RISC-V软件与工具链的现状及对未来的展望

孙铁群

2019.12.20

UC TECHIP
优矽科技

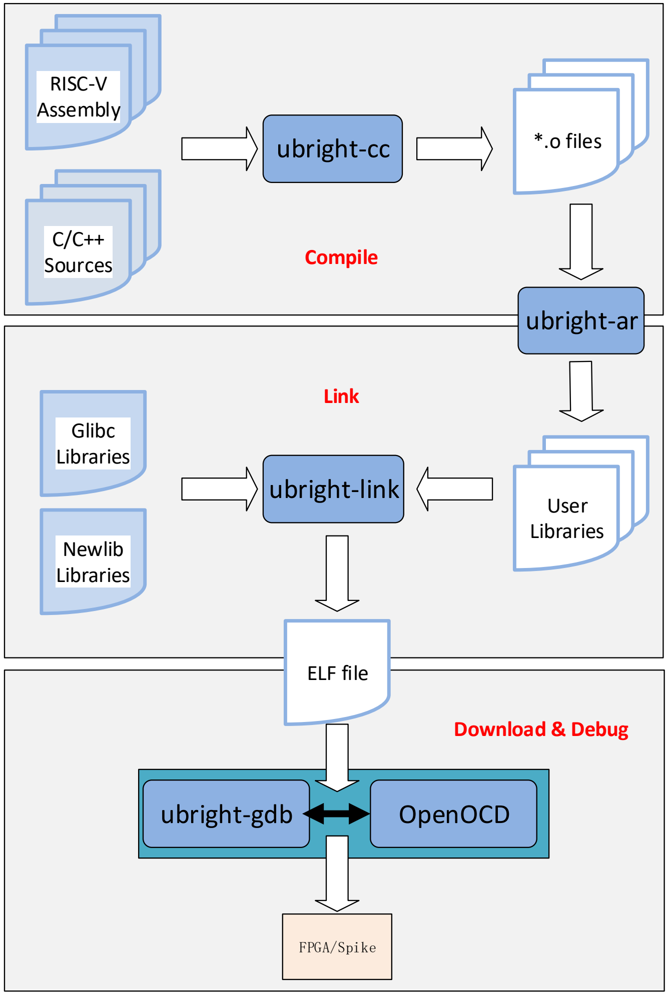
ETCC
2019年中国嵌入式技术大会
EMBEDDED TECHNOLOGY
Conference China 2019

提纲

- 什么是工具链？
- RISC-V有啥工具链可用？
- RISC-V编译器的性能如何？
- 现有工具链的局限性有哪些？
- 我们的解决方案是？

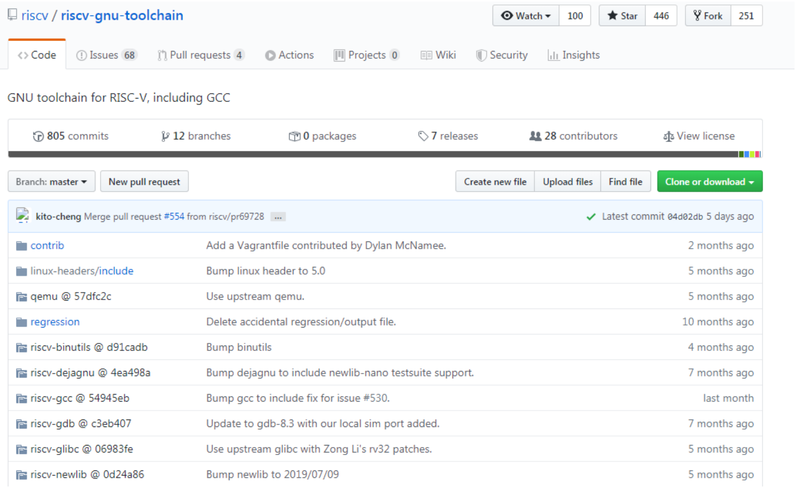
如何理解工具链的定义？

- 作用
  - 一组用于制作、调试软件的工具
- 组件
  - compiler
  - linkers
  - libraries
  - debugger
  - other tools …
- 例子
  - RISC-V GNU Toolchain
  - ARM GNU Toolchain
  - LLVM
  - Plan9


RISC-V Assembly
C/C++ Sources
ubright-cc
*.o files
Compile
ubright-ar
Link
Glibc Libraries
Newlib Libraries
ubright-link
User Libraries
ELF file
Download & Debug
ubright-gdb
OpenOCD
FPGA/Spike

