Does more initialization in preparation of running Linux:
  - Memory
  - SMP
  - Video
  - .dtb

Boot process of a device running OpenWrt (cont'd)

Linux kernel

- Megabytes of size
- Obviously could be changed
- May reside pretty much everywhere: local storage, network, etc.
- Supports miriads of devices
- Initializes CPU, SoC, required drivers
- Calls "init" in the end of boot process

Boot process of a device running OpenWrt (cont'd)

User-space

- init
- Board detection scripts
  - Read device tree "model" property

    ```
    model="$( cat /proc/device-tree/model )"
    ```
  - Parse /proc/cpuinfo

    ```
    model=$(awk 'BEGIN{FS="[ \t]+:[ \t]"} /model/ {print $2}' /proc/cpuinfo)
    ```
  - Read "magic number" from flash

    ```
    dd if=$part bs=4 count=1 skip=16 2>/dev/null | hexdump -v -n 4 -e '1/1 "%02x"'
    ```
- First time config
  - Run board-specific scripts
  - Autodetect Wi-Fi radio
- Auto-load kernel modules
- Autostart daemons: DHCP, HTTP, SSH
- Interactive sessions

What needs to be built

For target

1. Bootloader image
2. Linux kernel [with initramfs] image
3. [Root]filesystem
4. Additional packages
5. First-time install, update package or SD-card image

For host

1. Host helper tools
2. Cross-toolchain
3. SDK

OpenWrt Build System

- Heavily-patched Buildroot-based build system
- Kconfig for configuration (but no per-board defconfigs)
- [Complicated] Makefiles for building
  - Variable targets like:

```
$(BIN_DIR)/$$(KERNEL_IMAGE): $$(KDIR_KERNEL_IMAGE)
```

  - Extensive use of make's "call" function like:

```
$(call Device/Build/kernel,$(1))
```

  - Want to combine? You've got it:

```
$(KDIR)/tmp/$(call IMAGE_NAME,$(1),$(2)): $$(KDIR_KERNEL_IMAGE)
$$(ROOTFS/$(1)/$(3))
	@rm -f $$@
	[ -f $$(word 1,$$^) -a -f $$(word 2,$$^) ]
	$$(call concat_cmd,$(if
$(IMAGE/$(2)/$(1)),$(IMAGE/$(2)/$(1)),$(IMAGE/$(2))))
```


How OpenWrt gets built and installed on the target

1. Select configuration to build for
   `make menuconfig -> select SoC -> board -> fine tune`
2. Get sources
   `make download`
3. Build (host tools and target binaries)
   `make`
   a. Extract archive with sources
   b. Patch sources
   c. Configure
   d. Build
   e. Install (to host tools, staging and target folders)
4. Create target image
   a. Bootloader
   b. Linux kernel
   c. Filesystem
   d. Packages (.opkg files to be installed on target later on)

Building Linux kernel

Problem:

Build kernels for 1000+ devices

Solution:

Don't build for each device

Really?

Build kernels for SoCs/platforms instead of devices/boards

- Use external Device Tree blobs
- Build-in .dtb as a post-processing
- Do required fix-ups in runtime

Building Linux kernel (cont'd)

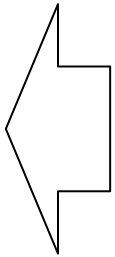
Problem

Not everything is in upstream kernel

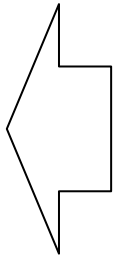
Solution

Use off-the-tree patches and configurations

```
ls -1 target/linux/generic/
...
config-3.18
config-4.1
config-4.4
patches-3.18
patches-4.1
patches-4.4
```



```
ls -1 target/linux/ramips/
...
dts
image
Makefile
modules.mk
mt7620
mt7621
...
patches-4.4
rt288x
...
```



```
ls -1 target/linux/ramips/mt7620/
config-4.4
profiles
target.mk
```


Off-the-tree patches for Linux kernel

Patches are well-categorized with prefixes.
235 patches applied to all 4.4 kernels and among them:

1. **63 backported from upstream**
2. **11 awaiting upstream merge**
3. 15 architecture-specific
4. 39 MTD and filesystem
5. 48 network

What makes OpenWrt special

- Vendor-independent Linux distribution
- Small memory footprint & low power requirements

  32+Mb RAM & Mb Flash; 200+ MHz CPU
- Support for 1000+ devices; some cost ~20$ and available worldwide
- Convenient for user: 1000+ packages, nice Web-UI
- Allows for infinite amount of customizations
- Support of "standard" Linux tools and features
- Forefront of wireless & networking development of Linux kernel

```
git log --author=openwrt --oneline | wc -l
2115
```


What makes OpenWrt not so special...

- Build system is not well-structured and learning curve is steep
- Still many off-the-tree patches to be upstreamed
- Core developers don't bother with writing verbose commit messages
- Unclear release cycle

How to get involved

- Check if your Wi-Fi router or SBC is already supported: https://wiki.openwrt.org/toh/start
- Get something cheap but supported from nearby mall
- Try stable release or snapshot on your device if there's an image for it
- Report issues via https://github.com/openwrt/openwrt/issues
- Get sources `git clone https://github.com/openwrt/openwrt.git` and build your own image
- Hack on sources and send patches to openwrt-devel@lists.openwrt.org

Interesting links

- https://lkml.org/lkml/2003/9/28/175 - Precursor of OpenWrt creation
- https://wiki.openwrt.org/inbox/snapshot - Filesystem snapshot feature of OpenWrt
- https://wiki.openwrt.org/doc/techref/procd#openwrt_operating_system_architecture - Comparison of components of common desktop Linux distro, OpenWrt & Android
- https://forum.openwrt.org/viewtopic.php?id=67204 - Building OpenWrt in Windows 10 (with help of "Windows Subsystem for Linux")
- https://wiki.openwrt.org/doc/techref/process.boot - The Boot Process
- https://docs.blackfin.uclinux.org/doku.php?id=bootloaders - Detailed BlackFin boot procedure explanation

SYNOPSYS®
Silicon to Software™

Thank You


SYNOPSYS®
Silicon to Software™