Fedora on RISC-V 64-bit

Introduction
Brief Overview
Latest Developments

Presented by
David Abdurachmanov
Independent Software Engineer
Lead of Fedora/RISCV
Jan 2nd, 2019 / FOSDEM 2019

History / Bootstrap

Why native build, but not cross-compiling

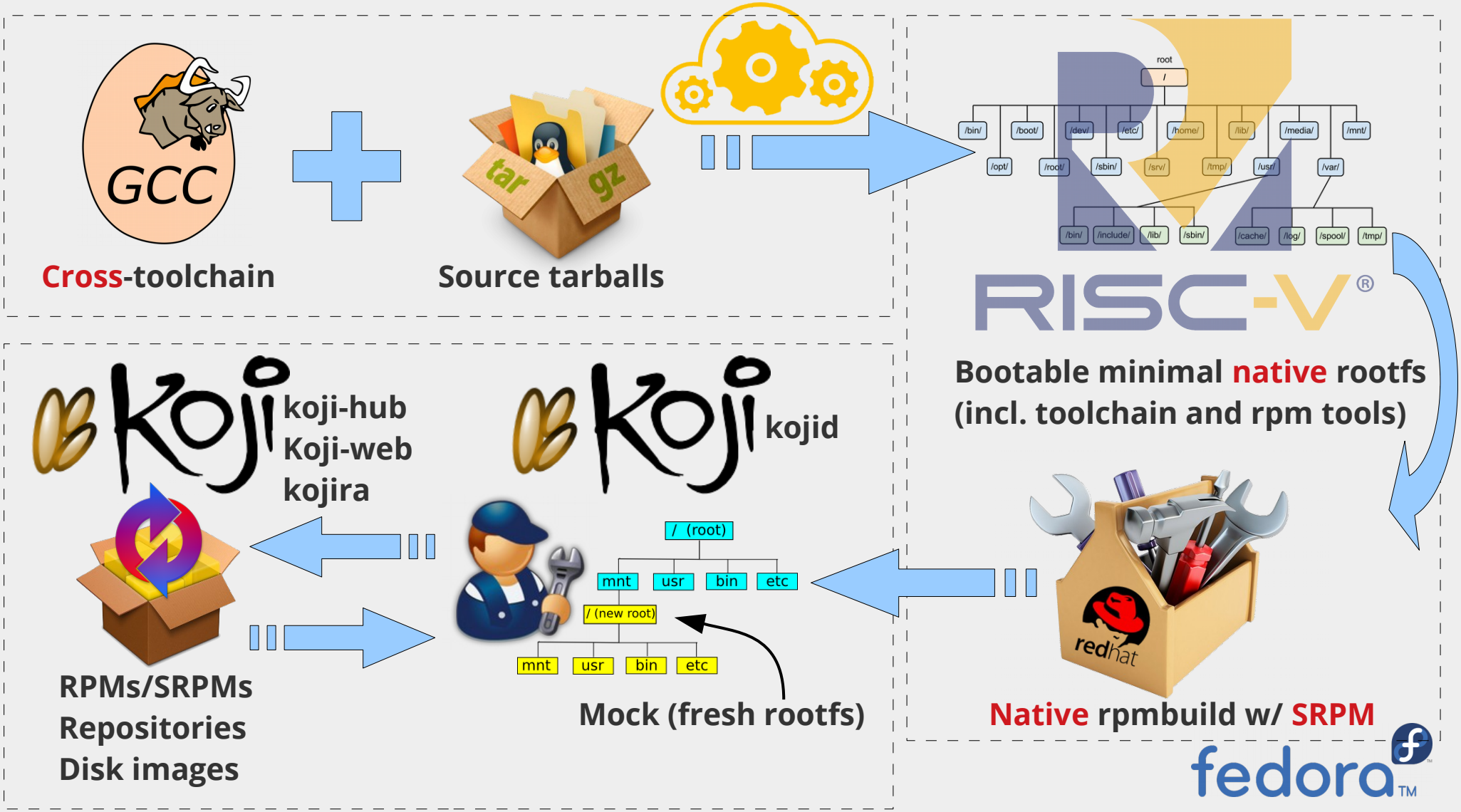
- Almost all the software are originally designed as native build
  - **Save time & effort for writing/maintaining compilation scripts & environments**
- Some software may attempt to run itself or its utilities during compilation stage
  - **Keep It Simple, Stupid (KISS)**
- Host performance is typically much better then the target
  - **The target platform (QEMU riscv64) is powerful enough for compilation jobs**

fedora™

What is Fedora bootstrap?

- Chicken and Egg situation
  - Fedora release is built upon the previous release, but this **cannot** be done for a brand new architecture, because we don’t have previous release
- Breakout
  - We **cross-compile** enough software/packages to “**bootstrap**” **minimal root file-system** for the new architecture (this might be a tricky endeavor)

How Fedora bootstrap works?



The 1st bootstrap

- **August 10, 2016 Richard WM Jones** manages to get **rpmbuild** running on riscv64 (RV64GC)
  - Very soon after that **David Abdurachmanov** and **Stefan O'Rear** joins the efforts
- **October 15, 2016** 5000 (~25%) packages of Fedora are built and we have a booting system with systemd

Richard's Blog:

First successful rpmbuild on RISC-V
Fedora/RISC-V, steady progress
Now building Fedora/RISC-V "stage4" disk images
Fedora/RISC-V is finished!

fedora™

Why did this project stop?

