華中科技大學
HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

HUST
OPENATOM CLUB

Robustone
基于Rust的RISC-V反汇编生态

Chen Miao

Robustone 项目组

Rust Why

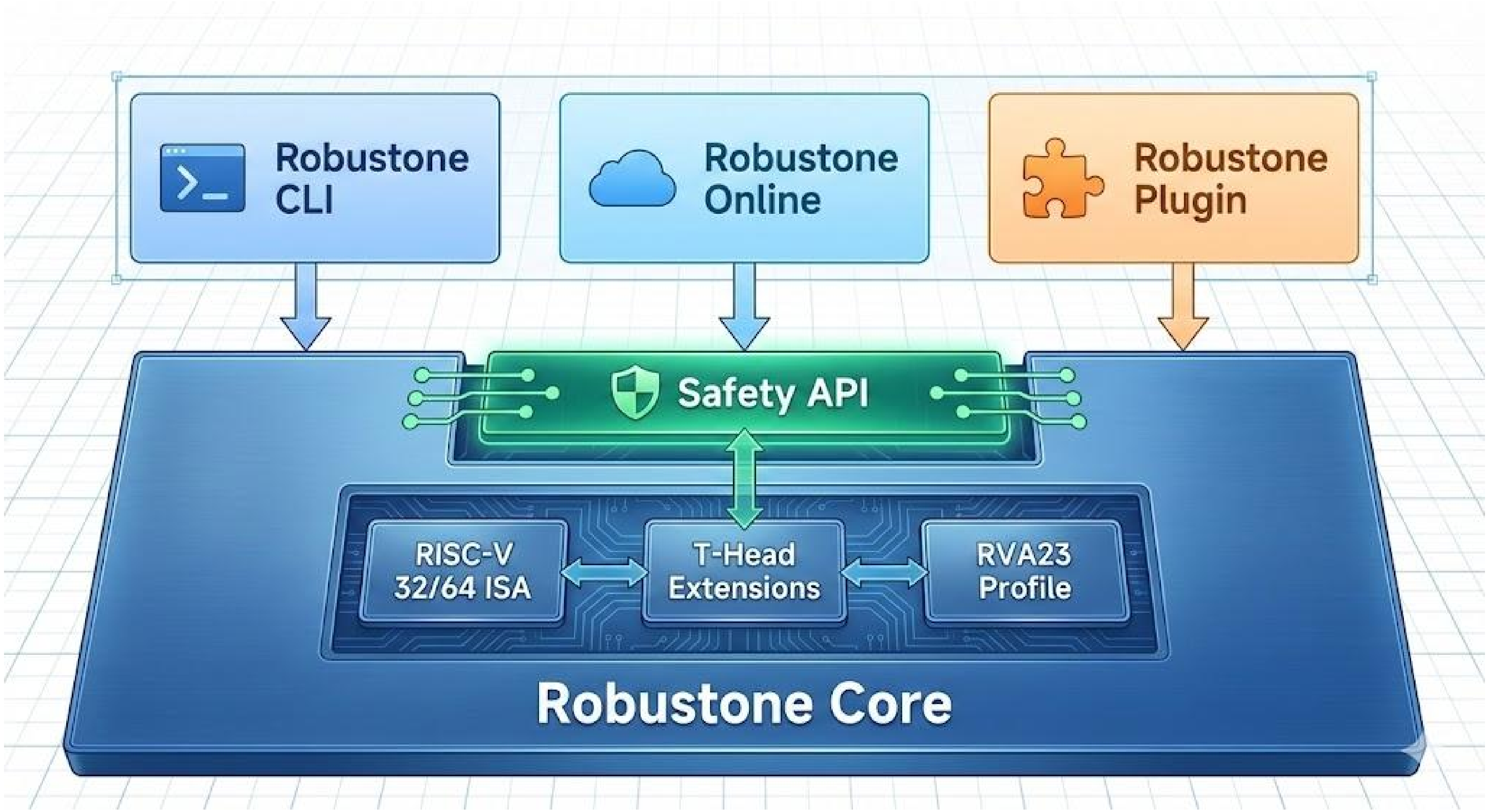
- 内存安全
  - CVE-2017-6952
    - Integer overflow in the cs_winkernel_malloc function in winkernel_mm.c in Capstone 3.0.4
  - CVE-2016-7151
    - Capstone 3.0.4 has an out-of-bounds vulnerability (SEGV caused by a read memory access) in X86_insn_reg_intel in arch/X86/X86Mapping.c.
- 未来趋势
  - Linux基金会将语言安全纳入开源软件安全范畴
  - vivo开源基于Rust自研的蓝河操作系统内核
  - 蚂蚁集团基于Rust打造了“星绽”框内核OS架构

