Go RISC-V Go: The State of Software Development Tools for RISC-V

KHEM RAJ

#lfelc @himvis
Agenda

- Toolchains
- System tools
- Language Runtimes
- Operating Systems
Emulators

• Upstream QEMU support RISC-V
  – Emulate RISC-V system on x86
  – Vice-versa
  – https://wiki.qemu.org/Documentation/Platforms/RISCV

• QEMU 5.0
  – Experimental support for hypervisor extensions
  – Goldfish RTC, generic Linux syscon drivers

• TinyEMU
  – https://bellard.org/tinyemu/
  – Buildroot based port for RV32 and RV64 toolchains
  – Emulates RV128IMAFDQC base ISA

• Spike - https://github.com/riscv/riscv-isa-sim
  – RISC-V ISA Simulator

• Renode - https://github.com/renode/renode
  – virtual development tool for multi-node embedded
Bootloaders

• Upstream Support
  – Coreboot
  – U-boot
  – BBL
  – OpenSBI
    • Implements RISC-V Supervisor Binary Interface Spec
      – https://github.com/riscv/riscv-sbi-doc
Toolchains

• GCC10
  – Added support for new assembly instructions
    • Emits fmv.x.w/fmv.w.x instead of fmv.x.s/fmv.s.x
    • Needs binutils 2.30+

• Bit Manipulation Instructions extension
    • Emitted in code directly
    • Available as intrinsics
• RISC-V Target support in LLVM 9
• RISC-V LTO support in clang 10
• Bit Manipulation Extension for RISC-V
• Support Upstream Since 2017
• Added support for privileged spec v1.9.1
• Assembler options to set ISA versions
  – -misa-spec
• GDB 9
  – GDB server is now supported on RISC-V GNU/Linux
  – Supports Linux Native and Target Configuration since 8.3
    • riscv*-*-linux*
JTAG Debug for RISC-V Cores

• Out of tree OpenOCD port
  – Multi-core, 64-bit, etc.

• Segger Embedded Studio
  – Single core, 32-bit only

• Lauterbauch TRACE32
System C library

• Musl now supports RISC-V 64bit (since 1.1.23)
  – 32bit port not available yet

• Glibc
  – 32bit port is submitted
  – Might show up in 2.32
  – 64-bit RISC-V requires a minimum kernel headers version of 5.0
System C library

• Newlib
  – Syscall improvements
    • only the required syscall argument registers are set.
  – Newlib nano libm supports RISC-V
  – RISC-V size optimized mem* functions
Languages

- **Go**
  - 1.14 contains experimental support for 64-bit RISC-V
    - [https://golang.org/doc/go1.14#riscv](https://golang.org/doc/go1.14#riscv)
    - [https://github.com/golang/go/issues/27532](https://github.com/golang/go/issues/27532)
  - CGO Support
    - [https://github.com/4a6f656c/go/tree/riscv64-cgo](https://github.com/4a6f656c/go/tree/riscv64-cgo)
  - Supported Out of tree
    - [https://github.com/4a6f656c/riscv-go/](https://github.com/4a6f656c/riscv-go/)
    - Based on 1.13 release
  - Used with OpenEmbedded/Yocto project
    - [https://github.com/riscv/meta-riscv](https://github.com/riscv/meta-riscv)
    - Will work on 1.14 support
  - Bootstrap
    - [https://github.com/carlosedp/riscv-bringup/blob/master/build-golang.md](https://github.com/carlosedp/riscv-bringup/blob/master/build-golang.md)
Rust

• Rust now uses llvm 10.x
  – https://github.com/rust-lang/rust/pull/67759
  – Means RISC-V backend is available
  – Tier-2 target
• Rust Embedded Workgroup
  – https://github.com/rust-embedded
• Baremetal
    • Uses 1.36+ Release
• riscv-rt - Minimal runtime/startup for RISC-V CPU’s
  – https://github.com/rust-embedded/riscv-rt
• Low level access to RISC-V processor
  – https://github.com/rust-embedded/riscv
    • Needs 1.42.0 or newer
• Wasmtime/Cranelift
  – A standalone runtime for WebAssembly
    • https://github.com/bytecodealliance/wasmtime
Language Runtimes

• Java
  – ZeroVM backend works
  – No hotspot

• Ocaml
  – Out of tree support
    • https://github.com/nojb/riscv-ocaml/commits/trunk
  – Crosscompiler docker image
    • https://github.com/kayceesrk/riscv-ocaml-cross
Linux Operating Systems

- Debian
  - [https://wiki.debian.org/RISC-V](https://wiki.debian.org/RISC-V)
  - SiFive "Freedom U540" SoC (quad-core RV64GC) / "HiFive Unleashed"
  - QEMU
Mid-2019
Follow Grey Line

Latest
https://buildd.debian.org/stats/graph-ports-big.png
Linux Operating Systems

• Fedora
  – Fedora for RISC-V is mirrored as a Fedora “alternative” Architecture
  – Fedora Minimal, Developer, GNOME images
  – Supported Targets
    • Virtual – QEMU
      – Can run on libvirt/QEMU with Graphics (Spice)
  – SiFive Unleashed
    • Fedora GNOME Image can run on SiFive Unleashed
      – Needs Expansion Board, PCIE graphics Card & SATA SSD)
Linux Operating Systems

