

Working Towards a Debian RISC-V Port

Manuel A. Fernandez Montecelo <maf@debian.org>

4th RISC-V Workshop :: July 12-13, 2016

Computer Science and Artificial Intelligence Lab (CSAIL)

Massachusetts Institute of Technology (MIT)

Outline

- What is Debian?
- What is a Debian Port?
- Why a Debian Port for RISC-V?
- Goals of this project
- The Plan
- Progress so far
- How the RISC-V community can help
- ... and what are the benefits
- Thanks
- Questions / comments

What is Debian? (1)

- The Project: A community of individuals
 - Mostly volunteers
 - No overall or significant control by companies
 - ... who care about Free (Open-Source) Software
 - Goal: to create a full, comprehensive “Operating System”
 - Large set of software packages, coherent and well integrated
 - “stable” releases every ~2 years (also used as “rolling release”)
 - Keeping “Software Freedom” as a core goal
 - Manifesto: “... developed openly in the spirit of Linux and GNU...”
 - Social Contract
 - Debian Free Software Guidelines (DSFG)



debian

What is Debian? (2)

- A GNU/Linux “distribution”
 - Started by Ian Murdock in 1993
 - First “stable” release in 1996
 - One of the major “flavours” (.deb packages)
 - More than 22k “source” packages in 2016
 - Basis of many derivatives (“downstreams”)
 - Ubuntu, Raspbian, SteamOS, Knoppix, gNewSense, Linux Mint Debian Edition, Tails...
- ... which also supports other kernels
 - Currently: FreeBSD, GNU Hurd



What is a Debian Port? (1)

- In short, {computer arch + kernel + libc} ABI
 - Mostly with Linux + GNU libc; but also:
 - kFreeBSD + GNU libc (amd64 and i386), GNU Hurd (i386)
 - Experimental ports, e.g. with Linux + musl (libc)
- Examples:
 - Current, well supported ports:
 - amd64 (x86_64)
 - i386 (targetting i686 nowadays)
 - mips/el, arm64/armel/armhf, powerpc/ppc64el, s390x
 - Older / not well supported:
 - m68k, alpha, hppa, sh4, sparc/64, x32 (x86_64 with ilp32)...

What is a Debian Port? (2)

- Three kinds of ports:
 - Officially supported
 - Part of the “stable” releases and fully supported
 - Unofficial / unsupported
 - Not in “stable” releases, but hosted in Debian infrastructure
 - Outside of Debian infrastructure
 - Some derivatives are arguably an “external” port
 - e.g. Raspbian, mostly recompilation / optimisation for Raspberry Pi
 - Anybody can start one of these without official help or coordination with Debian
 - But often there’s collaboration in one way or another

What is a Debian Port? (3)

- Criteria for “Officially supported” ports:
 - Provide (most of) the 22k+ “source” packages
 - Have enough people responsible for it, addressing issues when they arise
 - Have hardware available, redundant, well managed, etc.
- And in return:
 - Become part of the “stable” releases and have full support for all packages (all ports)
 - For 3-5 years (“stable” release lifetime)
 - Including security support for all, not only a “core” set of packages

What is a Debian Port? (4)

- “Unofficial” or “unsupported” ports:
 - When they don’t qualify as “official” for some reason
 - Hardware not available
 - Not enough people behind
 - Many packages fail to compile
 - e.g. missing Java or Haskell ecosystem (no working compiler), no GUIs, ...
 - ... but are hosted in the infrastructure:
 - Can be downloaded / upgraded from *.debian.org
 - Secure, cryptographically signed
 - Get continuous updates of software packages
 - When contributors upload new versions to the archive, they are compiled automatically for all ports, within minutes/hours

Why a Debian Port for RISC-V? (1)

