RuyiSDK - An Integrated and Customizable Toolkit for RISC-V Software Development

Weilin Cai[1], Yilin Chen[1] and Yunxiang Luo[1]

[1]Programming Language and Compiler Technology Lab, Institute of Software, Chinese Academy of Sciences (ISCAS)

Abstract

The RISC-V instruction set's design has given rise to a highly diverse ecosystem. However, the introduction of vendor-defined extensions has the potential to lead to fragmentation, creating challenges for developers in managing toolchains and adapting software. This paper presents RuyiSDK which is a comprehensive solution tailored for RISC-V developers. It is designed to address these challenges by integrating existing foundational software, promoting the adaptation of unsupported applications, and cultivating a vibrant developer community. RuyiSDK offers a package index that consolidates toolchains, emulators, and more, along with profile files that describe how to perform cross-platform builds, as well as software and RISC-V boards co-development. A key component of RuyiSDK is the Package Manager, which not only inherits the capabilities of traditional package managers but also incorporates advanced features such as virtual environment creation, device provisioning, and plugin support. The Package Manager transparently applies virtual environment profiles to the corresponding toolchains by reading the package index. Users only need to specify the target development board, without needing to manage toolchain differences manually. The device provisioning feature provides interactive guidance for keeping system images up to date for specific development boards, while test reports offer a certain level of quality assurance. The plugin system enables vendors and users to extend the Package Manager with custom features, thereby making it adaptable to various workflows. By providing a flexible, efficient, and transparent cross-platform development environment, RuyiSDK empowers developers to focus on innovation, thereby unlocking the full potential of RISC-V hardware while mitigating ecosystem fragmentation.

Introduction and Motivation

The modular design of the RISC-V instruction set has fostered a highly diverse ecosystem. However, the introduction of vendor-defined extensions has led to potential fragmentation. For SoCs based on specific CPU cores, achieving optimal performance often require customized toolchains, emulators, firmware, and operating systems. To save on maintenance costs, vendors typically only maintain specific versions of toolchains and emulators, rather than actively continuing maintenance or pushing the code upstream.

RuyiSDK's goal is to provide a comprehensive, one-stop solution tailored for RISC-V developers. It aims to address these challenges by integrating foundational software, promoting adaptation of unsupported applications, and cultivating a vibrant developer community. RuyiSDK maintains GCC and LLVM forks, providing open-source vendor extension support and pushing patches upstream. It also maintains distributions such as RevyOS, based on Debian, to adapt and continuously support development boards using specific SoCs. The aim is to provide a stable and upstream-evolving software ecosystem for all development boards.

The software provided by RuyiSDK, together with upstream and vendor-provided tools, forms a large and complex ecosystem. For developers and enthusiasts new to RISC-V, this ecosystem can be overwhelming. This paper primarily introduces the design, implementation, and development progress of the Ruyi Package Manager and its Package Index, which aim to make everything more accessible and developer friendly.

Methodologies

The design of the Ruyi Package Manager is illustrated in Figure 1. It clones the Ruyi Packages Index to the local system or updates it based on an existing cache. Users can then select and download the required packages online and perform offline development using the Ruyi Virtual Environment, or flash system images to development boards using the Device Provisioning Wizard.

The Ruyi Device Provisioning Wizard provides an interactive Q&A interface that helps users flash system images onto their devices and access relevant development resources. When users receive a new RISC-V development board, this feature assists them in obtaining the latest firmware and system updates. If a development board can be connected to the host machine in a supported manner and somehow accurately identified, the wizard is able to recognize the board and list the latest available firmware and system images in the world. Alternatively, the user can manually specify the board model to initiate the process.

Once the development board has been updated and successfully booted, RISC-V software development can begin. Developers can use the Ruyi Package Manager to install the appropriate toolchain and create a Ruyi Virtual

Environment. In a typical cross-development workflow, source packages are extracted into the workspace, built using GCC, and tested with emulators. For specific SoCs, extra arguments are required to enable or disable instruction set extensions. Ruyi Packages Index includes a set of entity lists that record information about development boards, SoCs, toolchains, and microarchitectures, allowing developers to create a Ruyi Virtual Environment and begin development simply by specifying the model of the board.