Robustone Why

特性	Capstone	Robustone
RISC-V ISA 支持范围	Only RISC-V 32/64 G	RISC-V 32/64 GC、RVA23、T-Head
可扩展度	/	与RISC-V类似，将每一个扩展设计成为一个模块

- Robustone 的核心优势
  - 完全现代化重写：通过 Rust 语言 从底层设计，构建一个更加模块化和灵活的结构。
  - 语言互操作性与安全性：为其他语言提供安全接口
  - 跨平台/架构优化：更适合进行跨平台、跨架构的使用和部署。

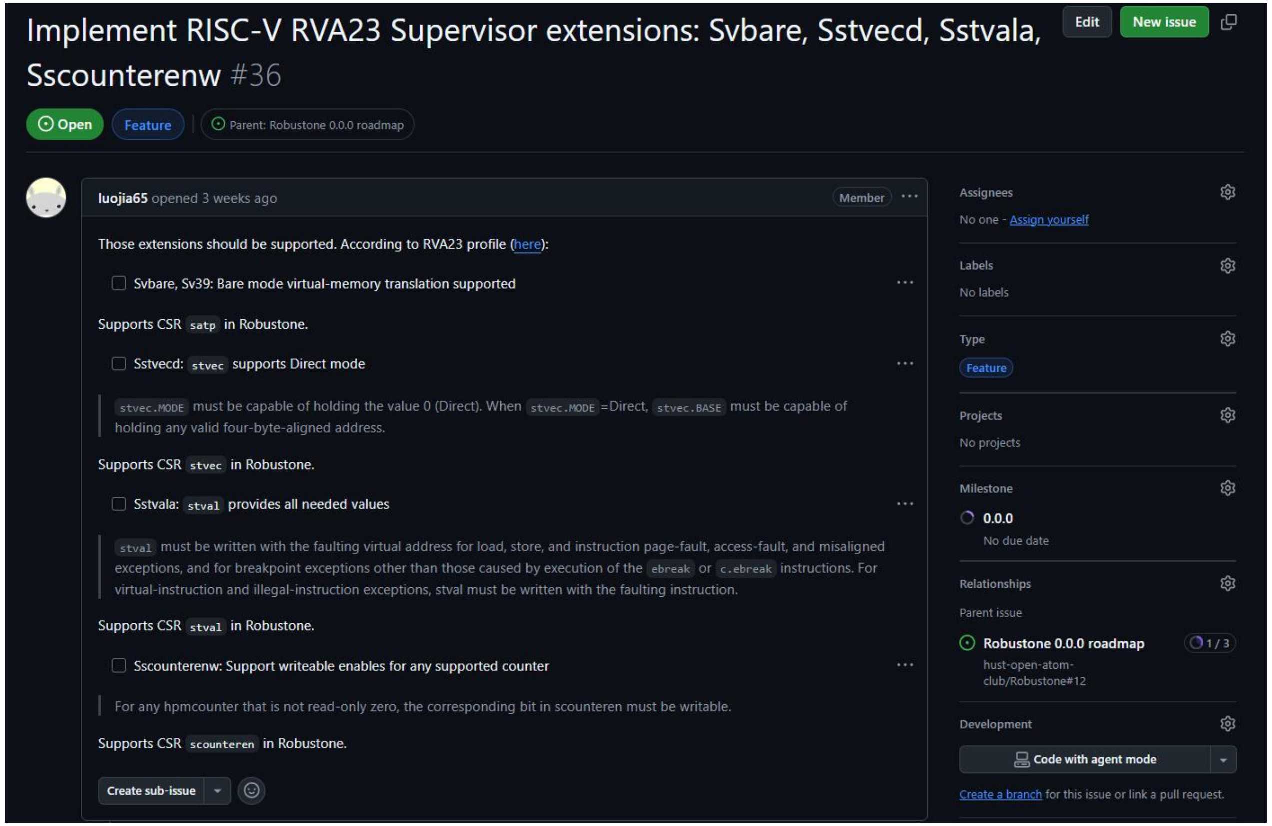
Robustone生态



Robustone Core

Robustone Core 是基于 Capstone 反汇编框架使用 Rust 语言重写的版本，不仅继承了 Capstone 强大的反汇编能力，还进一步利用 Rust 在并发性、性能和安全方面的优势，增强了系统的健壮性和可扩展性。Robustone Core 旨在为上层应用提供模块化的 API 设计，依托 Rust 语言本身的特性，这一结构能够被高效且安全地实现。通过 Rust 书写的 API 能够保证内存安全，从而为 CLI、Online、Plugin 以及其他潜在用户提供稳定可靠的基础服务。