- Personal reasons:
 - Interested in both Debian and RISC-V
 - I'm already a Debian Developer
 - I like RISC-V design and goals
 - They are a good fit, philosophically and technically
 - And I would like to improve and help to grow both projects
 - Worked previously in the OpenRISC or1k port
 - ... and had lots of fun!!
 - And ultimately, I would like to run Debian when I get my own RISC-V based hardware

Why a Debian Port for RISC-V? (2)

- Technical and philosophical reasons:
 - Debian welcomes and already supports many ports
 - Many similar architectures, from “RISC-V Genealogy”:
 - MIPS (mips, mipsel),
 - Hitachi SuperH (sh4)
 - IBM PowerPC (powerpc, ppc64, ppc64el)
 - ... and DEC Alpha, SPARC (32 and 64), older ARMs, PA-RISC
 - Similar or compatible goals, and methods, e.g.:
 - Aiming for openness, freedom
 - Both based, built upon same tools
 - GNU toolchain —among others—, Linux, FOSS ecosystem
 - Striving for good technical solutions over other considerations

Goals of this project

- To have Debian ready to install & run on RISC-V based systems
- Fully integrated with Debian infrastructure
 - Not a “one shot”, but continuously updated

The Plan (1)

➤ Steps:

- 1) Bootstrap and create viable, basic OS disk images
- 2) Get it to the state of “unofficial” / “unsupported”, until all criteria for “stable” are met
 - Prepare infrastructure (build daemons, etc.), documentation, etc
 - Import bootstrapped set in the infrastructure
 - Fully rebootstrap
 - Build as many packages as possible
 - Keep the port running, in good state
- 3) In the future, when feasible, “official” — as well supported as main ports

The Plan (2)

- Software-wise, targets are:
 - Linux kernel
 - GNU libc and userland
 - Rationale: all current / well supported Debian ports target these

The Plan (3)

- Hardware-wise, the target is 64-bit little-endian
 - “default flavour” recommended by creators (AFAIK)
 - Popular target (AFAIK)
 - planned implementations for end-users, e.g. lowRISC
 - dev boards or servers that might become available in the next few years, likely 64-bit LE too
 - Both 64-bit and little-endian have been the focus for general purpose computing for years

The Plan (4)

- Why not 32-bit or 128-bit variants:
 - RISC-V side:
 - design, toolchain and simulator support have not been explored as deeply for 32-bit (AFAIK)
 - and 128 is simply not realistic at this time
 - Debian side:
 - 32-bit ports already struggle to get large packages built
 - e.g. Firefox or Chromium, LibreOffice, KDE
 - This problem will only become worse with time
 - Embedded / IoT / etc. likely not interested
 - Neither Debian nor “full”, general purpose OSs

Progress so far (1)

- Started on Nov 2014
 - Working on it on and off
- Frequent pauses:
 - Waiting for Specs ready
 - Waiting for Upstreaming of toolchain, Linux, etc.
 - not essential, but inconvenient to manage
 - Waiting for availability of:
 - Emulators / simulators (mainly, Qemu), matching specs
 - Toolchain implementation, matching specs
 - Different components of the toolchain to work well with each other

Progress so far (2)

- 300~400 “essential” packages built
 - Some very basic:
 - GNU coreutils, make, gawk, bash, tar, gzip, bzip2, xz-utils, flex, bison, wget, curl, less, ...
 - Largish, many depends., curses UI...:
 - media codecs (png, jpeg, flac, ogg, opus, xvid, ...)
 - dpkg, apt, gnupg, openssl, nano, links (text-mode web browser)
 - some X-Windows packages
 - Many libraries / supporting packages
 - Not all of them interesting per se, but needed by others (e.g. XML libraries, ncurses)

Progress so far (2)

- How were they built?
 - Mostly cross-compiled
 - Some compiled “natively” (inside emulators / simulators)
 - e.g. Perl
 - Some of them built several times
 - to untangle circular dependencies
 - to enable further functionality
 - e.g. after non-essential dependencies became available
 - Just to test
 - e.g. dpkg cross-compiled, then built natively