GNU Toolchain现状

- GCC
  - 基于RISC-V Spec v2.2
  - 支持RV32IMAFDC/RV64IMAFDC
  - 主要由SiFive、Andes维护
  - https://github.com/riscv/riscv-gcc
- bintuils & GDB
  - 基于RISC-V Spec v2.2
  - 支持RV32IMAFDC/RV64IMAFDC
  - 支持RISC-V Vector v0.8
  - 主要由SiFive、Andes维护
  - https://github.com/riscv/riscv-binutils-gdb
- Newlib
  - 支持RV32/RV64
  - https://github.com/riscv/riscv-newlib
- Glibc
  - 支持RV32/RV64
  - https://github.com/riscv/riscv-glibc


riscv / riscv-gnu-toolchain

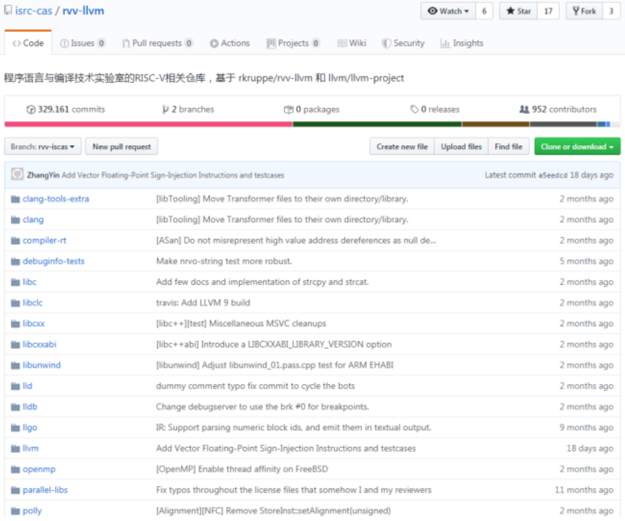
GNU toolchain for RISC-V, including GCC

805 commits · 12 branches · 0 packages · 7 releases · 28 contributors · View license

kito-cheng Merge pull request #554 from riscv/pr69728 … ✓ Latest commit 04d02db 5 days ago

contrib	Add a Vagrantfile contributed by Dylan McNamee.	2 months ago
linux-headers/include	Bump linux header to 5.0	5 months ago
qemu @ 57dfc2c	Use upstream qemu.	5 months ago
regression	Delete accidental regression/output file.	10 months ago
riscv-binutils @ d91cadb	Bump binutils	4 months ago
riscv-dejagnu @ 4ea498a	Bump dejagnu to include newlib-nano testsuite support.	7 months ago
riscv-gcc @ 54945eb	Bump gcc to include fix for issue #530.	last month
riscv-gdb @ c3eb407	Update to gdb-8.3 with our local sim port added.	7 months ago
riscv-glibc @ 06983fe	Use upstream glibc with Zong Li's rv32 patches.	5 months ago
riscv-newlib @ 0d24a86	Bump newlib to 2019/07/09	5 months ago

