让我们试试RustSBI！

——2024年RustSBI项目简介

洛佳 2024年10月

华中科技大学 网络空间安全学院

目录

壹

完整的安全引导解决方案

RISC-V的引导程序和固件环境与其它架构是不同的，它除了引导启动内核，还将常驻后台，不断提供内核所需的运行时功能。这对安全引导程序的实现方法给出了新的要求和挑战。

Rust: 21世纪的编程语言新星

01

安全、高效？我全都要

编译型语言，借用与所有权检查，语言间互操作性好，安全边界清晰，开发效率高、调试成本小

02

小语法，大功能

零开销抽象，过程宏、卫生宏，软件约束与泛型约束，模块与包管理语法，async/await异步编程

03

不再惧怕裸机开发

无畏并发，#[entry]宏，RTOS与embassy，原厂支持的代码、文档和生态，开源合作便利

04

完整的系统软件生态

多平台、多目标工具链，调试与烧录软件，适配系统开发的IDE，系统软件常用库和厂家软件等

什么是RISC-V SBI？

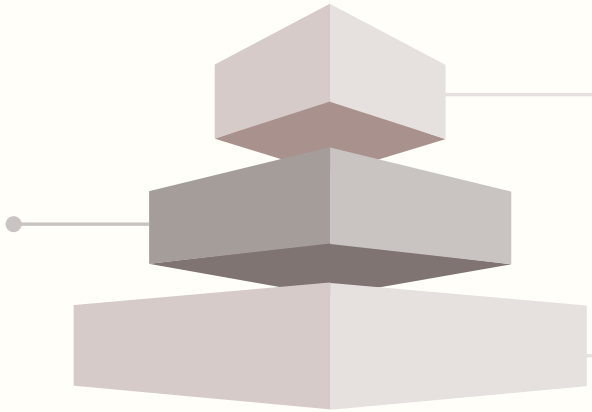
UEFI

LinuxBoot

U-Boot (S)

