Fedora on RISC-V
Status update

Wei Fu 傅炜
Software Engineer @ Red Hat
Jun 30th 2018
RISC-V Workshop 2018 Shanghai
AGENDA

● Status: updates for Fedora on RISC-V
  ○ FYI: bootstrap for Fedora on RISC-V
  ○ History: First and second bootstraps
  ○ Status: Final bootstrap
  ○ Facility: Koji system/build farm
  ○ Demo: Fedora on Unleashed (and QEMU)
  ○ Firmware TODO list
  ○ Development Info

● Distro: Why Fedora First
  ○ Fedora/RHEL/CentOS
  ○ How to get RHEL support

● Goal: Industry standard RISC-V server
  ○ Learn from other architecture
  ○ Proposal draft

● Acknowledgments
Status:
updates for Fedora on RISC-V
Why native build, but not cross-compiling

The advantage of native build.

almost all the software are originally designed as native build. For cross-compiling, we not only have to port the software itself, but also need to create and maintain the compilation scripts.

Saving time & effort for making/maintaining Compilation scripts

some system software does not only have the main bin file, but also includes a lot of utilities. Some of them will run in the compiling stage, but some will run at runtime. This increases the Compilation scripts' complexity.

Keep It Simple, Stupid

Another reason for using cross-compiling is the Host's performance is much better than the target. But now, we have QEMU support, and Fedora is for 64-bit RISC-V which is powerful too, moreover we have Koji.

Target platform is powerful enough for compiling job.
What is Fedora bootstrap

**Chicken And Egg Situation**
Generally, one Fedora release is built upon the previous release. But this can **NOT** be done for a brand new architecture, because we don't have a "previous release" at that point.

**Breakout**
We must **cross-compile** enough software/packages to "**bootstrap**" the new architecture. This can be a tricky endeavor.
How Fedora bootstrap works

This project is designed to help you bootstrap Fedora to a brand new architecture.
### Fedora bootstrap on RV64 vs AArch64

<table>
<thead>
<tr>
<th>Build System</th>
<th>Stage</th>
<th>AArch64</th>
<th>Stage</th>
<th>RISC-V RV64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host (x86)</td>
<td>1</td>
<td>Simulator (ARM model)</td>
<td>1</td>
<td>QEMU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>a</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>b</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Hi-Five Unleashed</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Bootable Rootfs</td>
<td>2</td>
<td>GNU cross-compiler toolchain</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>a</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>b</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Berkley Bootloader</td>
</tr>
<tr>
<td>Target (Native)</td>
<td>3</td>
<td>rpmbuild</td>
<td>3</td>
<td>Linux kernel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>a</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>b</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>basic rootfs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cross-compile and install &quot;rpm&quot; packages &amp; dependencies</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>mock</td>
<td>4</td>
<td>install pre-build RPMs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>a</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>b</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>rebuild RPMs from SRPMs natively</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>install the new RPMs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>build stage4 image</td>
</tr>
</tbody>
</table>

RISC-V Autobuilder on QEMU

4. koji (builders) Distribution bootstrap
5. Rebuild in koji
6. koji-shadow

TBD
History: First bootstrap

October 15, 2016, The First bootstrap for Fedora 25 on RV64 was finished by: David Abdurachmanov, Richard WM Jones and Stef O'Rear.

DJ Delorie provided significant contributions to glibc for RISC-V.

This project built more than 5,000 packages. The kernel and image could boot on QEMU RV64.

Richard’s Blog: Fedora/RISC-V is finished!
Why did this project stop?

Since Fedora has an **upstream first policy**, and it also applies to Fedora/RISC-V.

Even though by **13 Nov 2017**, the upstream kernel merged in RISC-V:

[GIT PULL] RISC-V Port for Linux 4.15 v9, after the merging of **GCC** and **binutils** patchsets for RISC-V.