LLVM现状



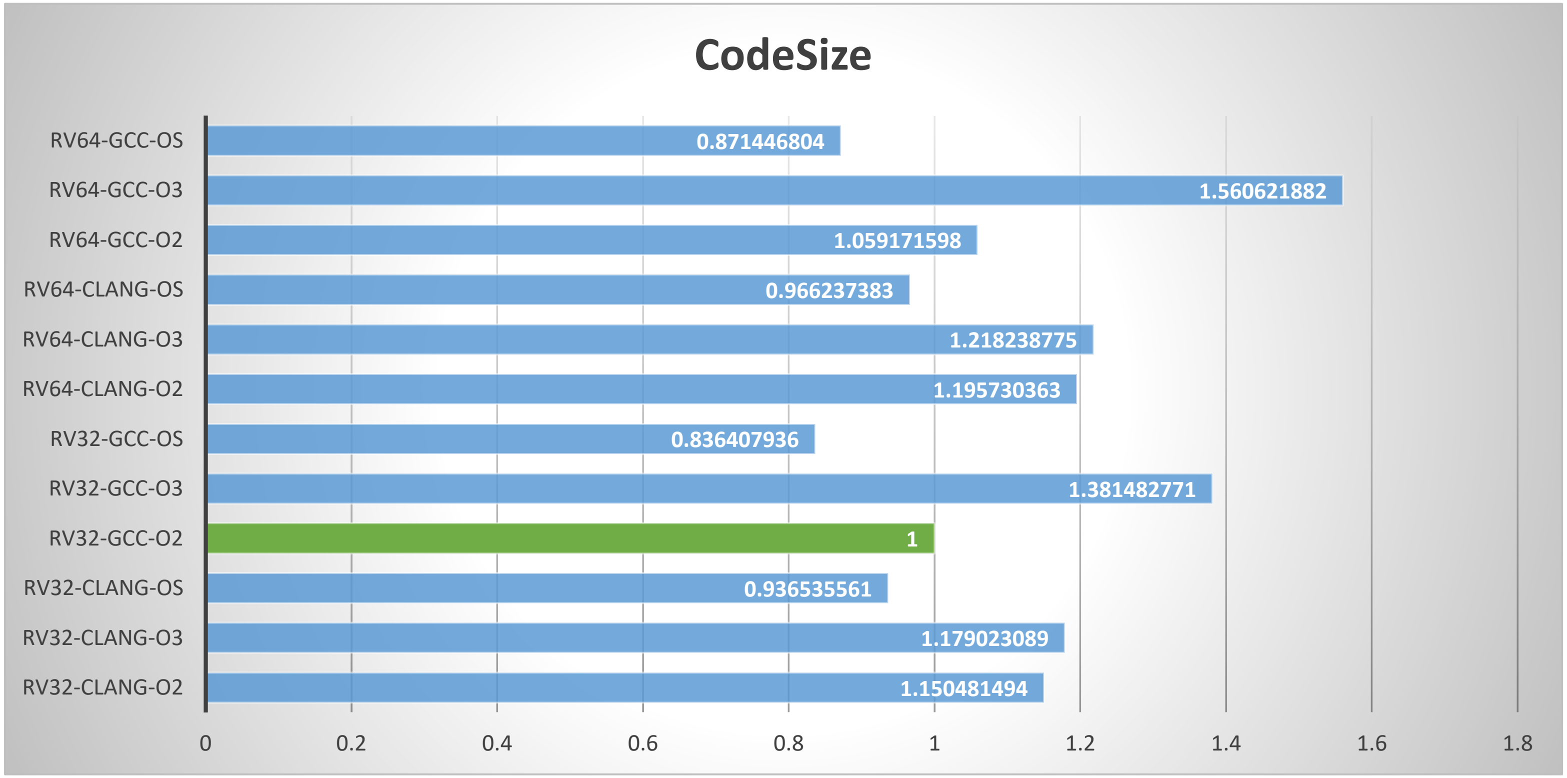
程序语言与编译技术实验室的RISC-V相关仓库，基于 rkruppe/rvv-llvm 和 llvm/llvm-project

ZhangYin Add Vector Floating-Point Sign-Injection Instructions and testcases — Latest commit a5eedcd 18 days ago

clang-tools-extra	[libTooling] Move Transformer files to their own directory/library.	2 months ago
clang	[libTooling] Move Transformer files to their own directory/library.	2 months ago
compiler-rt	[ASan] Do not misrepresent high value address dereferences as null de…	2 months ago
debuginfo-tests	Make nrvo-string test more robust.	5 months ago
libc	Add few docs and implementation of strcpy and strcat.	2 months ago
libclc	travis: Add LLVM 9 build	2 months ago
libcxx	[libc++][test] Miscellaneous MSVC cleanups	2 months ago
libcxxabi	[libc++abi] Introduce a LIBCXXABI_LIBRARY_VERSION option	2 months ago
libunwind	[libunwind] Adjust libunwind_01.pass.cpp test for ARM EHABI	2 months ago
lld	dummy comment typo fix commit to cycle the bots	2 months ago
lldb	Change debugserver to use the brk #0 for breakpoints.	2 months ago
llgo	IR: Support parsing numeric block ids, and emit them in textual output.	9 months ago
llvm	Add Vector Floating-Point Sign-Injection Instructions and testcases	18 days ago
openmp	[OpenMP] Enable thread affinity on FreeBSD	2 months ago
parallel-libs	Fix typos throughout the license files that somehow I and my reviewers	11 months ago
polly	[Alignment][NFC] Remove StoreInst::setAlignment(unsigned)	2 months ago

- LLVM
  - 基于RISC-V Spec v2.2
  - 支持RV32IMAFDC/RV64IMAFDC
  - 主要由LowRISC维护
  - 9.0.0以上版本默认支持RISC-V
    - http://releases.llvm.org/download.html#9.0.0
  - PLCT实验室正在实现对Vector（v0.7.1）指令的支持
    - https://github.com/isrc-cas/rvv-llvm
- LLD
  - 8.0.0以上版本开始支持RISC-V
  - 比GNU Linker更快、更小
  - 主要由Andes维护