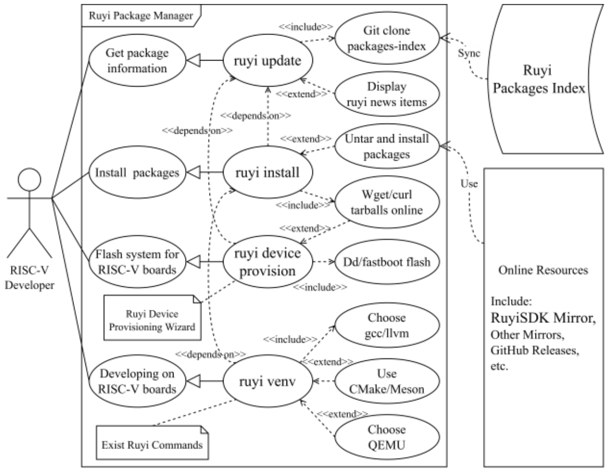

Figure 1: Design of Ruyi Package Manager.

For maintainability, the actual binary repositories and the package index are separated. The Ruyi Packages Index is just a set of TOML configuration files, which includes not only download links but also metadata for Ruyi Virtual Environment, the Ruyi Device Provisioning Wizard, plugins and more. Ruyi packages fall into six main categories: toolchain, emulator, source tarball, board image, analyzer, and extra package. Most of these packages are distributed as binaries released by RuyiSDK or other organizations. Toolchains and emulators include upstream versions, vendor-specific versions and RuyiSDK maintained versions, to ensure broad support for various development boards. Meanwhile, board image packages are synchronized with another RuyiSDK subproject called the RISC-V Board and OS Support Matrix, which ensures alignment with upstream releases while also providing test reports and compatibility assessments. Since the Ruyi Packages Index consists of plain text files, it can be easily managed using a Git repository, avoiding the need to redistribute vendor-released binaries.

Result

The Ruyi Package Manager has already implemented its core functions and is still under rapid development and iteration. It releases a new version every two weeks and tests on over eight Linux distributions and three architectures (x86_64, riscv64, and aarch64), supports most Linux distributions via Nuitka binaries.

The Ruyi Package Index currently provides over 60 packages for 30 different development boards. RuyiSDK has already achieved comprehensive support for RISC-V development board from MilkV and Sipeed, offering not only dedicated toolchains, emulators, and system images, but also source code demos. In adittion, some toolchains maintained by RuyiSDK provide support not yet fully available in upstream projects, for example, gnu-plct-rv64ilp32-elf supports the rv64ilp32 ABI, and llvm-plct includes support for the xtheadvector extension.

With the Ruyi Virtual Environment, users do not need to manually specify -mcpu, -march, or -mabi parameters for a target RISC-V board. In recent versions, manual selection is still required for toolchain and emulator used in a virtual environment, but future versions will utilize the entity list to assist with this process. The Ruyi Packages Index already includes entity lists for 33 devices and 19 microarchitectures, ranging from the generic rv64gc, Sifive U74, Xuantie C910, to the currently uncommon Xiangshan Nanhu architecture.

The Ruyi Device Provisioning Wizard now allows users to flash (via dd or fastboot) the latest images to RISC-V boards and access embedded board development resources, with the assurance of image quality verified through test reports in the RISC-V Board and OS Support Matrix. Typically, a development board comes in different variants, such as those with varying sizes of onboard DDR memory. Users can simply specify the model and variant of the development board to get a list of supported operating systems. If the board requires a firmware update, the recommended version will be automatically flashed, often by fastboot. Note that dd images to MicroSD card usually requires root privileges. The Ruyi Device Provisioning Wizard will only invoke sudo after obtaining the user's consent.

Another challenge in promoting RISC-V boards is that closed-source software is often released only for x86_64 architecture. The Ruyi Package Index offers Box64 package which can run with systemd-binfmt, enabling users to easily configure environments for running Linux applications across architectures or using Wine to run Windows applications on RISC-V systems.

To enhance customizability, package category names are configurable, the package index URL is configurable, allowing users to manage their own packages and repositories using Ruyi. The Ruyi Package Manager includes commands to assist with repository management and can be used on any architecture, as it is developed entirely in pure Python. Both the package manager and the package index support plugins, enabling further extensibility.

Conclusion

Ruyi Package Manager has demonstrated wide compatibility across platforms and distributions, supporting both native and cross-development for RISC-V. Its powerful features, such as virtual environments, device provisioning, plugin extensibility, and user customizability, make it a practical and scalable toolkit for the growing RISC-V software ecosystem. It enables developers to focus on innovation and helps expand the influence of RISC-V, contributing to its development as a mainstream architecture.