- Since Fedora has an **upstream first policy** – it also applies to Fedora/RISCV
- **Nov 13, 2017** the upstream kernel merged RISC-V support: [GIT PULL] RISC-V Port for Linux 4.15 v9
  - **GCC** and **bintutils** were merged earlier
- We needed **glibc** patchset (RISC-V glibc Port) to be merged for the final bootstrap on riscv64 (RV64GC)
  - This **finalized** and **locked** the **ABIs** on which we can depend

The 2nd & 3rd (the final) bootstraps

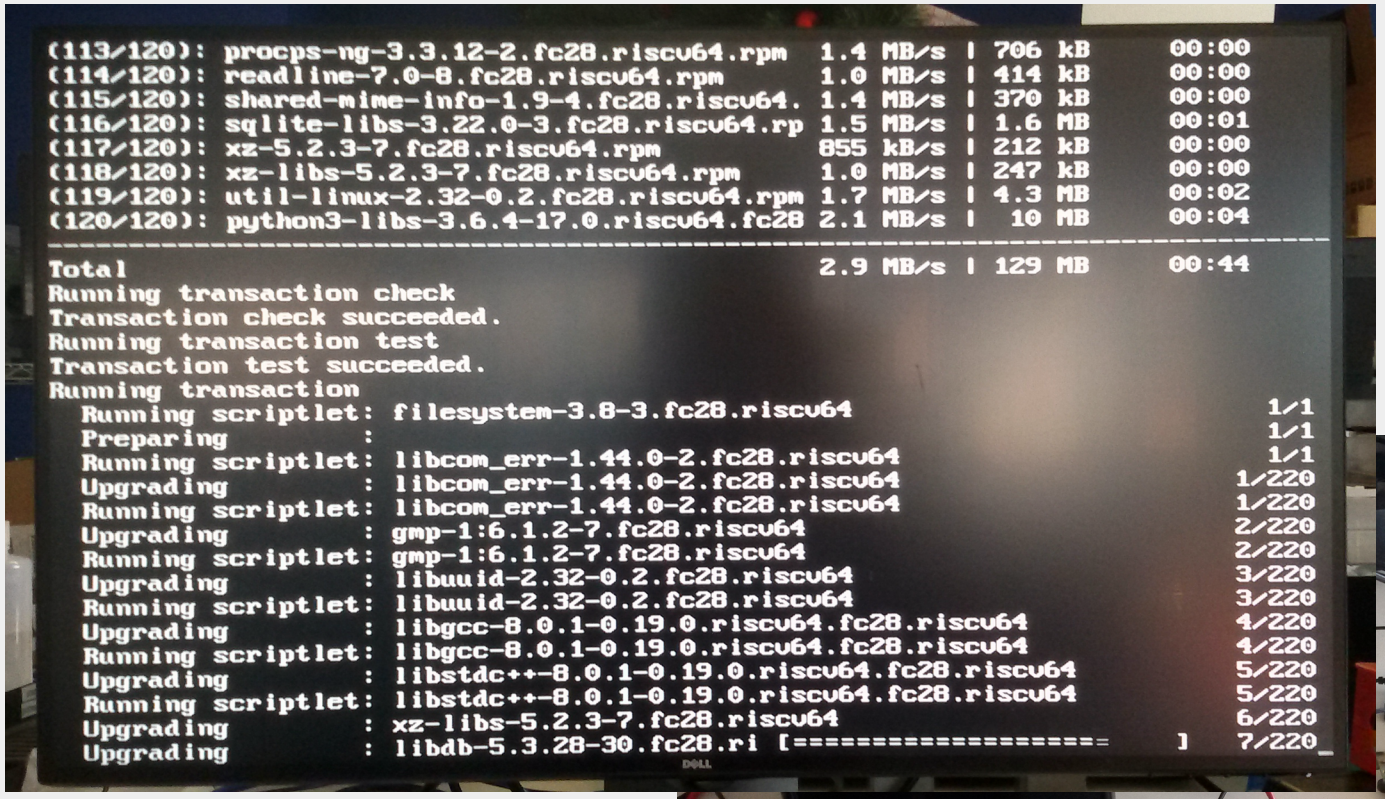
- On **Dec, 2017**, before the **glibc** changes were merged, a rebuild was already underway for Fedora 27/28
- Once the expected (**final**) changes for glibc were merged then the final rebuild could be started
- **Jan 29, 2018** "RISC-V glibc port v7" patchset was merged
- **March, 2018** the final bootstrap is finished
- **Apr 15, 2018** Fedora moves to use **Koji** based infrastructure (default for Fedora) building packages in a same way as other architectures

Richard's Blog:
Fedora/RISC-V: the final bootstrap
Fedora/RISC-V: Runnable stage 4 disk images
Fedora/RISC-V: ssh and dnf working