CodeSize: vs

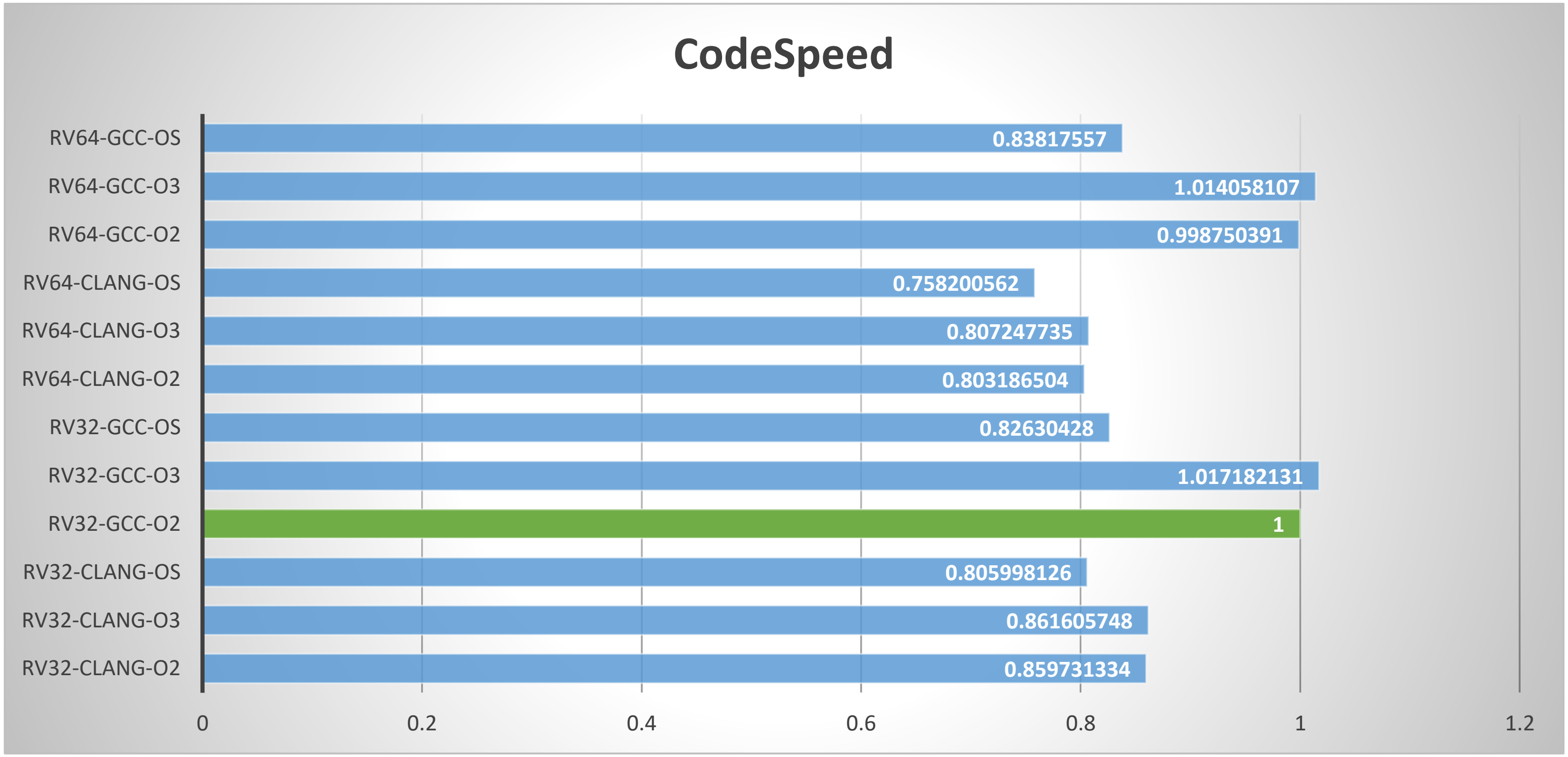


Smaller is better
Benchmark is Coremark

RV64 CodeSize > RV32 CodeSize

GCC is better!

