Getting Started with RISC-V Software Development

Khem Raj <raj.khem@gmail.com>
IRC: @khem
Bay Area RISC-V Group, 23-May-2024

Contributions

- Events Like this one
- Project Websites
- Design
- Bug triages
- Documentation
- Mailing list discussions
- Helping new users/developers
- Connecting community members
- Yes “Code” counts too

RISC-V “Upstream First” Approach

- RISC-V Software Ecosystem
  - No dedicated staging area
  - Specification work
    - https://github.com/riscv
  - Software collaboration work
    - https://github.com/riscv-collab
  - Software maintained by RISC-V foundation
    - https://github.com/riscv-software-src

Toolchains

- GNU toolchain port - Upstream
- LLVM/Clang port - Upstream
- QEMU port - Upstream
- Musl C Library
  - RV64 port upstream
  - RV32 Port Added in 2024

```
Date: Thu, 29 Feb 2024 21:14:28 -0500
From: Rich Felker <dalias@...c.org>
To: musl@...ts.openwall.com
Subject: musl 1.2.5 released

This release adds extension functions statx and preadv2/pwritev2, with
fallback implementations for older kernels, and adds two new ports:
loongarch64 and riscv32. Minor changes to the printf family of
functions have been made for conformance to new standards
interpretations/requirements. TLSDESC support for riscv64 has also
been added.
```

Toolchains

- Crosstool-NG - Build from source
  - https://github.com/crosstool-ng/crosstool-ng
- Prebuilt Toolchains
- Bootlin
  - GCC/glibc/musl/uclibc based (hosted only)
    - https://toolchains.bootlin.com/
    - For RV32 and RV64
- RISC-V Collaboration (clang/llvm and newlib/baremetal also available)
  - https://github.com/riscv-collab/riscv-gnu-toolchain/releases/tag/2024.04.12

Language runtimes

- Go - RISCV64 supported since 1.14 release
  - RISCV32 port not available
- Rust
  - tier 2 support for RISC-V RV64
  - RISCV32 support Experimental
  - Bare-metal supports more machines
    - https://github.com/riscv-rust/riscv-rust-quickstart
- Dart SDK targets RISCV64
- More Details
  - https://wiki.riscv.org/display/HOME/Language+Runtimes

Bootloaders

- U-boot - “virt” machine 32-bit/64-bit
  - https://github.com/u-boot/u-boot/blob/master/doc/board/emulation/qemu-riscv.rst
- OpenSBI - https://github.com/riscv-software-src/opensbi
- Coreboot

Kernel

- 32/64bit ports are upstream
  - 4.15 RV64
  - 5.4 RV32 (ILP32)
- Mailing list
  - linux-riscv@lists.infradead.org
- Patches
  - https://lore.kernel.org/linux-riscv/
- IRC
  - #riscv on libera.chat

Kernel TODO

```
kraj@apollo ~/work/linux master ↑
› ./Documentation/features/list-arch.sh riscv | grep TODO
      core/ cBPF-JIT             : TODO |                            HAVE_CBPF_JIT #  arch supports cBPF JIT optimizations
     debug/ optprobes            : TODO |                         HAVE_OPTPROBES #  arch supports live patched optprobes
     debug/ user-ret-profiler    : TODO |                HAVE_USER_RETURN_NOTIFIER #  arch supports user-space return from system call profiler
   locking/ cmpxchg-local        : TODO |                      HAVE_CMPXCHG_LOCAL #  arch supports the this_cpu_cmpxchg() API
   locking/ queued-spinlocks     : TODO |                ARCH_USE_QUEUED_SPINLOCKS #  arch supports queued spinlocks
      time/ virt-cpuacct         : TODO |                 HAVE_VIRT_CPU_ACCOUNTING #  arch supports precise virtual CPU time accounting
        vm/ ioremap_prot         : TODO |                        HAVE_IOREMAP_PROT #  arch has ioremap_prot()
        vm/ PG_uncached          : TODO |                    ARCH_USES_PG_UNCACHED #  arch supports the PG_uncached page flag
```

Zephyr

- Several RISC-V based boards supports
  - Default SDK supports RISC-V out of box
- Status
  - https://docs.zephyrproject.org/latest/hardware/arch/risc-v.html
- Getting Started
  - https://docs.zephyrproject.org/latest/develop/getting_started/index.html
- Porting Guide
  - https://docs.zephyrproject.org/latest/hardware/porting/index.html
- Other RTOSes
  - FreeRTOS, RTEMS …

Debian

- Supports RV32/RV64
- RV64 is well supported
  - https://wiki.debian.org/Ports/riscv64
- Installing
  - Sipeed/LicheeRV
- IRC - #debian-riscv on OFTC
- Mailing List
  - https://lists.debian.org/debian-riscv/

riscv64 is now an official architecture

- *To*: debian-riscv@lists.debian.org
- *Subject*: riscv64 is now an official architecture
- *From*: Aurelien Jarno <aurel32@debian.org>
- *Date*: Sun, 23 Jul 2023 14:03:04 +0200
- *Message-id*: <ZL0W+B8iPdy/1UC1@aurel32.net>