fedora™

Fedora/RISCV in real life

1st time Fedora on RV64GC hardware



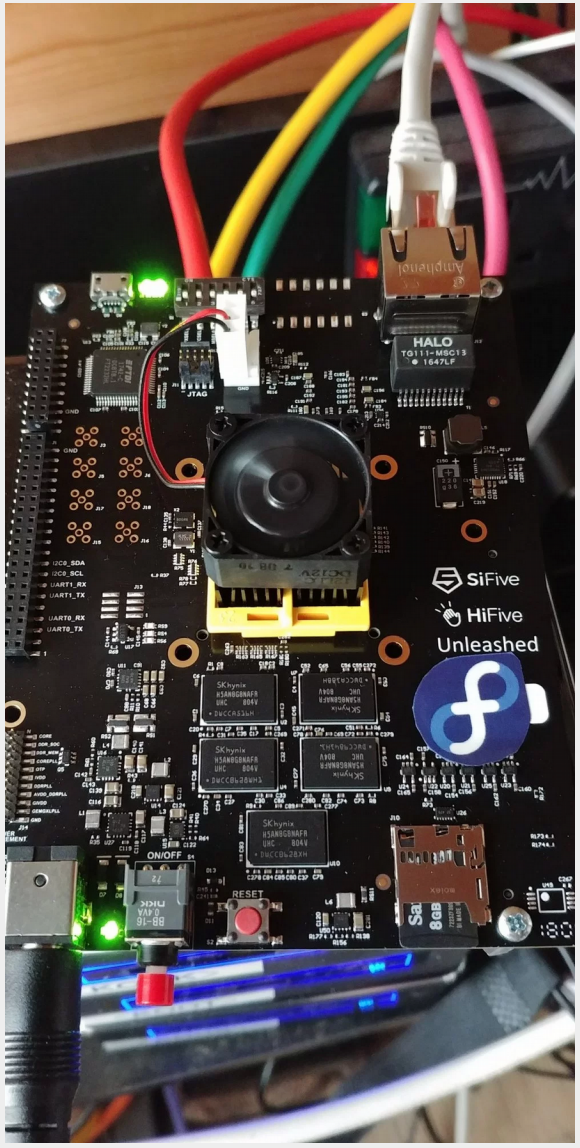
Fedora 28 upgrade running

Richard's Blog:

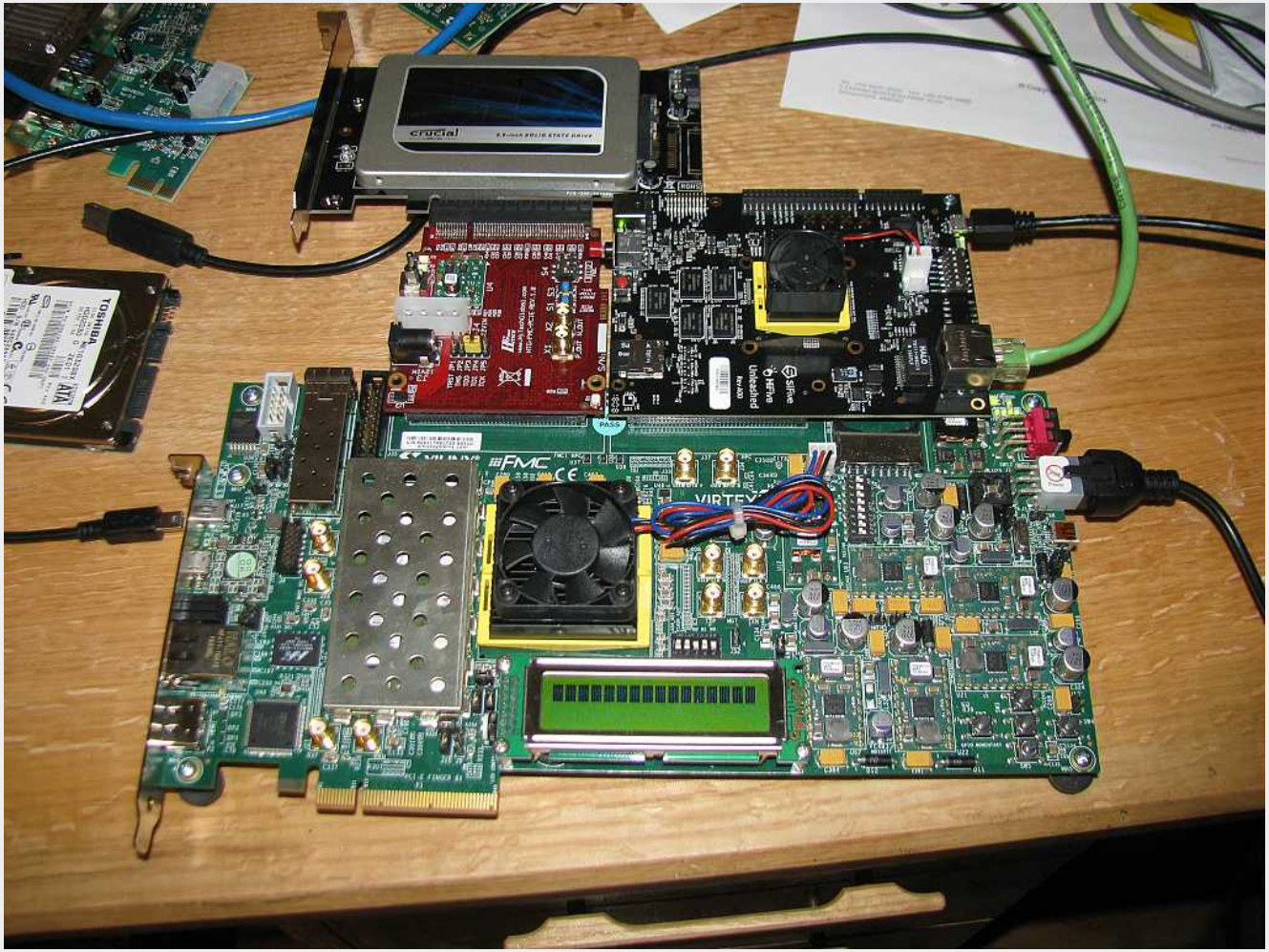
Fedora/RISC-V running on the HiFive Unleashed board