CodeSpeed: vs

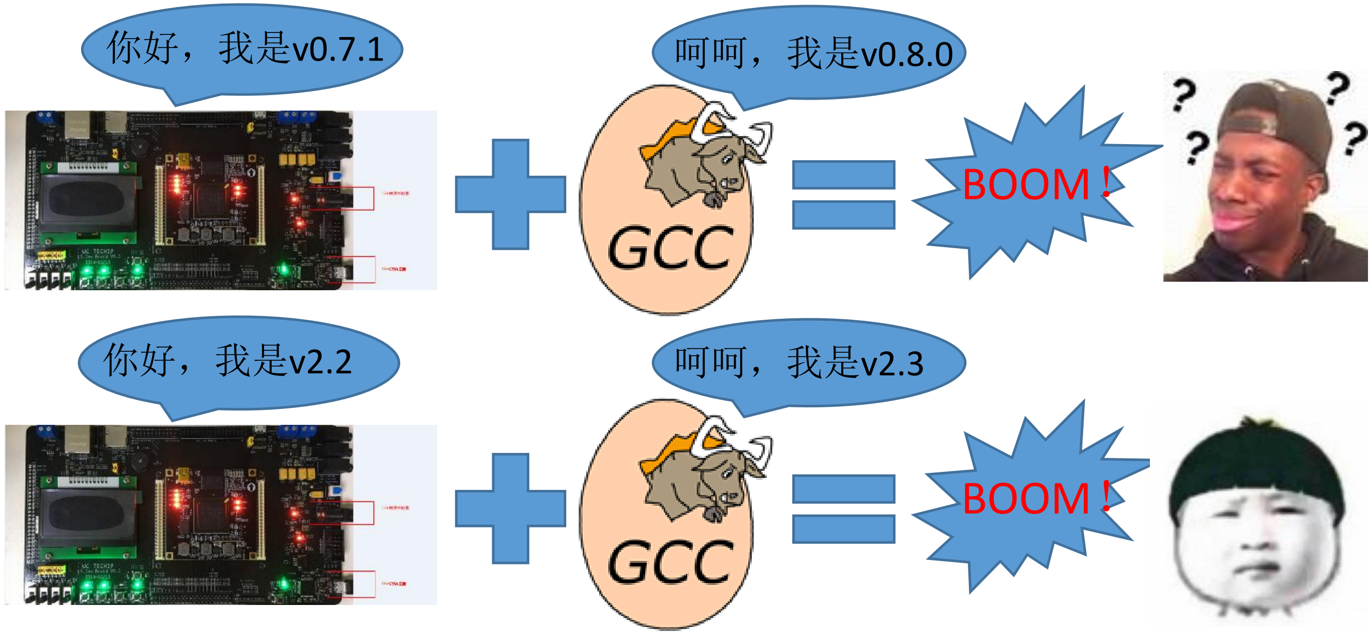


Larger is better
Benchmark is Coremark

GCC is better!