内核态

执行内核阶段引导程序，同时运行内核



用户态

执行功能丰富的用户程序

机器态

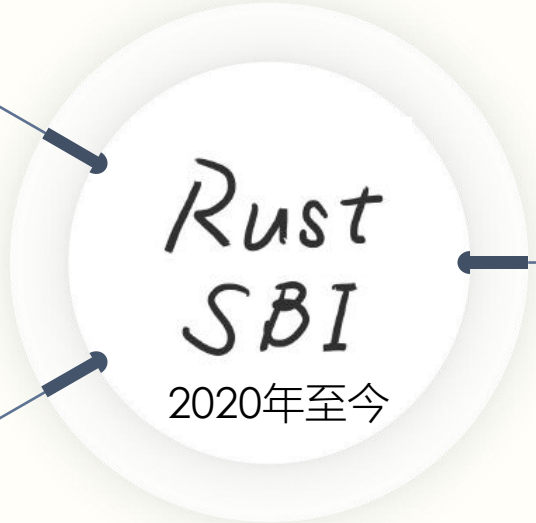
启动内核，并提供内核运行所需的基本功能

→ RISC-V SBI接口

使用Rust实现SBI接口

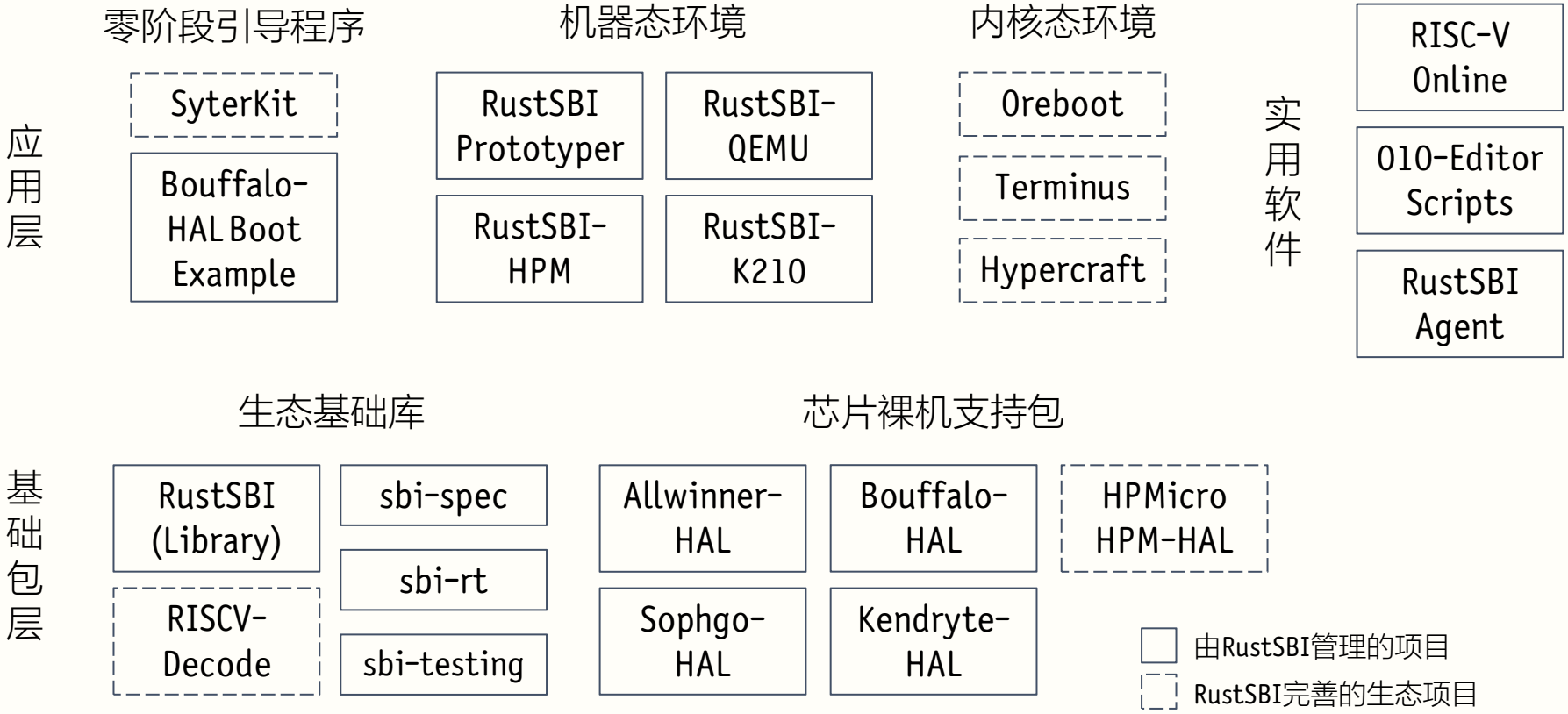
起源于鹏城实验室与rCore联合举办的开源夏令营活动，前身是MeowSBI

运用安全、高效的语言，构造属于RISC-V的安全引导解决方案

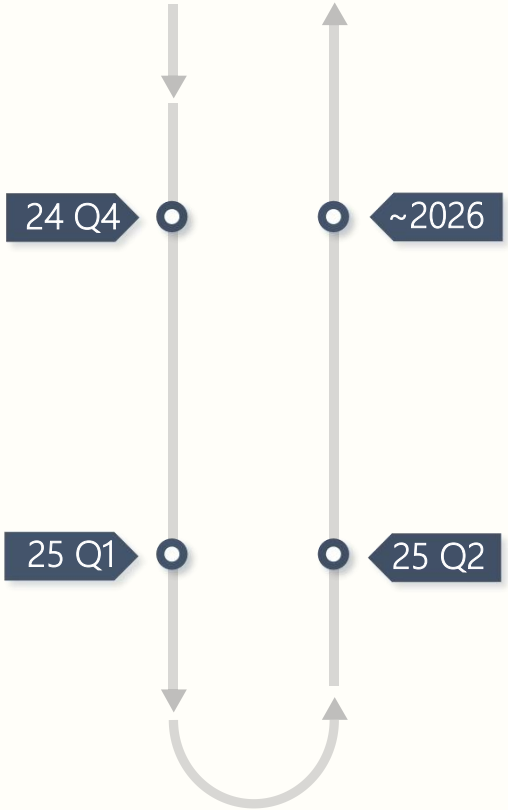


目前审核、贡献团队合计约15人，横跨华中科技大学等多个高校，已举办线下开源活动21次

今日的 RustSBI 架构概览



RustSBI原型设计与实现路径



模拟器上支持发行版系统

QEMU上使用RustSBI作为机器态固件，支持Ubuntu、Arch Linux和Fedora等发行版

获得更广阔的市场应用

与多个国内、国际芯片原厂建立合作，适配Rust裸机开发和RustSBI引导方案

支持真实硬件的统一支持包

基于SyterKit或Bouffalo-HAL Boot等前级软件，启动RustSBI并支持S态各类内核

UEFI+RustSBI综合固件方案

完成内核态引导程序部分，与机器态RustSBI结合，构成支持UEFI的引导方案；生态引导软件支持启动多款RTOS系统

贰

裸机程序、支持平台与生态

RISC-V的引导程序和固件环境与其它架构是不同的，它除了引导启动内核，还将常驻后台，不断提供内核所需的运行时功能。这对安全引导程序的实现方法给出了新的要求和挑战。