Fedora build boards

Richard (Fedora and private, 2 boards)



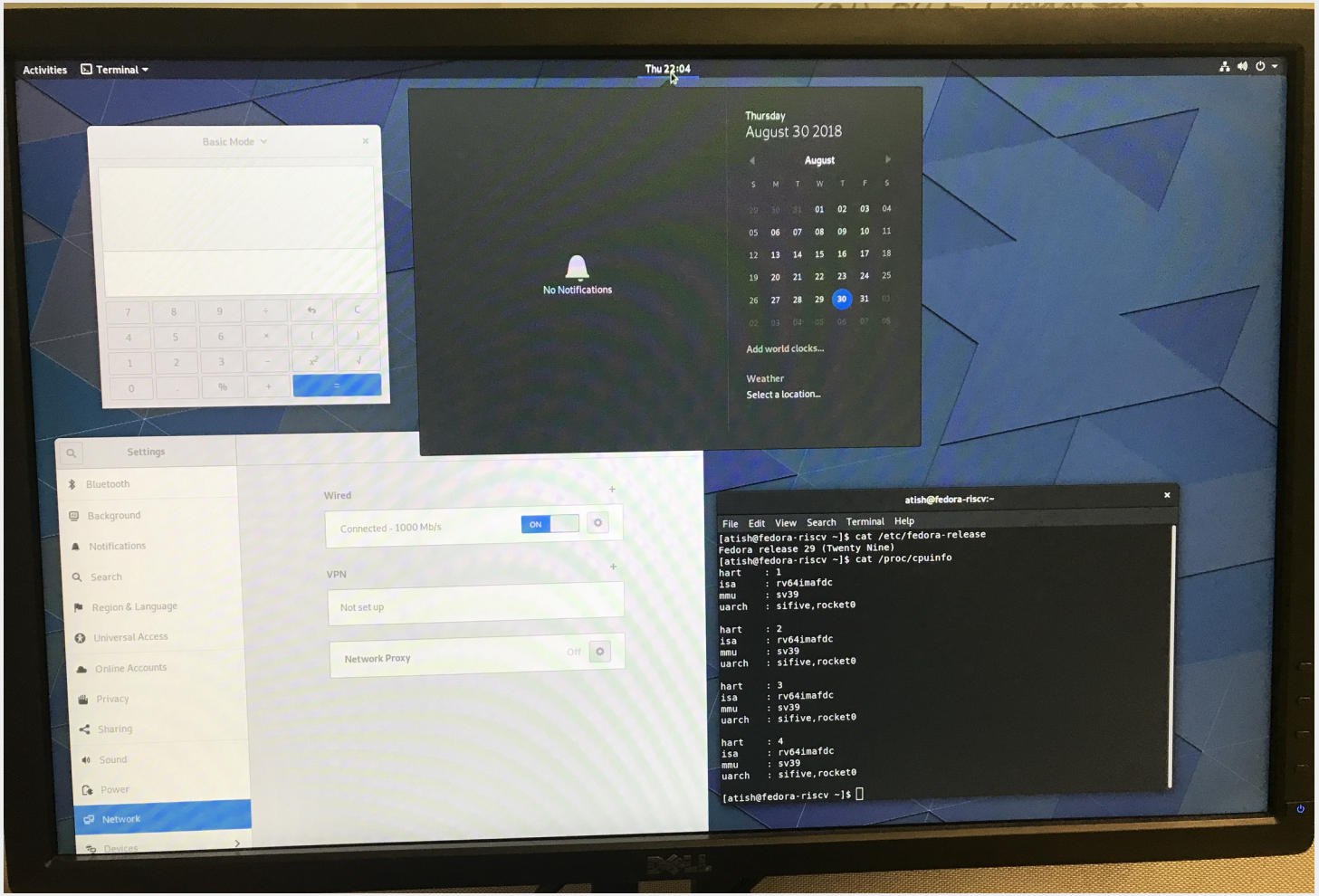
DJ (glibc/Fedora, 1 board)



Richard's Blog:
HiFive Unleashed booting

fedora™

GNOME Desktop w/ Wayland on hardware!