Progress so far (3)

- Most packages built with no modifications
 - Some would need, but they already benefit from:
 - changes to packages for previous ports:
 - build files updated for arm64, ppc64el, OpenRISC or1k
 - patches to e.g. support multiple endian-ness (usually forwarded upstream, not always applied)
 - changes to building tools from previous ports
 - ... both of which are advantages of using Debian for this
- Modifications for those which need them
 - usually minimal, e.g. disable dependencies much higher in the stack, like Qt GUI libraries
 - some support specific to RISC-V (e.g. word sizes)

Progress so far (4)

- ... so they are built, but do they work?
 - Mostly... YES
 - booting the system, moving around
 - using common tools (mount, grep, ls, find, tar, gzip...)
 - can edit files (nano, with ncurses), compile new packages!
 - Problems:
 - Found bug with syscalls, e.g. file locking wasn't working
 - dpkg could not install new packages
 - Spike and Qemu don't support network
 - Many packages cannot be tested properly
 - ABI changes mean to restart from scratch
 - Unfinished Privileged ABI, and unexpected updates in Userland's

How the RISC-V community can help

- Stabilise specs / ABI
- “Upstream” toolchain support
 - For Debian: GNU GCC, glibc, binutils, Linux, Qemu
 - If Qemu not upstreamed, at least keep up-to-date with specs and add common devices (like network and more disks)
- Stabilise Linux/glibc ABI
 - ... and fixes
- ... so basically, what Arun Thomas said in
 - “Building the RISC-V Software Ecosystem” in 3rd workshop

... and what are the benefits (1)

- Just by the Debian port being started:
 - Problems in the software stack were found
 - ... and hopefully are or will be fixed
 - some commits already accepted
 - Technical people learnt about this effort first, then about the architecture
 - I already got contacted by several people interested in one way or another

... and what are the benefits (2)

- When real-world software publicly available:
 - People more likely to pay attention to the hardware
 - that's one of the main reasons of the success of the Raspberry Pi
 - Large collection of software, useful for many reasons
 - Easy to test for base ISA, extensions and hardware
 - ARM (the company) does use Debian as testbed for changes or fixes in the architecture/ABI
 - e.g. easily test if a given software is exploitable with extra protections of lowRISC
 - Easy to benchmark with software in common use and avoid problems of synthetic benchmarks
 - as “Computer Architecture: A Quantitative Approach” would have it ;-)

... and what are the benefits (3)

- By using the Debian port directly on your RISC-V system (when ready!!):
 - Debian is easy to install and keep up-to-date
 - Install software
 - `apt-get install awesome-package`
 - Keep your system up to date easily and securely
 - `apt-get upgrade`
 - Easier development for the RISC-V ecosystem
 - e.g. cross-build:
`$ apt-get install gcc-riscv64-linux-gnu`
`$ riscv64-linux-unknown-gnu-gcc -o test test.c`

Special thanks

- Kurt Keville (CSAIL, MIT)
 - Sw testing, invitation, logistics, ... and encouragement!
- Christian Svensson
 - For starting the OpenRISC or1k port, being a pleasure to work with and getting me hooked!
- Bytemark (Hosting provider)
 - For sponsoring hardware (this project and Debian)
- Family
 - For their support, love and encouragement (also w/ this specific project and travel)

Also thanks

- Debian contributors / community
 - For creating an amazingly useful and fun project
- RISC-V contributors / community
 - For the patience of the people from the community in dealing with my requests for help, comments and patches
 - For creating another amazingly useful and fun project
 - And for the courage of starting from scratch

Questions / comments



Manuel A. Fernandez Montecelo <maf@debian.org>
<https://wiki.debian.org/RISC-V>

Artwork based on “Green_Debian_2”
<http://collab.debian.net/si0ux/gallery/green-debian-2si?view=true>
<http://www.linuxs.mine.nu/volkany/>