Status of Fedora's RISC-V Porting Efforts

Richard W.M. Jones, David Abdurachmanov, Andrea Bolognani, Kashyap Chamarthy

April 30, 2025

Abstract

Fedora Linux has made notable progress porting to the RISC-V architecture over the past eight years. Most of the packages are ported to RISC-V. In this session will cover the latest progress, the status of the remaining packages, and briefly evaluate emulation capabilities for those without hardware. We'll also discuss challenges and discuss the path towards making RISC-V a primary architecture on Fedora.

Introduction

Fedora Linux is a distribution known for its speed, wide content set, and support for multiple architectures. Efforts to enable RISC-V date to 2016, before physical hardware was available. At that time developers needed to run RISC-V on an FPGA or use emulation via software such as QEMU.

Fedora contains tens of thousands of open-source packages, many of which require architecture-specific code to perform optimally. The term "porting" here refers to building standard Fedora packages for RISC-V with their original, unmodified upstream sources.

As of spring 2025, a range of development boards are available, and Fedora continues to gain ground in RISC-V enablement. Fedora 41 RISC-V images were released in January 2025, two months after the primary architectures (x86, Arm, etc). However, by April 2025, Fedora 42 RISC-V images were released[1] *alongside* the primary architectures—this marks a major milestone; more on this in the next section. In this session, we'll provide an update on a few topics:

- What is the porting progress of Fedora packages for RISC-V (`RV64GC` baseline)? What are the challenges so far?
- What can you do when you don't have hardware? What's possible with QEMU's emulated RISC-V, and what are the supported "targets", i.e. boards?
- What does the future look like in terms of making RISC-V a primary architecture on Fedora? How complete and stable is the distribution?

This session will be a follow-up to the 2024 update[2] given at DevConf.CZ in Brno by Richard W.M. Jones and David Abdurachmanov.

Porting Progress

Fedora's main development branch, called *Rawhide*, has about 24,000 packages, and the vast majority of them have already been ported to `riscv64`. A way to measure Fedora's progress here is by tracking the number of "unresolved dependencies"—that is, whether all of a package's dependencies ("deps") can be satisfied *within* the repository. We constructed a '`dnf repoclosure`' command to find out and these are the results for the last three Fedora releases (F40, F41, and F42):

- 6943 unresolved deps across 777 packages for F40
- 3881 unresolved deps across 478 packages for F41
- 372 unresolved deps across 137 packages for F42

We see that from F40→F41, there's a 44% reduction in unresolved dependencies, while from F41→F42, we see almost a 90.4% reduction—an *exceptional* progress during the Fedora 42 development cycle.

Challenges

Among others, these are the remaining Fedora packages waiting to be ported. The Fedora RISC-V tracker[3] lists about 150 packages that still require work before they can be merged into mainline Fedora. A selection of important packages that need work, or already in progress, include: LLVM, Linux kernel, R language, OpenJDK, Clang, LibreOffice, CMake, Cython, some Golang libraries, and a few more.

Some of these packages have already been built on RISC-V hardware with *modified* Fedora sources. They need further investigation—e.g. investigate failing tests, submit a patch to their corresponding upstream project, work with relevant maintainers, etc.

As a concrete example, let's look at LLVM, the compiler infrastructure project with several complex subprojects and test suites. As of this writing, it has about 67 test failures[4] across three of its sub-projects:

'libarcher' (it enables GPU offloading), 'libomp' (parallel computing infrastructure), and 'MLIR' (it helps build reusable, domain-specific compilers). The pending work here is to investigate the said test failures, work with LLVM upstream and Fedora maintainers, submit patches where we can, get them reviewed, and finally integrate them into Fedora. Here's[5] the in-progress RISC-V enablement work for LLVM on Fedora. This is all traditional, open-source development process.