嵌入式 Rust 与引导程序的结合：三个观点

01

引导程序和嵌入式软件同属裸机应用

02

RTOS和M态引导程序环境都可被零阶段引导程序启动

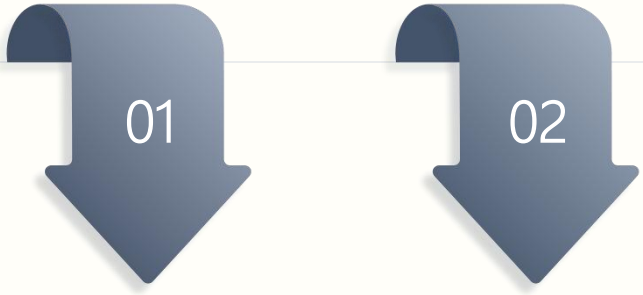
03

物联网应用和引导程序都依赖于网络与互联软件栈

引导程序、嵌入式应用和操作系统共享同一个Rust生态！

执行环境

```rust
#[entry] // This macro would initiali
fn main(p: Peripherals, _c: Clocks) {
    // Display the bootloader banner.
    show_banner();

    // Initialize the DRAM.
    let dram_size: usize = syterkit::
    println!("DRAM size: {}M 🐏", dr

    // Dump information about the sys
    clock_dump(&p.ccu);

    // Start boot command line.
    let (command_buffer: [u8; 128], h
```



同步环境

基于循环和放松的阻塞操作完成引导功能。简单、稳定

异步环境

合理调度任务，错开设备初始化，充分利用设备性能。高效、实用

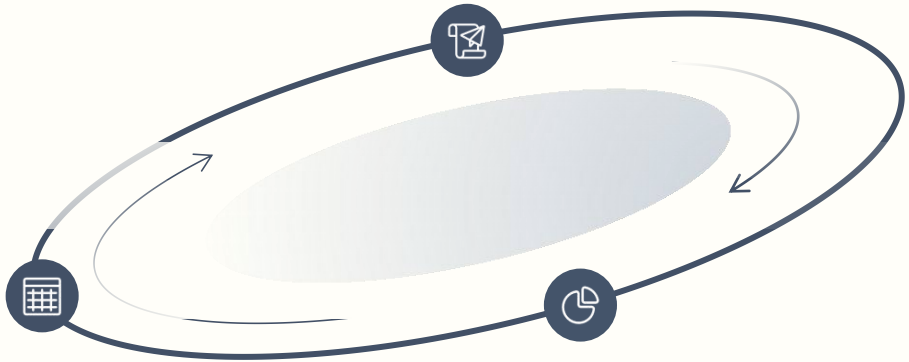
```rust
#[embassy_executor::main(entry = "hpm_hal::
async fn main(spawner: Spawner) -> ! {
    let p = hal::init(Default::default());

    defmt::info!("Board init!");

    spawner.spawn(blink(p.PE14.degrade(), 1
    spawner.spawn(blink(p.PE15.degrade(), 2
    spawner.spawn(blink(p.PE04.degrade(), 3

    defmt::info!("Tasks init!");

    loop {
        defmt::info!("tick");
```

零阶段引导程序——以SyterKit为例

全志ROM引导程序

符合ROM eGON约定，可启动

多款不同架构的全志公司芯片



底层调试功能

仅128KB SRAM下具有基础

调试功能，可调试底层驱动

启动机器态应用

可从闪存、SD卡等依据DTB

启动M态Linux等操作系统

项目链接：https://github.com/YuzukiHD/SyterKit（请star！）

延长现有硬件的生命周期

使用satp/sptbr替换、sfence模拟和页错误模拟等软件功能，基于SBI固件实现（RustSBI），实现旧特权级版本硬件运行新版本内核

硬件到硬件的兼容性

当文档开放，开源社区会激发出无限的生产力！RustSBI的发展得益于厂家对开源社区的大力支持，也为各个厂家的商业方案反哺力量

开放文档和手册

软硬件结合的指令集架构

RISC-V的CSR寄存器一部分由软件实现，借助SBI固件可提供指令集扩展模拟等功能[1]，良好设计的SBI固件可提高系统的运行效率[2]