RISC-V Spec版本迭代带来的问题

- 如何适配不同版本的硬件和软件工具链?
  - 例如，RISC-V Vector Spec一直在更新中……



你好，我是v0.7.1

呵呵，我是v0.8.0

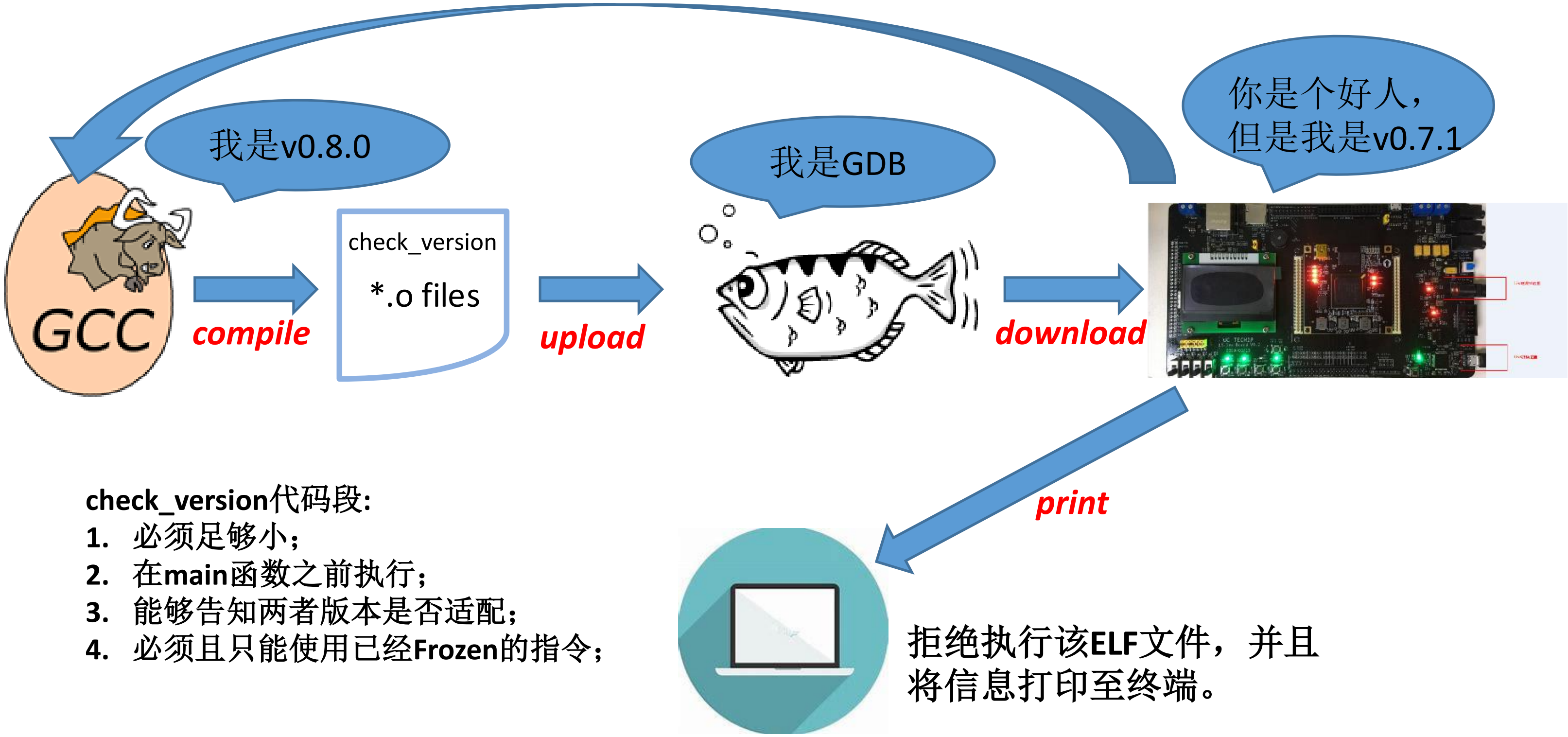
你好，我是v2.2

呵呵，我是v2.3

Debug了半天，发现竟然是版本不匹配！！！

RISC-V Spec版本迭代带来的问题

- 解决方案1：增加一小段代码检查版本信息？



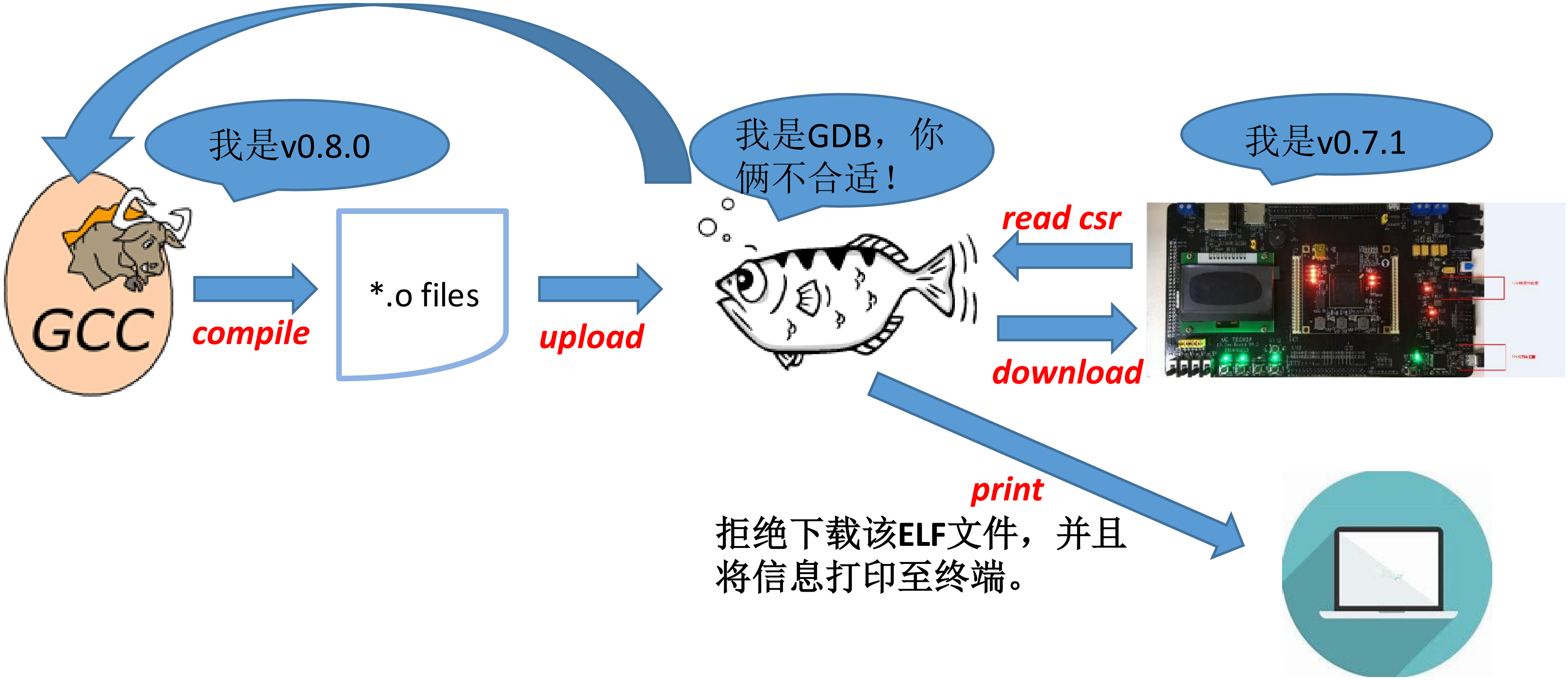
check_version代码段:

1. 必须足够小；
2. 在**main**函数之前执行；
3. 能够告知两者版本是否适配；
4. 必须且只能使用已经**Frozen**的指令；

拒绝执行该**ELF**文件，并且
将信息打印至终端。

RISC-V Spec版本迭代带来的问题

- 解决方案2：增加ELF Attribute, 利用GDB/Linker检查版本信息？



我是v0.8.0

GCC

compile

*.o files

upload

我是GDB，你俩不合适！

我是v0.7.1

read csr

download

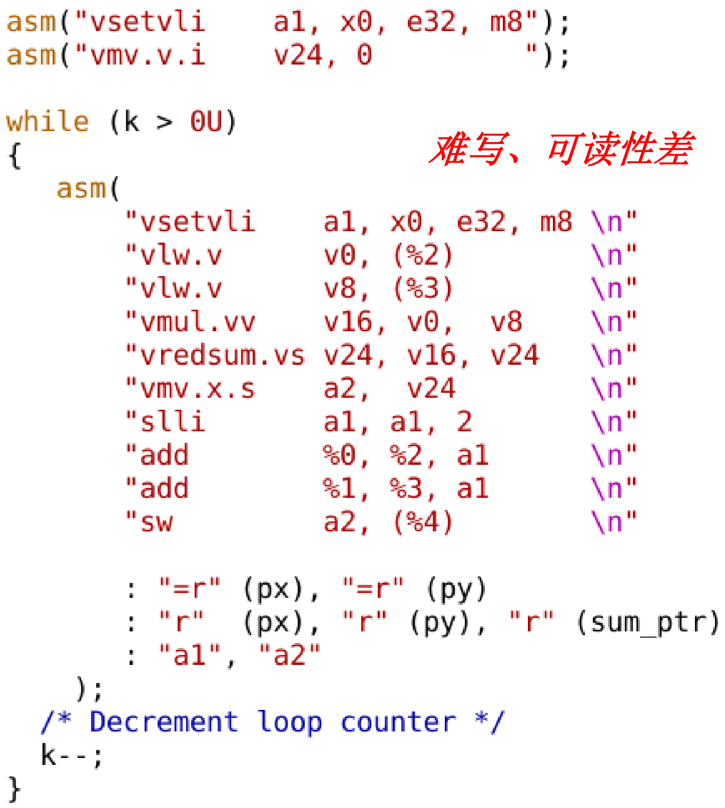
print

拒绝下载该ELF文件，并且
将信息打印至终端。

Ref:https://groups.google.com/a/groups.riscv.org/forum/?nomobile=true#!topic/sw-dev/aZhMG7NIVTk

如何使用RISC-V Vector指令？

- 尴尬的现状：只有汇编器可用

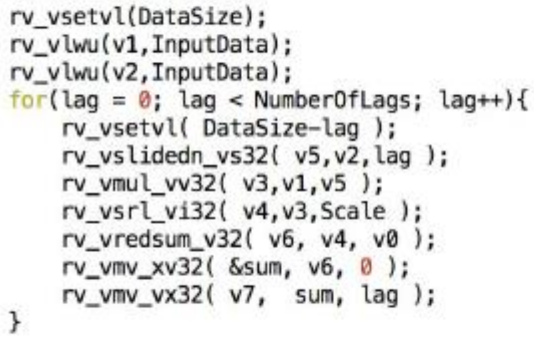


```
asm("vsetvli    a1, x0, e32, m8");
asm("vmv.v.i    v24, 0          ");

while (k > 0U)
{
   asm(
       "vsetvli    a1, x0, e32, m8 \n"
       "vlw.v      v0, (%2)        \n"
       "vlw.v      v8, (%3)        \n"
       "vmul.vv    v16, v0,  v8    \n"
       "vredsum.vs v24, v16, v24   \n"
       "vmv.x.s    a2,  v24        \n"
       "slli       a1, a1, 2       \n"
       "add        %0, %2, a1      \n"
       "add        %1, %3, a1      \n"
       "sw         a2, (%4)        \n"

       : "=r" (px), "=r" (py)
       : "r"  (px), "r" (py), "r" (sum_ptr)
       : "a1", "a2"
   );
  /* Decrement loop counter */
  k--;
}
```