与 Capstone 不同的是，Robustone Core 主要专注于 RISC-V 架构的持续优化与支持，目前重点覆盖 RISC-V 32/64 基础指令集、RVA23 配置以及供应商扩展指令集，力求紧跟 RISC-V 生态发展，为用户提供最新、最完整的 RISC-V 反汇编实现。



Implement RISC-V RVA23 Supervisor extensions: Svbare, Sstvecd, Sstvala, Sscounterenw #36

Parent: Robustone 0.0.0 roadmap

luojia65 opened 3 weeks ago

Those extensions should be supported. According to RVA23 profile (here):

- □ Svbare, Sv39: Bare mode virtual-memory translation supported

Supports CSR `satp` in Robustone.

- □ Sstvecd: `stvec` supports Direct mode

> `stvec.MODE` must be capable of holding the value 0 (Direct). When `stvec.MODE` =Direct, `stvec.BASE` must be capable of holding any valid four-byte-aligned address.

Supports CSR `stvec` in Robustone.

- □ Sstvala: `stval` provides all needed values

> `stval` must be written with the faulting virtual address for load, store, and instruction page-fault, access-fault, and misaligned exceptions, and for breakpoint exceptions other than those caused by execution of the `ebreak` or `c.ebreak` instructions. For virtual-instruction and illegal-instruction exceptions, stval must be written with the faulting instruction.

Supports CSR `stval` in Robustone.

- □ Sscounterenw: Support writeable enables for any supported counter

> For any hpmcounter that is not read-only zero, the corresponding bit in scounteren must be writable.

Supports CSR `scounteren` in Robustone.