- OpenSUSE
  [https://en.opensuse.org/openSUSE:RISC-V](https://en.opensuse.org/openSUSE:RISC-V)
  - Tumbleweed images
  - Runs using systemd-nspawn
  - Packages
    - [https://download.opensuse.org/ports/riscv/](https://download.opensuse.org/ports/riscv/)
- Gentoo
Embedded Linux

- OpenEmbedded/Yocto Project
- RV32/RV64 Support in Latest 3.1 Release
  - https://lists.yoctoproject.org/g/yocto/message/49201
  - QEMU 64bit is in Core
  - Linux-yocto supports RV64
  - Supports Both Musl and Glibc
  - Available ABIs riscv64 riscv32 riscv64nf riscv32nf
- Architecture/machine layer ( meta-riscv )
  - https://github.com/riscv/meta-riscv
    - Supports baremetal SDK
      - RV32/RV64
      - Newlib and baremetal SDKs
    - freedom-u540
    - QEMU RV32
    - Supports Cross building Go packages
- Supports Clang as system compiler for RV32/RV64
- QEMURISCV64 can run Yocto automatic tests ( ptest )
  - bitbake -ctestimage <image>
Embedded Linux

• Buildroot - Upstreamed
  – Full support for RV32 and RV64 architecture
    • [https://www.embecosm.com/2019/04/01/buildroot-support-for-32-bit-risc-v/](https://www.embecosm.com/2019/04/01/buildroot-support-for-32-bit-risc-v/)
    • Supports kernel 5.4.x
    • Regularly tested
      – [http://autobuild.buildroot.net/?arch=riscv64](http://autobuild.buildroot.net/?arch=riscv64)
    • Supports both musl/glibc
      – Musl not supported on RV32
    • Supports QEMU machines (riscv64-virt, riscv32-virt)
Embedded Linux

• OpenWRT
  – Added support for SiFive RISC-V
    • https://openwrt.org/docs/techref/hardware/soc/soc.sifive
      – HiFive Unleashed (development board)
      – FPGA-based implementation (Virtex7)
      – QEMU
  – Supports musl/glibc
  – https://freesoft.dev/program/125876759
BSD Operating Systems

• FreeBSD – Since 2016
  – Supported Devices
    • HiFive Unleashed (SiFive FU540)
    • Spike
    • QEMU
  – https://riscv.org/software-tools/freebsd/
  – https://wiki.freebsd.org/riscv
  – IRC - #freebsd-riscv

• NetBSD
  – Added in 10.0
Embedded Software/RTOS

- FreeRTOS – Upstream
  - https://www.freertos.org/Using-FreeRTOS-on-RISC-V.html
- Supported Boards
  - RTOS Demo for RISC-V MiFive M2GL025 / Renode
  - RISC-V RV32M1 VEGAboard Demo (RI5CY Core)
  - QEMU sifive_e Model

AWS Announces RISC-V Support in the FreeRTOS Kernel

Posted On: Feb 26, 2019

RISC-V support is now available in the FreeRTOS kernel, a feature enabling embedded developers to create IoT applications on the officially supported FreeRTOS kernel for microcontrollers that use the free, open, and extensible RISC-V Instruction Set Architecture (ISA).
Embedded Software/RTOS

• Zephyr
  – Upstream Since 1.13.0
    • support for HiFive1
    • https://www.sifive.com/blog/getting-started-with-zephyr-rtos-v1.13.0-on-risc-v
  – Supports LiteX soft SoC RV32 VexRiscv CPU
  – Part of Standard Zephyr SDK
    • https://github.com/zephyrproject-rtos/zephyr/releases/tag/zephyr-v2.3.0
  – Hard-float support
  – Compiler tunes (march/mabi)
Embedded Software/RTOS

• RTEMS
  – Upstream support for simulators
  – RISC-V Clang/LLVM support in the RTEMS WAF build system
  – Add RISC-V GDB for RTEMS
    • [https://devel.rtems.org/ticket/3453](https://devel.rtems.org/ticket/3453)
  – RISC-V toolchain support 64-bit chips
    • riscv target supports both rv32 and rv64
  – Freedom E310 Arty A7 FPGA BSP Added
    • [https://devel.rtems.org/ticket/3785](https://devel.rtems.org/ticket/3785)
Embedded Software/RTOS

• Xv6 Supports RISC-V
  – modern reimplementation of Sixth Edition Unix in ANSI C
  – Used in Educational institutions

• HelenOS
  – Micro-kernel approach
Help Needed

• Java
  – No hotspot port yet
• V8
• NodeJS
• Dart
Come Join the Us

- Most Software is Upstream!
  - Use a project’s regular communication mechanisms
- Specific to RISC-V
  - [https://github.com/riscv/](https://github.com/riscv/): Contains in-progress ports
  - [sw-dev@groups.riscv.org](mailto:sw-dev@groups.riscv.org): Software discussion
  - [patches@groups.riscv.org](mailto:patches@groups.riscv.org): Patches to RISC-V ports
- #riscv on Freenode: General RISC-V discussion
- linux-riscv@lists.infradead.org: RISC-V Linux Port
- Stack Overflow
  - [https://stackoverflow.com/questions/tagged/riscv](https://stackoverflow.com/questions/tagged/riscv)
Come Join the Us

- RISC-V International Maintained Software Status
- Submit Pull requests for updating it
  - [https://github.com/riscv/riscv-software-list](https://github.com/riscv/riscv-software-list)
Thanks for your time