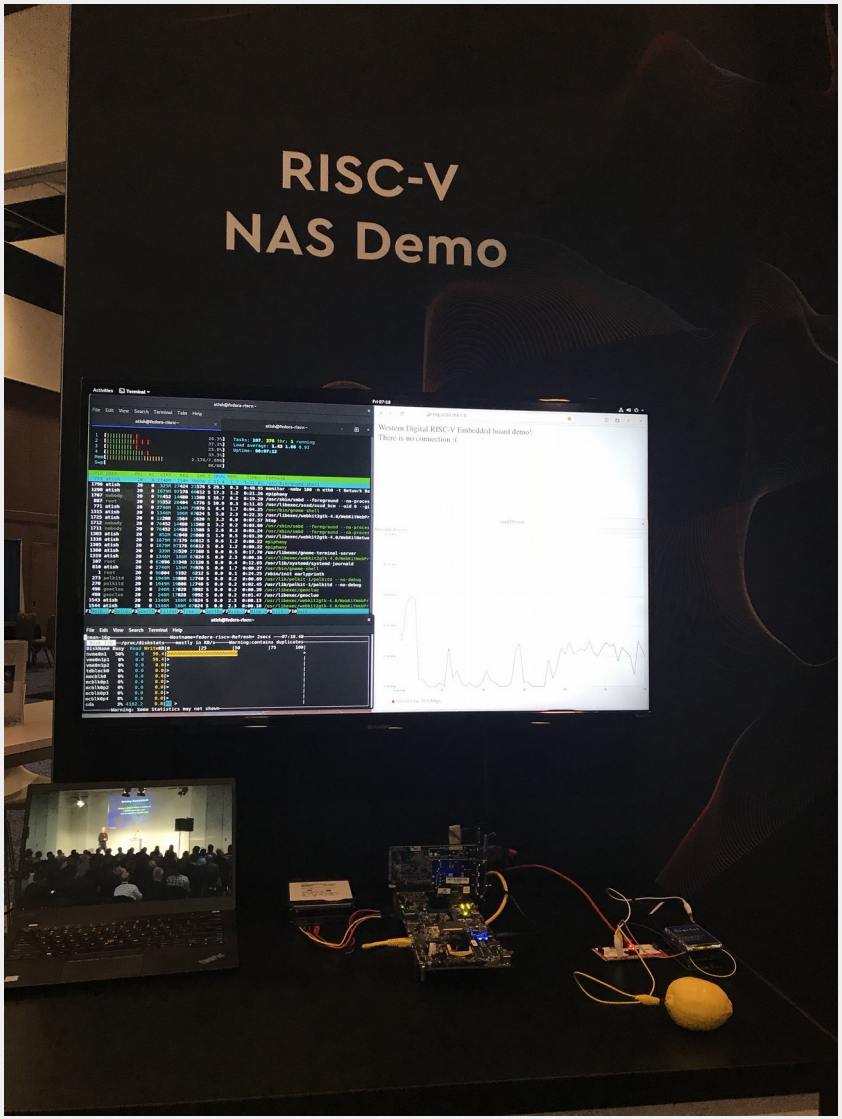


SiFive HiFive Unleashed + Microsemi HiFive Unleashed Expansion + Radeon HD 6450 GPU

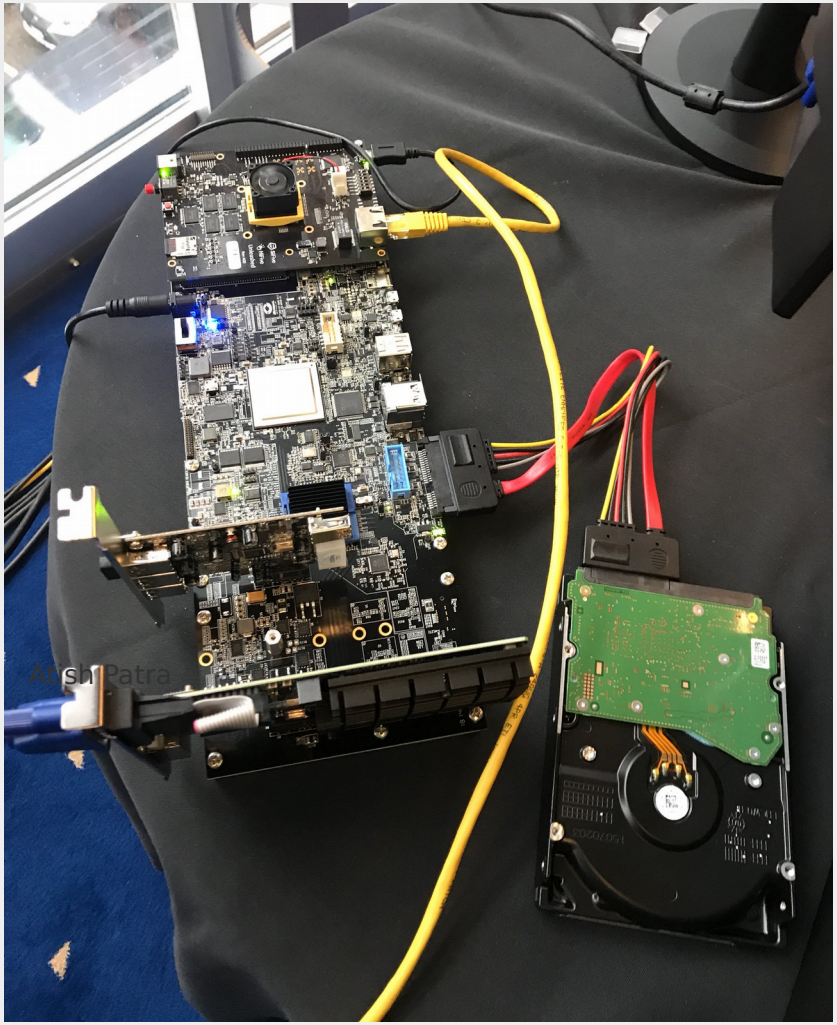
Instructions by Atish Patra (Western Digital Corporation)