难写、可读性差

intrinsics?

更易维护、可读性高



```
rv_vsetvl(DataSize);
rv_vlwu(v1,InputData);
rv_vlwu(v2,InputData);
for(lag = 0; lag < NumberOfLags; lag++){
    rv_vsetvl( DataSize-lag );
    rv_vslidedn_vs32( v5,v2,lag );
    rv_vmul_vv32( v3,v1,v5 );
    rv_vsrl_vi32( v4,v3,Scale );
    rv_vredsum_v32( v6, v4, v0 );
    rv_vmv_xv32( &sum, v6, 0 );
    rv_vmv_vx32( v7,  sum, lag );
}
```

实现编译器自动向量化难度大，
但是手写汇编又太耗时。
是否可以利用intrinsics的方法？

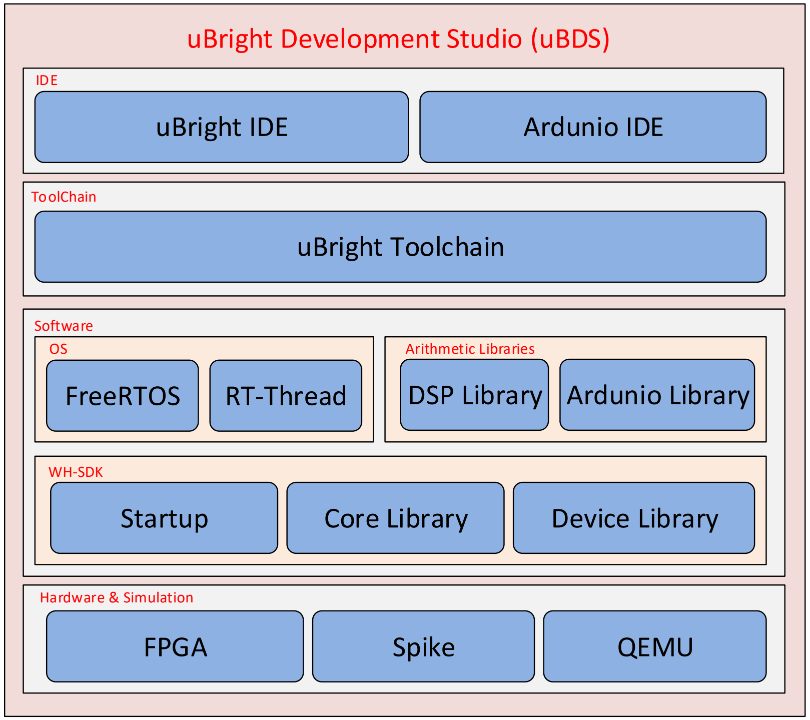
如何屏蔽底层上的差异？

- RISC-V高度灵活性带来的问题----碎片化？
  - 模块化 = 碎片化？
  - 软件能否屏蔽硬件上的差异？

- 如何向上层用户屏蔽底层上的差异？
  - 情况1：不同厂商定义了不同的扩展指令集
  - 情况2：不同厂商使用了不同的工具链
  - 情况3：不同厂商使用了不同的开发平台

统一软件开发平台？

- Manage and Code
  - uBright Eclipse-based IDE is suited for manage bare-metal and real-time operating system projects
  - uBright IDE has integrated Spike simulator
  - uBDS is compatible Ardunio IDE
- Build
  - uBright Toolchain is built on RISC-V GNU Toolchain
  - uBright Toolchain supports excellent optimizations
  - compatible with LLVM
- RTOS
  - compatible with RT-Thread and FreeRTOS
- Libraries
  - supports optimized DSP Library for RISC-V Vector
  - supports Ardunio Library for embedded development
  - supports Core-Library and Device-Library for bare-metal



Thank You!

UC TECHIP
优矽科技

ETCC

2019年中国嵌入式技术大会
EMBEDDED TECHNOLOGY
Conference China 2019