Assignees
No one - Assign yourself

Labels
No labels

Type
Feature

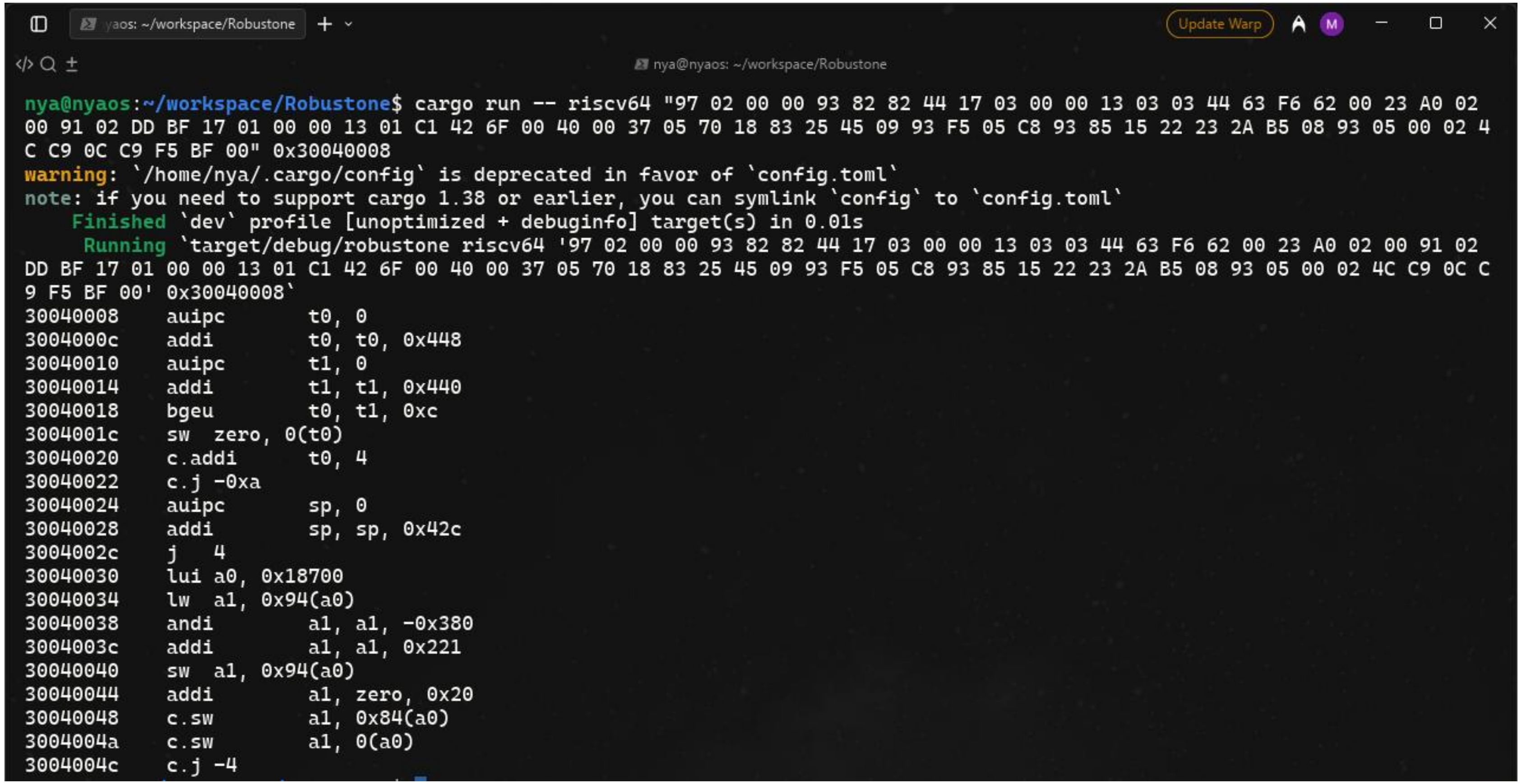
Projects
No projects

Milestone
0.0.0
No due date

Relationships
Parent issue
Robustone 0.0.0 roadmap 1 / 3
hust-open-atom-club/Robustone#12

Development
Create a branch for this issue or link a pull request.