fedora™

Fedora 29 for NAS/Media Server

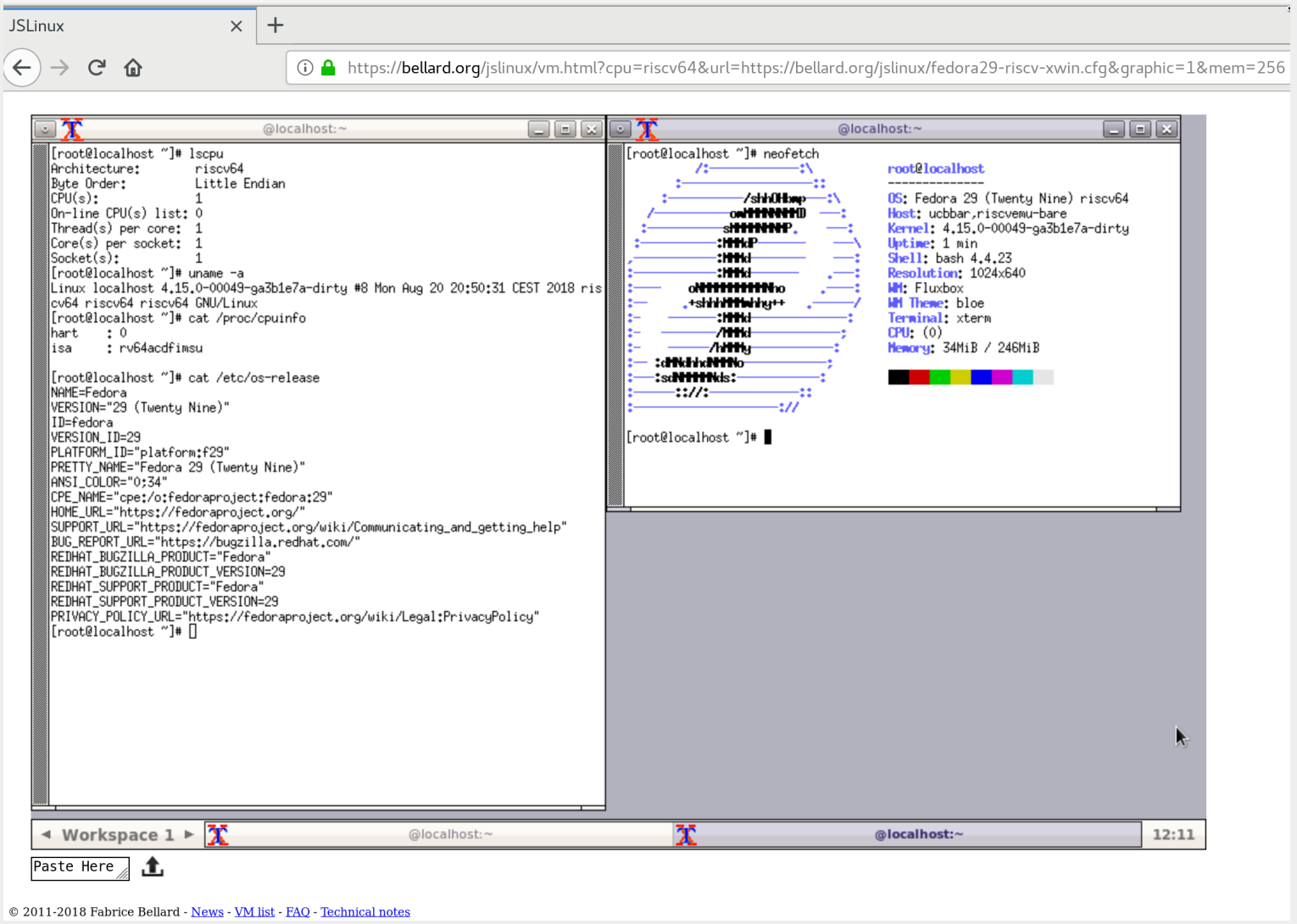


Fedora 29 as Media Server



Demos and pictures (from twitter) from Atish Patra (WDC)

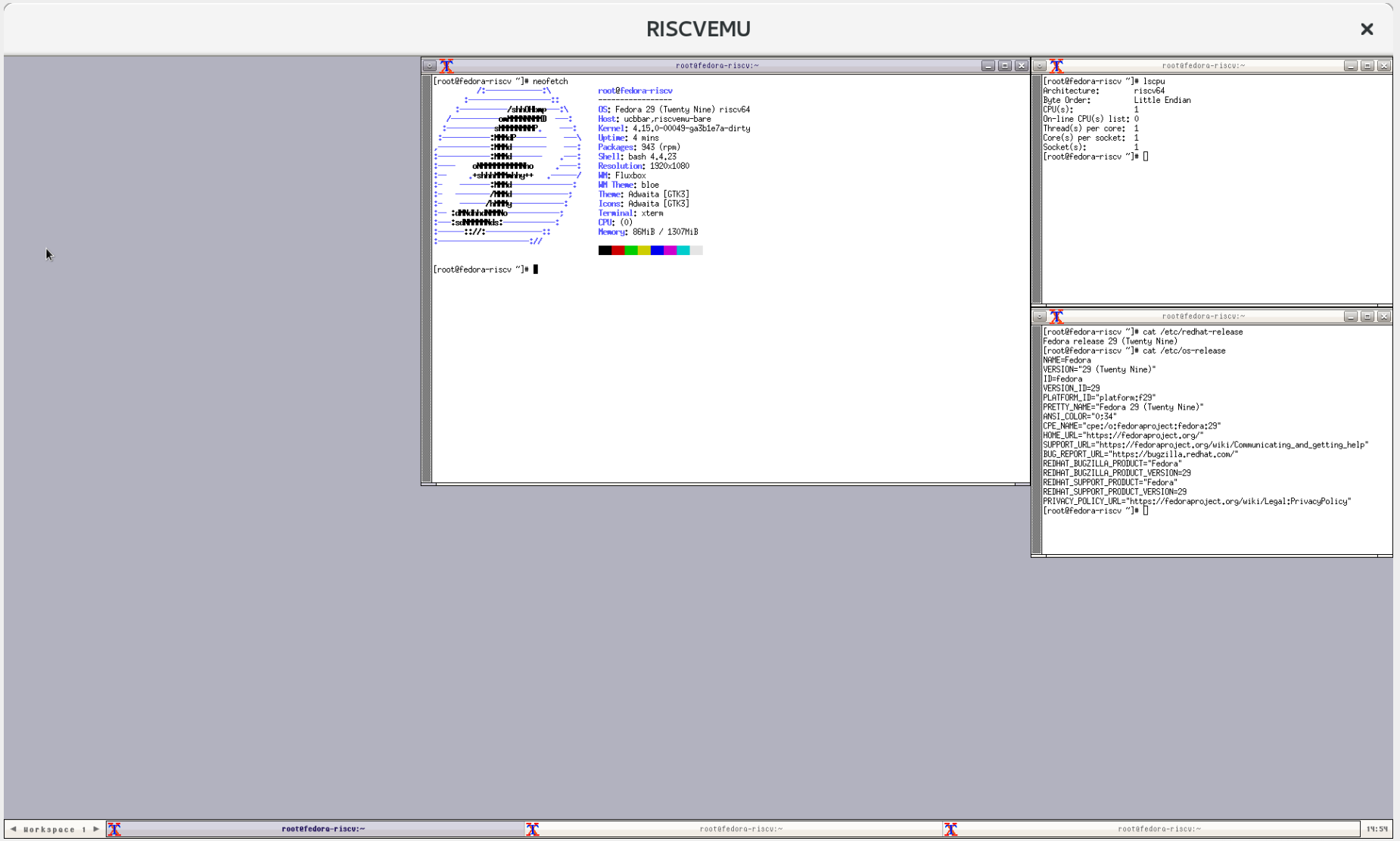
Fedora 29 X11 in a web browser!