Emulated RISC-V with QEMU

Not everyone interested in RISC-V might have access to hardware. This is where emulation shines. QEMU, the open-source machine emulator and virtualizer, comes with a RISC-V emulator[6]. On Fedora you can get it by installing the package *qemu-system-riscv-core*. It comes with a handful of development boards, or "machine types" in QEMU parlance. E.g. SiFive's *HiFive Unleashed*; to see the full list, run: '`qemu-system-riscv64 -machine help`'.

What board to choose? Unless you're developing for a specific board, or care about "reproducing the idiosyncrasies of a particular bit of hardware", the best option is to go with the '`virt`' board. It is not tied to any specific hardware, and is designed for use in virtual machines. QEMU has a complex command-line, and manually generating the command-line can be quite tedious. Follow these[7] instructions to get started with QEMU's RISC-V emulator on Fedora. For example, the "`Fedora-Server-Host-Generic`" and "`Fedora-Cloud-Base-Generic`" images aren't tied to specific hardware, and can be tested with QEMU.

Toward RISC-V as a primary architecture

Fedora defines two tiers[8] of architectures: *primary*, and *alternative*. The *primary* architectures serve majority of the users, currently, `x86_64` and `aarch64`; build failures here are considered "fatal"—i.e., you can't merge a change into a package. While *alternative* architectures, such as RISC-V, are maintained by "motivated Architecture Maintainer Teams". Here the build failures are "non-fatal" in the sense that they can't block deliverables for the *primary* architectures.

To make `riscv64` a *primary* architecture on Fedora, there are a few hurdles: (1) finish porting the remaining unported packages; (2) make a Fedora RISC-V build-system available to regular Fedora packagers, to build their software on RISC-V; and most importantly, (3) datacenter-grade hardware—today's developer boards such as StarFive's *VisionFive 2* and SiFive's *HiFive Premier P550* are promising but don't match x86 or Arm performance for general computing. True primary architecture status requires rackable servers with traditional server-management features (BMC, IPMI, etc). As RISC-V continues gaining traction, these hardware limitations will likely resolve with time.

Get Involved

Whether you have RISC-V hardware or not, there are many ways to contribute: help port the remaining set of packages to Fedora mainline, test and report bugs in packages you care about, debug test failures, submit patches upstream, improve documentation, and more. If this piques your interest, drop by the Fedora RISC-V Matrix channel, or write to Fedora RISC-V special interest group (SIG) discussion forum[9].

Thanks and Acknowledgements

(In alphabetical order.)

- David Abdurachmanov, Rivos Inc.
- Andrea Bolognani, Red Hat, Virtualization and RISC-V
- Kevin Fenzi, Red Hat, Fedora Infrastructure Lead
- Wei Fu, Red Hat, RISC-V
- Richard W.M. Jones, Red Hat, Virtualization, Fedora, and RISC-V

References

[1] *Fedora 42 RISC-V image availability.* URL: https://discussion.fedoraproject.org/t/fedora-42-risc-v-non-official-images-are-available/148866.

[2] *An update on Fedora's RISC-V port, 2024.* URL: https://pretalx.com/devconf-cz-2024/talk/Q7XB3M/.

[3] *Fedora RISC-V tracker.* URL: https://abologna.gitlab.io/fedora-riscv-tracker/.

[4] *LLVM test failures on Fedora.* URL: https://kashyapc.fedorapeople.org/risc-v/logs/test-failures-llvm-rawhide-p550.txt.

[5] *LLVM Fedora pull request.* URL: https://src.fedoraproject.org/rpms/llvm/pull-request/408.

[6] *QEMU RISC-V emulator.* URL: https://www.qemu.org/docs/master/system/target-riscv.html.

[7] *Fedora RISC-V on QEMU.* URL: https://fedoraproject.org/wiki/Architectures/RISC-V/QEMU.

[8] *Fedora Architectures.* URL: https://fedoraproject.org/wiki/Architectures#Structure.

[9] *Fedora RISC-V SIG.* URL: https://discussion.fedoraproject.org/tags/c/project/7/risc-v-sig.