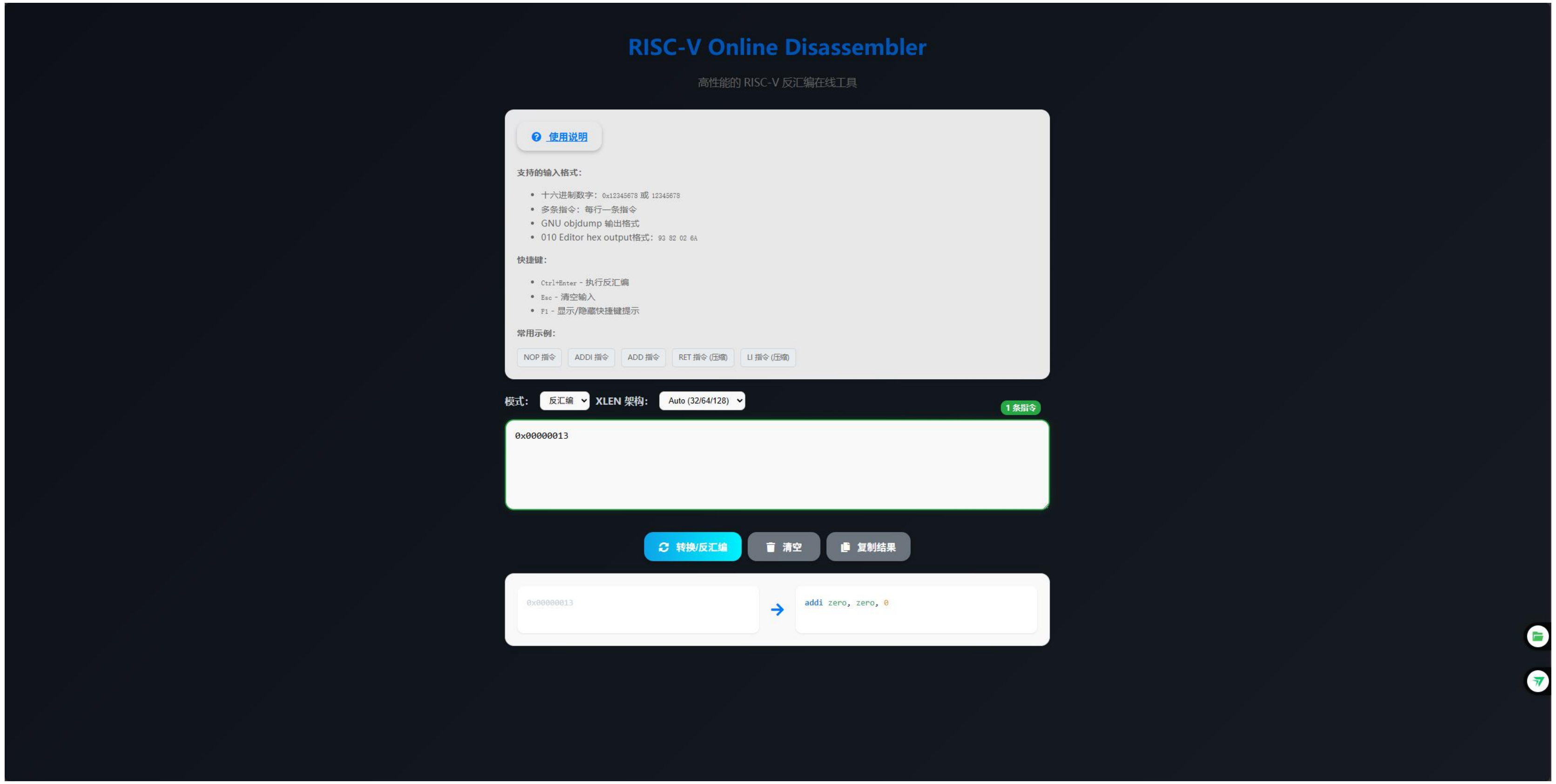
Robustone CLI



```
nya@nyaos:~/workspace/Robustone$ cargo run -- riscv64 "97 02 00 00 93 82 82 44 17 03 00 00 13 03 03 44 63 F6 62 00 23 A0 02
00 91 02 DD BF 17 01 00 00 13 01 C1 42 6F 00 40 00 37 05 70 18 83 25 45 09 93 F5 05 C8 93 85 15 22 23 2A B5 08 93 05 00 02 4
C C9 0C C9 F5 BF 00" 0x30040008
warning: `/home/nya/.cargo/config` is deprecated in favor of `config.toml`
note: if you need to support cargo 1.38 or earlier, you can symlink `config` to `config.toml`
    Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.01s
     Running `target/debug/robustone riscv64 '97 02 00 00 93 82 82 44 17 03 00 00 13 03 03 44 63 F6 62 00 23 A0 02 00 91 02
DD BF 17 01 00 00 13 01 C1 42 6F 00 40 00 37 05 70 18 83 25 45 09 93 F5 05 C8 93 85 15 22 23 2A B5 08 93 05 00 02 4C C9 0C C
9 F5 BF 00' 0x30040008`
30040008    auipc       t0, 0
3004000c    addi        t0, t0, 0x448
30040010    auipc       t1, 0
30040014    addi        t1, t1, 0x440
30040018    bgeu        t0, t1, 0xc
3004001c    sw  zero, 0(t0)
30040020    c.addi      t0, 4
30040022    c.j -0xa
30040024    auipc       sp, 0
30040028    addi        sp, sp, 0x42c
3004002c    j   4
30040030    lui a0, 0x18700
30040034    lw  a1, 0x94(a0)
30040038    andi        a1, a1, -0x380
3004003c    addi        a1, a1, 0x221
30040040    sw  a1, 0x94(a0)
30040044    addi        a1, zero, 0x20
30040048    c.sw        a1, 0x84(a0)
3004004a    c.sw        a1, 0(a0)
3004004c    c.j -4
```

Robustone CLI 是基于 Core 直接编译的原生命令行工具，能够完整支持 RISC-V 32/64 GC 等指令集与扩展，为开发者在本地环境提供高效、精准的反汇编能力。

Robustone Online



RISC-V Online Disassembler

高性能的 RISC-V 反汇编在线工具

❓ 使用说明

支持的输入格式：

- 十六进制数字：`0x12345678` 或 `12345678`
- 多条指令：每行一条指令
- GNU objdump 输出格式
- 010 Editor hex output格式：`93 82 02 6A`

快捷键：

- `Ctrl+Enter` - 执行反汇编
- `Esc` - 清空输入
- `F1` - 显示/隐藏快捷键提示

常用示例：

NOP 指令 | ADDI 指令 | ADD 指令 | RET 指令 (压缩) | LI 指令 (压缩)