Visit https://bellard.org/jslinux/ to try it!

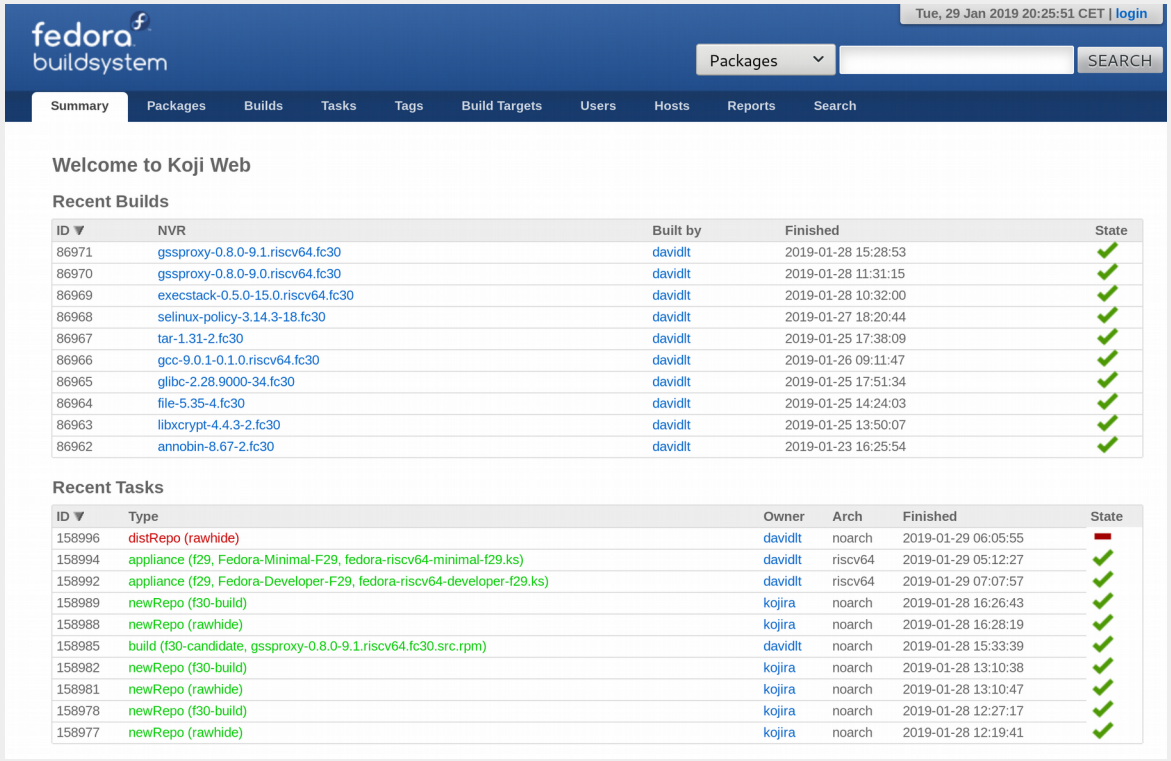
fedora™

Fedora 29 X11 on TinyEMU (RISCVEMU)



The Current State

Koji build farm (fedora.riscv.rocks)



Current build farm:

- **3** SiFive HiFive Unleashed (one with SSD)
- **2** x86_64 nodes for main sever and repository creation
- **1** x86_64 VM with **Ceph** for backup (**restic** based)
- **64** QEMU instances
- **~30** QEMU instances can be added
- VMs are managed by **libvirt**

The repository data (incl. SRPM and debug{info,source} packages) are replicated to the official Fedora infrastructure. It can be used to kickstart a new Koji instance.