We were still waiting for

the glibc patchset (RISC-V glibc Port) to be merged,
then we can do the final bootstrap on RV64.
For the final bootstrap, the engineers did the second bootstrap on Dec 2017 for Fedora 27 as preparation!

Then once the expected changes for glibc were upstreamed, he could do the bootstrap again as the final one.
The final piece got merged!

On 29th Jan 2018, the “RISC-V glibc port v7” patchset was committed. As the final piece was upstreamed, then the final bootstrap on RV64 was prepared to run.

Patchset from mailing list: RISC-V glibc port, v7
Status: bootstrap is **DONE**

The Fedora bootstrap for RISC-V was finished in March 2018. On 2018 April 15th, we moved to koji, and now are building packages normally, just like all other architectures.

**Richard’s Blog:**
- Fedora/RISC-V: the final bootstrap
- Fedora/RISC-V: Runnable stage 4 disk images
- Fedora/RISC-V: ssh and dnf working
- Fedora/RISC-V running on the HiFive Unleashed board
The status of Fedora RPMs

Currently, **Fedora 29/Rawhide** packages are building in **the Koji build system**.

We are also building nightly disk images called "Fedora Developer", which has extra packages installed for developers.

Welcome to try! Please use the latest one.

---

**The Fedora rawhide RPMs REPO for riscv64**

**The bootable Nightly Build images for riscv64**
Facility: Koji system/build farm

Current build farm:
- **3 HiFive Unleasheds** (one with SSD)
- **6 qemu** instances on Intel servers
- **2 x86_64 nodes** for faster repo generation

**Building Distro:** Fedora 29/Rawhide
Boot Fedora on QEMU
Boot Fedora on QEMU

Install pre-built binaries

- dnf **copr** enable rjones/riscv
- dnf install riscv-qemu

Download pre-built Fedora for RV64

- **bbl** from fedorapeople
- And the nightly builds image

QEMU

- **bbl** = Bootloader + Linux kernel
- Fedora-Developer-Rawhide-yyyymmdd.n.0-sda.raw.xz

Using Unxz and guestfsh on *.raw.xz file

Boot pre-built Fedora on QEMU

- qemu-system-riscv64
- -m 4G
- -kernel bbl
- -append "console=ttyS0 ro root=/dev/vda"
- -drive file=Fedora-Developer*.raw,format=raw,id=hd0

[1] will not be needed starting Fedora 29/Rawhide.
Boot Fedora on Unleashed from SD

Download pre-built Fedora for RV64

*bbl* from fedorapeople
And the nightly builds image

**bbl.u540** = Bootloader + Linux kernel + bootargs
Fedora-Developer-Rawhide-yyyymmdd.n.0-sda.raw.xz

modified pre-built binaries

Modified bootargs in *bbl*

*bootargs*:

```
root=/dev/mmcblk0p2
rootfstype=ext4 rw rootdelay=5 rootwait
console=ttys10
```

Extract fedora rootfs image from *xz* file

Using Unxz and guestfish on *.*.raw.xz file

Full instruction in Fedora Wiki page

flash binaries into the SD of Unleashed

Replace *bbl* in 1st partition

```
sudo dd if=bbl.u540 of=/dev/sdx1
```

```
sudo dd if=*.raw of=/dev/sdx2
```

“e2fsck” and “resize2fs”
Demo: Boot Fedora on Unleashed from SD
**Embedded Firmware: U-boot**

**U-boot**: Alexander Graf is working on EBBR boot flow (u-boot + GRUB2 via EFI). The patchset for enabling RISC-V in the u-boot has been upstreamed.

The GRUB2 patchset is under reconstruction for more functions.

*Currently, this is experimental, it can not boot into the Linux kernel because of the lack of EFI support in Linux kernel.*
Firmware: UEFI
(TODO)

**UEFI:** Unified Extensible Firmware Interface. Firmware interface between the platform and the operating system. Major interfaces are boot services (**BS**) and run-time (**RT**) services.