模式：反汇编 XLEN 架构：Auto (32/64/128)

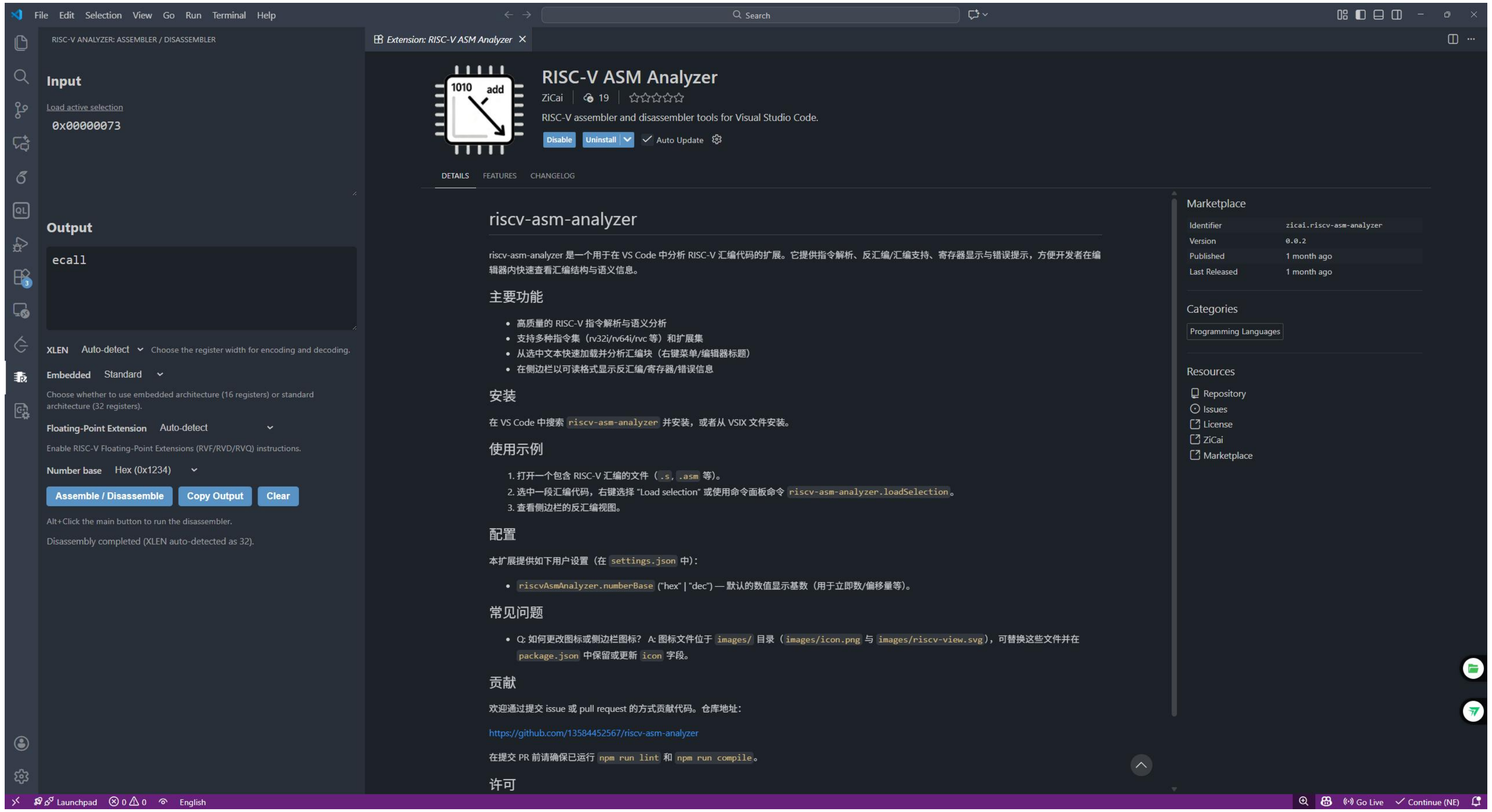
1 条指令

```
0x00000013
```

```
0x00000013 → addi zero, zero, 0
```

基于 Core 构建的 Robustone Online 服务，利用 WebAssembly（WASM）技术将核心反汇编能力安全地运行于浏览器端，直接为网页用户提供在线的反汇编帮助。

Robustone Plugin



RISC-V ANALYZER: ASSEMBLER / DISASSEMBLER

Input

Load active selection

```
0x00000073
```

Output

```
ecall
```

XLEN: Auto-detect. Choose the register width for encoding and decoding.