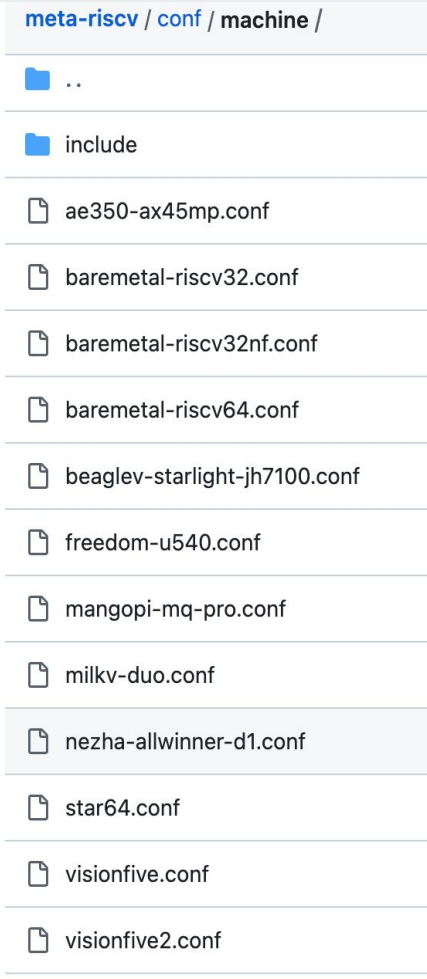
```
Dear all,

Some of you have been following on IRC, some of you have noticed the
ACCEPTED mails from dak on the mailing list, some of you may have
noticed the recent closure of bug #1033658. For all the others, I am
happy to share the good news:

  riscv64 is now an official Debian architecture
```

Linux Distributions - Fedora

- Fedora/RISC-V project
  - Koji - http://fedora.riscv.rocks/koji/
  - Several SBCs are Available
    - T-Head - https://fedoraproject.org/wiki/Architectures/RISC-V/T-Head
    - Allwinner - https://fedoraproject.org/wiki/Architectures/RISC-V/Allwinner
- Installing Fedora
  - https://fedoraproject.org/wiki/Architectures/RISC-V/Installing
- IRC - #fedora-riscv
- Discourse - https://discussion.fedoraproject.org/tag/risc-v-sig

Embedded - Yocto/OpenEmbedded

- Flexible and customizable
- QEMU machines supported in Core layer
  - RV64/RV32
- Some machines via RISC-V machine layer
  - https://github.com/riscv/meta-riscv
  - Vendor layers
- RISC-V is not officially tested in yocto autobuilders
- IRC - #yocto
- Mailing Lists - https://www.yoctoproject.org/community/mailing-lists/



Building Yocto

```
$ mkdir riscv-yocto && cd riscv-yocto
$ repo init -u https://github.com/riscv/meta-riscv  -b master -m tools/manifests/riscv-yocto.xml
$ repo sync
$ repo start work --all
$ . ./meta-riscv/setup.sh
$ MACHINE=qemuriscv64 bitbake core-image-full-cmdline
$ MACHINE=qemuriscv64 runqemu nographic
```

Embedded - Buildroot

- Simple and small
- Supports RV32/RV64
- Supports RV32 - NOMMU
- Mailing List - https://lists.buildroot.org/mailman/listinfo/buildroot
- Support - https://buildroot.org/support.html

Embedded - Buildroot

$ git clone git://git.busybox.net/buildroot
$ cd buildroot
$ make **qemu_riscv64_virt_defconfig**
$ make
$ qemu-system-riscv64 -M virt -kernel output/images/bbl -append "root=/dev/vda ro console=ttyS0" -drive file=output/images/rootfs.ext2,format=raw,id=hd0 -device virtio-blkdevice,drive=hd0 -netdev user,id=net0 -device virtio-net-device,netdev=net0 -nographic

OS Distributions

- OpenSuSE
  - https://en.opensuse.org/openSUSE:RISC-V
- Gentoo
  - https://wiki.gentoo.org/wiki/Project:RISC-V
- Ubuntu
  - https://ubuntu.com/download/risc-v
- OpenWRT
  - https://openwrt.org/docs/guide-user/virtualization/qemu#openwrt_in_qemu_risc-v
- AOSP/Android
  - https://github.com/riscv-android-src
  - https://github.com/aosp-riscv

RISE - https://riseproject.dev/

- Setup to collaborate on accelerating the development of open source software for the RISC-V architecture
- Publishes RFPs for gaps identified
  - https://wiki.riseproject.dev/display/HOME/RISE+RFP

Overview Of RISC-V Software Ecosystem

- https://wiki.riscv.org/display/HOME/RISC-V+Software+Ecosystem

Thanks for your time