



Contribution ID: 391

Type: **not specified**

Preparing RISC-V Linux for RVA23

The number of RISC-V extensions is ever increasing. To manage the wide variety of extensions that are available to hardware vendors, RISC-V International (RVI) has introduced “profiles” that define groupings of extensions for different classes of hardware.

The currently relevant profile is named RVA23 and specifies a set of extensions that supervisor and userspace software vendors can rely on for general use on performance focused hardware. Key aspects of RVA23 are:

1. 64-bit
2. Vector
3. Hypervisor
4. CFI

The RISC-V kernel thus far has conformed to the idea that a single kernel that can support *every* extension should be able to be built to run on *any* RISC-V hardware. To accomplish this:

1. The C code in the kernel is compiled with minimal extensions, namely rv64imac. In order to use any other extension, the routine must be written in assembly and at runtime selectively enabled based upon if the extension is present in the isa string that is passed into the kernel from the firmware. Select C files may be compiled with other extensions such as fd.
2. The assembly code in the kernel is assembled with rv64imafdv. Similarly to C, any other extensions need to be selectively enabled at runtime. For the code that is executed without knowledge of what is in the isa string, the misa CSR is read to determine if the relevant extensions, namely fdv, are supported by the hardware.

One of the primary issues with this is that the kernel C code is not being compiled with any of the performance enhancing extensions, such as bitmanip. Another issue is that the kernel has 2^n runtime configurations depending on which extensions are in the isa string, where n is the number of runtime extensions. For each of these configurations, either a static branch was set, an alternative flipped, or the isa string checked – introducing more performance penalties and extra developer overhead. For more invasive extensions that change userspace ABI, it is not always feasible to have dynamic patching so separate kernels must be compiled.

Adapting the kernel to support RVA23, and future profiles, can alleviate many or all of these issues. When RVA23 can be assumed, the kernel can be compiled with performance enhancing instructions, skip checking misa or the isa string for the large majority of extensions, and confidently support extensions that change userspace ABI.

This talk will explain how this is possible and the proposed patches to achieve this goal of a RVA23 compatible kernel that is able to take advantage of all of the benefits.

Primary author: JENKINS, Charlie (Rivos)

Presenter: JENKINS, Charlie (Rivos)

Session Classification: RISC-V MC

Track Classification: RISC-V MC