NEW: https://dl.fedoraproject.org/pub/alt/risc-v/
OLD: https://fedorapeople.org/groups/risc-v/ (incl. stage4 RPMs and disk images)

Two active projects: **Fedora 29** and **Fedora 30/Rawhide**.

fedora™

Stats

- **24210** successful and **3509** failed builds
  - Best week was **~4500** (20+% of Fedora) successful builds
  - **142798** total RPMs (incl. noarch, debug) produced
- Comparison between arches for **f30** koji tags

	Total packages	**Total built**	**RPMs (arch specific)**	**RPMs (incl. debug*)**
riscv64	22229	20156	15883	34130
aarch64	21782	21621	19984	41119
x86_64	21782	21621	20394	41858

Infrastructure problems

- Our main server (fedora-riscv.tranquillity.se) runs koji-hub, koji-web, NFS (i.e. storage), kojira, database and also kojid (for repo generation)
  - It's 2x 4C/4T E5430 (Q4'07), 16GB RAM and 2x HDDs are struggling: high memory usage, high IO wait, network bw, ...
  - *.tranquillity.se was added into HSTS preload list: users are redirected to HTTPS (self-signed, client certificate authentication) which they cannot easily access
- Our workers (QEMU and boards) regularly fail: e.g. CPU stalls
  - Requires manual hard reset or re-creation of VM (if file-system was damaged)
- Requires constant monitoring and maintenance

fedora™

What's missing?

- No signed RPMs (possible, but needs resource impact assessment)
- No Bodhi (no need, but we keep tags compatible)
- No Pungi (no need, koji handles distribution repositories and disk images)
- Disk images like Workstation or/and Server (needs investigation)
- Modularity support?
- No BLS (Boot Loader Spec) support, similar to armv7hl
- Updating kernel requires manual intervention (bootloader has embedded kernel)
- **%check** (i.e. tests) are globally disabled

fedora™

Custom bits

- Koji-shadow performs the same builds from another koji instance, but we don’t use it
- Custom non-automated script adds packages (but not remove), prepare SRPM and schedule in our koji instance
  - BuildSRPMFromSCM is expensive thus we assemble SRPM in Docker on x86_64
  - Only submits a build if we have required dependencies
  - We only looks into latest successful builds for particular tag (e.g. f30) thus our build order is different and some builds (NVR) might be missing (this creates some problems)
- We have dist-git SCM overlay (contains not yet upstreamed ports or/and changes to SPEC): “**Release:**” incl. “**.X.riscv64.**”

fedora™

Disk images

- 4 different disk images:
  - **Developer:** ~1GB download (fits in 8GB SD) contains everything for development, RPM building, koji builder, minimal X11 environment, major text editors, file management, disk/image working tools, compilers/debuggers
  - **GNOME:** Developer + @gnome-desktop
  - **Minimal:** ~500MB download, @core, @buildsys-build, kernel
  - **Nano:** smaller, @core, kernel and no docs
- Only **Developer** and **Minimal** are scheduled automatically
  - Disk images are built on physical boards because of xz compression

Where to get disk images?

- Post-processed & minimally tested:
  - https://dl.fedoraproject.org/pub/alt/risc-v/disk-images/
    - Naked filesystem, use **/dev/vda**
    - Restored SELinux context and set **enforcing=1**
    - It was booted a few times with QEMU or/and libvirt
- Latest disk images from koji (might have issues):
  - http://fedora.riscv.rocks/koji/tasks?order=-completion_time&state=closed&view=flat&method=createAppliance
    - Use **/dev/vda1**
- Old stage4 disk images with static (not recommended)
  - https://fedorapeople.org/groups/risc-v/disk-images/

fedora™

Supported targets

- Virtual:
  - **QEMU**: official releases work
  - **libvirt/QEMU**: official releases work
  - **TinyEMU**: requires modification for recent changes (BBL and kernel separation, no support for separate initrd file)
- Physical:
  - **SiFive HiFive Unleashed**: upstream kernel lacks support, thus custom kernel build is required, which we don't have for the latest disk images
- All instructions are available on Architectures/RISC-V/Installing wiki page

fedora™

Few (potentially annoying?) bits

- Make sure your **config.{guess,sub}** are **up-to-date 2018-07-03** (latest RISC-V related change)
- RISC-V **only** has **word-sized atomics** thus **requires libatomic**
  - If **-pthread** is used GCC will add **--as-needed -latomic –no-as-needed** (done via LIB_SPEC)
  - It will not do anything if you use **-lpthread** instead
  - GCC will not inline those libatomic calls thus compilation fails with undefined references
- /* The RISC-V ABI specifies that the dynamic section has to be read-only. */ #define DL_RO_DYN_SECTION 1
  - This seems to be wrong, a leftover from MIPS files (in glibc)
- Libraries live in **/usr/lib64/lp64d/** (symlink to **/usr/lib64/)**

Future plans

- Main server migration (HW RAID10, 3x RAM)
- Finish upstreaming audit (userspace)
- Finish upstreaming SECCOMP (kernel & userspace)
- Upstream Dlang related changes
- Upstream changes from our dist-git SCM overlay
- Add Ada support to GCC, then look into GHC and FPC
- Enable global tests (%check)
- Enable SCM-to-SRPM tasks in Koji
- Improve “shadow koji” functionality
- Replace BBL with U-Boot + OpenSBI
- QEMU master builds for x86_64 (COPR) and riscv64

fedora™

Demo

Thank you ..

The GCC Compile Farm Project

.. and countless other individuals and companies, who have contributed to RISC-V specifications and software eco-system!

Questions?

Contact:
david.abdurachmanov@gmail.com
davidlt @ freenode / #fedora-riscv
twitter: @bitmask_reg