Embedded: Standard

Choose whether to use embedded architecture (16 registers) or standard architecture (32 registers).

Floating-Point Extension: Auto-detect

Enable RISC-V Floating-Point Extensions (RVF/RVD/RVQ) instructions.

Number base: Hex (0x1234)

Assemble / Disassemble | Copy Output | Clear

Alt+Click the main button to run the disassembler.

Disassembly completed (XLEN auto-detected as 32).

Extension: RISC-V ASM Analyzer

RISC-V ASM Analyzer

ZiCai | 19 | ☆☆☆☆☆

RISC-V assembler and disassembler tools for Visual Studio Code.

Disable | Uninstall | ✓ Auto Update

DETAILS | FEATURES | CHANGELOG

riscv-asm-analyzer

riscv-asm-analyzer 是一个用于在 VS Code 中分析 RISC-V 汇编代码的扩展。它提供指令解析、反汇编/汇编支持、寄存器显示与错误提示，方便开发者在编辑器内快速查看汇编结构与语义信息。

主要功能

- 高质量的 RISC-V 指令解析与语义分析
- 支持多种指令集（rv32i/rv64i/rvc 等）和扩展集
- 从选中文本快速加载并分析汇编块（右键菜单/编辑器标题）
- 在侧边栏以可读格式显示反汇编/寄存器/错误信息

安装

在 VS Code 中搜索 `riscv-asm-analyzer` 并安装，或者从 VSIX 文件安装。

使用示例

1. 打开一个包含 RISC-V 汇编的文件（`.s`，`.asm` 等)。
2. 选中一段汇编代码，右键选择 “Load selection” 或使用命令面板命令 `riscv-asm-analyzer.loadSelection`。
3. 查看侧边栏的反汇编视图。

配置

本扩展提供如下用户设置（在 `settings.json` 中)：

- `riscvAsmAnalyzer.numberBase` ("hex" | "dec") — 默认的数值显示基数（用于立即数/偏移量等)。

常见问题

- Q: 如何更改图标或侧边栏图标？ A: 图标文件位于 `images/` 目录（`images/icon.png` 与 `images/riscv-view.svg`)，可替换这些文件并在 `package.json` 中保留或更新 `icon` 字段。

贡献

欢迎通过提交 issue 或 pull request 的方式贡献代码。仓库地址：

https://github.com/13584452567/riscv-asm-analyzer

在提交 PR 前请确保已运行 `npm run lint` 和 `npm run compile`。

许可

Marketplace

Identifier	zicai.riscv-asm-analyzer
Version	0.0.2
Published	1 month ago
Last Released	1 month ago

Categories

Programming Languages

Resources

- Repository
- Issues
- License
- ZiCai
- Marketplace

Robustone Plugin 作为 Visual Studio Code 的扩展，让开发者能在熟悉的 IDE 中便捷地进行无缝反汇编，极大提升开发效率。

Robustone生态现状

目前，Robustone 主要专注于 RISC-V 架构生态，已实现对 RISC-V 32/64 基础指令集、RVA23 配置及 T-Head 扩展的初步支持。我们清醒地认识到，当前支持的指令集范围仍较为有限，这既是现状，也是我们未来发力的重点。项目目前仍处于早期发展阶段，由社区及俱乐部内部成员积极维护。Robustone 的发展高度依赖社区贡献，我们诚挚欢迎对反汇编技术、RISC-V 生态感兴趣的开发者加入，共同推进指令集支持的广度与深度，并参与 Core 层的重构与优化工作。

同时，我们也非常期待对 API 设计、插件系统开发、Online 服务集成等方向有兴趣的开发者参与进来，共同构建更完善、更易用的 Robustone 工具链生态。无论你是对底层指令解析还是上层应用集成有热情，都欢迎你成为我们社区的一员，一起推动项目向前发展。

致谢

- 姓名：陈磊
- 邮箱：chenmiao@openatom.club
- Github ID：@ChenMiaoi

- 引用：
  - Robustone
  - robustone-online
  - capstone

公众号：开源内核安全修炼
微信号：
kernel_sec_pratice