HPE engineers have made some good progress on it. HPE successfully booted Tianocore EDK2 on SiFive Freedom U500 VC707 FPGA Dev Kit.

*Currently, this is experimental, it can boot into EFI Shell through remote console, but cannot load the Linux kernel yet. We have to standardize UEFI spec and RISC-V EDK2 implementation in order to conform with a variety of RISC-V platforms.*
Firmware: ACPI

**ACPI: Advanced Configuration and Power Interface**
Static tables provided by system firmware to the standard ACPI compliant OS for system info and configuration, include primary run-time interpreted control methods, power management error handling and RISC-V processor architecture agnostic.

**SMBIOS:**
Provide SMBIOS tables for booting system to non-ACPI compliant OSes and UEFI applications on variant RISC-V processor architecture agnostic.

**RISC-V PCI UEFI Expansion ROM (type 3):**
Some PCI card require PCI expansion ROM to initialize PCI device during pre-OS system POST, such as VGA, storage, etc. PCI devices. Therefore RISC-V port of UEFI driver for PCI card is required.

HPE engineers have started working on this(proposal).

**HPE:**
- Gilbert Chen
- Abner Chang <renba.chang@gmail.com>

**Red Hat:**
- Fu Wei <wefu@redhat.com>
Development Info:

- IRC: #fedora-riscv (FreeNode)
- Fedora Wiki pages:
  - Main: https://fedoraproject.org/wiki/Architectures/RISC-V
  - Instruction of installation: https://fedoraproject.org/wiki/Architectures/RISC-V/Installing
  - Info about Bootstrapping and Building
- Main developers’ GitHub:
  - https://github.com/rwmjones
  - https://github.com/sorear
  - https://github.com/davidlt
- Koji for RISC-V:
  - Main page: http://fedora-riscv.tranquillity.se
  - dist-repos: http://fedora-riscv.tranquillity.se/repos-dist/
  - SCM: http://fedora-riscv.tranquillity.se:3000/
- Nightly build
Distro:
Why Fedora First
Fedora/RHEL/CentOS
How to get RHEL support?

get Fedora support first

Fedora is the pioneer on new technology. It is corporate supported by Red Hat.

It also feeds the RHEL product. Everything that is considered to be stable and useful for demanding enterprises, might be moved in phases towards the RHEL distribution.

so we are working on Fedora first.
Goal:
Industry standard RISC-V PC/server
Why are we here?

Open Source is awesome!
Open ISA is amazing!
RISC-V is wonderful!

Embrace Open Source!
Believe in Open ISA!
Focus on RISC-V!
Learn from other architectures
What We Need:

<RISC-V PC/Server Base System Architecture>
- like the SBSA:
  - Define minimal architectural features
  - Define minimal server SoC features

<RISC-V PC/Server Base Boot Requirements>
- like the SBBR
  - Define minimal firmware platform
  - Enables mainstream general purpose OS
Some Proposal drafts

<Industrial Specification Change Requirements for RISC-V Processor - Based PC/Server System>
  • SMBIOS / ACPI (MADT/PVAT) / SDEI / Redfish

<RISC-V PC/Server System Boot Requirements for UEFI Enabled Platform>

HPE: Gilbert Chen
    Abner Chang <renba.chang@gmail.com>
    Red Hat: Fu Wei <wefu@redhat.com>

<RISC-V Unix-class Platform Specification>
from https://github.com/riscv/riscv-platform-specs/
Acknowledgments
Alphabetical Listing by Company Name

David Abdurachmanov

Hewlett Packard Enterprise
Abner Chang

JL Semi.
Xiongfei Guo

Al Stone
DJ Delorie
John Feeney
Richard Jones

RISC-V
THANK YOU

plus.google.com/+RedHat
linkedin.com/company/red-hat
youtube.com/user/RedHatVideos

facebook.com/redhatinc
twitter.com/RedHatNews