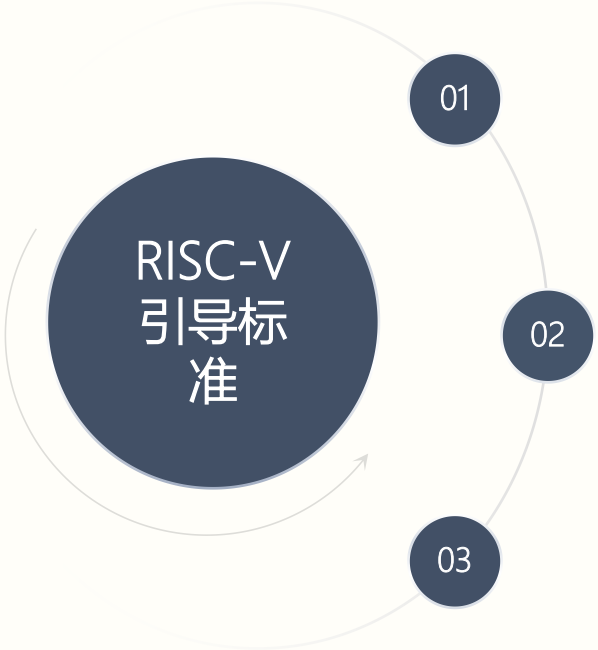
[1] 如@dramforever的H扩展指令集模拟项目
[2] 一个例子是：https://github.com/rustsbi/prototyper/pull/8

叁

启动与支持操作系统

RISC-V的引导程序和固件环境与其它架构是不同的，它除了引导启动内核，还将常驻后台，不断提供内核所需的运行时功能。这对安全引导程序的实现方法给出了新的要求和挑战。

裸SBI、LinuxBoot和UEFI



裸SBI启动

M态固件由hart_id、opaque参数直接启动S态的各类内核。这种途径的启动速度最快，可定制性最好。

LinuxBoot引导程序

是将前级Linux与SBI环境、定制rootfs共同打包到固件，由kexec启动真正Linux的方法。可复用Linux的驱动，开发量小。

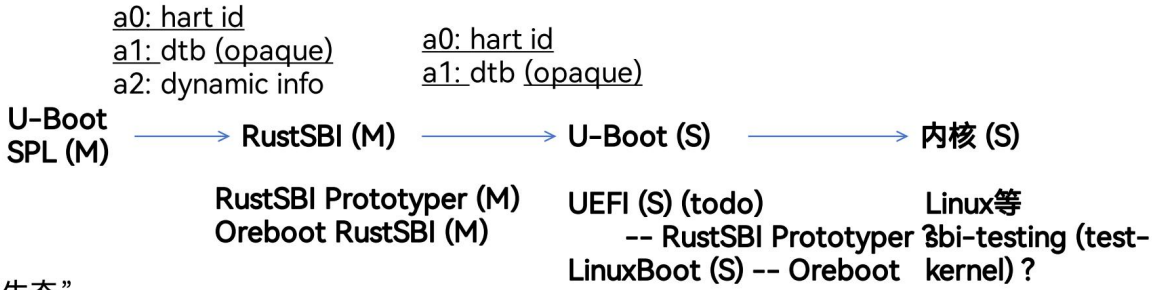
UEFI引导程序

成熟的商业化标准，可与ACPI、DT向内核提供完整的设备列表和抽象方法。它的可移植性和产品化程度在三者中最好。

如何引导发行版

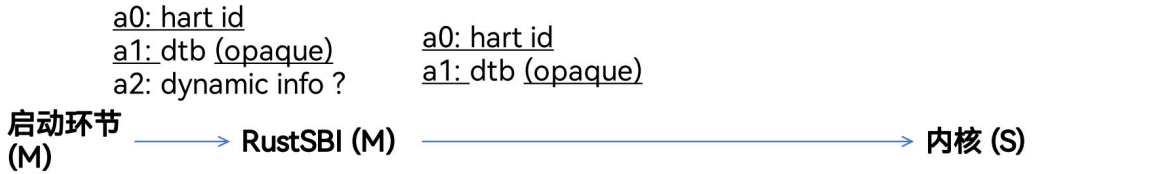
“可移植性”

镜像格式：itb (U-Boot定义)、EFI等



“闭环生态”

镜像格式：种类繁多



Cache as RAM, or SRAM

DDR or PSRAM

SBI 提升内核性能的实现方式

提升RISC-V内核性能

页刷新优化[1]

当待刷新地址段≥4倍页长度时，优先使用全局刷新指令，减少对性能的影响

栅栏队列[2]

每核收到的远程栅栏指令先进队列后执行，使得远程栅栏请求可即时返回

异步启动

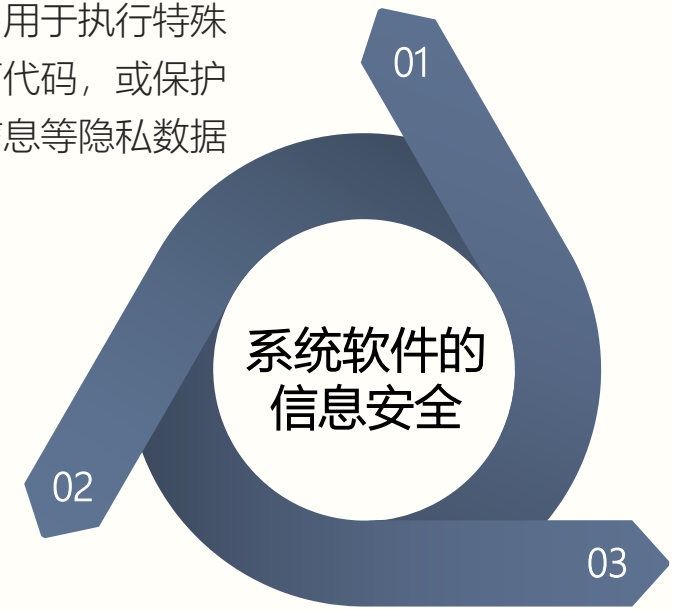
HSM扩展具有状态查询功能，可作为异步启动的状态查询原语，降低等待时间

[1] https://github.com/rustsbi/prototyper/pull/8 [2] https://github.com/rustsbi/prototyper/pull/16

操作系统的安全性需求

可信执行环境

高可靠应用中，用于执行特殊安全需求的隔离代码，或保护密钥、生物信息等隐私数据



安全引导

从硬件信任根开始逐级验签，建立系统软件本身的可信性

快速加解密

密码学应用对系统的密码学算力有独特要求，合理调度平台专有硬件完成密码学需求

肆

展望：可信环境与AI系统应用

RISC-V的引导程序和固件环境与其它架构是不同的，它除了引导启动内核，还将常驻后台，不断提供内核所需的运行时功能。这对安全引导程序的实现方法给出了新的要求和挑战。

CoVE SBI：RISC-V TEE的未来

AP-TEE

TCB与非TCB代码隔离的ABI标准

填补空缺

架构文档与安全性要求间存在空白

联动模块

特权级标准，H指令集，IOMMU等

RustSBI Agent 项目介绍

开发者的AI助手

具有RustSBI、RISC-V和芯片专业知识的对话机器人

节省开发时间

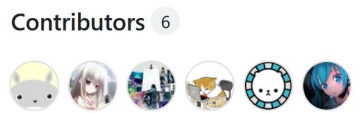
RustSBI开发者3/5的时间用于查询文档——可以省下来！

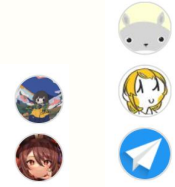
涉及的AI领域

RAG、提示词工程等是RustSBI Agent的主要技术

致谢

- 感谢华中科技大学网络空间安全学院对华科校内开源社区的大力支持
- 感谢项目的指导老师们：周威老师、慕冬亮老师和王杰老师。感谢**李明老师提供的演讲机会**
- 感谢参与项目的所有同学。参与项目的小组长有：硬件支持组朱俊星同学，发行版组邢志昂同学，AI组马铭芮同学
- 感谢无数的社区贡献者，尤其是@andelf、@tkf2001和@cyrevolt（Daniel Maslowski）
- 感谢以实际行动支持RustSBI项目的公司们，包括深圳矽速科技、南京博流、上海先楫半导体、珠海全志科技和重庆鹿仔科技公司（排名不分先后）
- 感谢PLCT实验室的大力支持





感谢观看

感谢老师们和同学们的支持！

“让我们试试RustSBI！”

洛佳 华中科技大学