基于新型编程语言设计实现操作系统内核

陈 渝　　李明

清华大学

第五届国产嵌入式操作系统技术与产业发展论坛

2023年08月12日

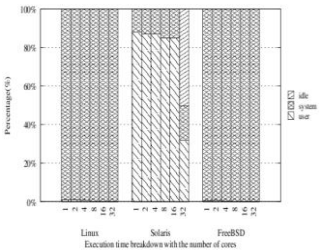
个人简介

陈渝，国防科技大学本硕博毕业，清华大学长聘副教授，CSAE-CCF-CICV操作系统联合实验室首席科学家，中国汽车工程学会基础软件分会副主任委员，中国计算机学会系统软件专委常委委员，MIT访问学者

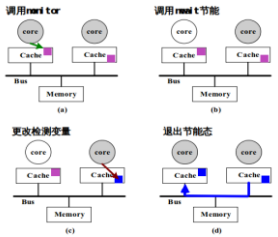
研究方向以**操作系统**为核心，涉及系统安全、性能优化、程序分析等领域。近十余年来，主要在操作系统/虚拟机性能优化，系统软件安全分析，取得了有一定国际/国内影响的研究成果。

操作系统性能优化

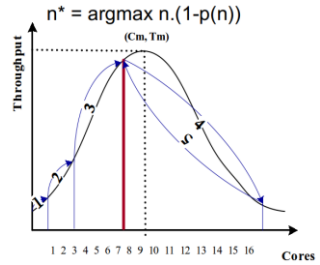
性能分析与方法



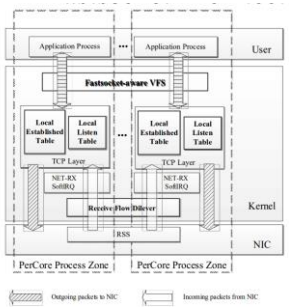
锁机制建模与可扩展优化



调度算法,内存管理

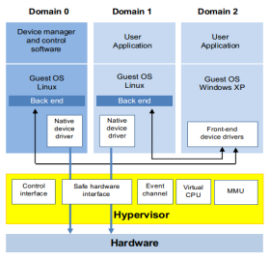


内核IO优化机制与系统

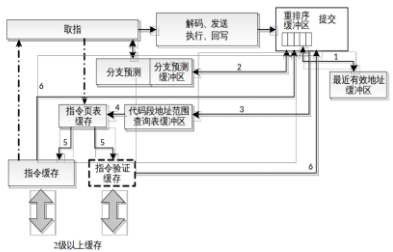


操作系统安全分析

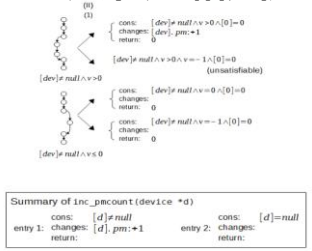
OS分区隔离安全机制



防止代码重用攻击的软硬件协同保护机制



基于不一致路径对检测的引用计数缺陷分析



目 录

研究现状

操作系统是具有战略意义的关键信息系统核心基础设施

- 操作系统是管理计算机硬件与软件资源的基础程序，是最基本、最为重要的基础性软件系统
- 被列为科技日报35项“卡脖子”核心技术之一（排第三）


科技日报

《丧失先机，没有自研操作系统的大国之痛》（2018年4月23日）

操作系统的核心问题与应用需求

- **安全问题（重点）**：操作系统安全是计算机安全的软件基石
  - Linux等基于C语言的操作系统漏洞有扩大化的趋势
- 性能问题：在多核、低延迟、新型硬件支持等方面需改进
  - 随着核数增多，内核锁竞争导致吞吐量下降
- 生态问题：需要能够兼容已有生态，利于建设应用环境

2023年度的世界黑客大赛，成功实现针对特斯拉的 TOCTOU 的竞态条件漏洞攻击

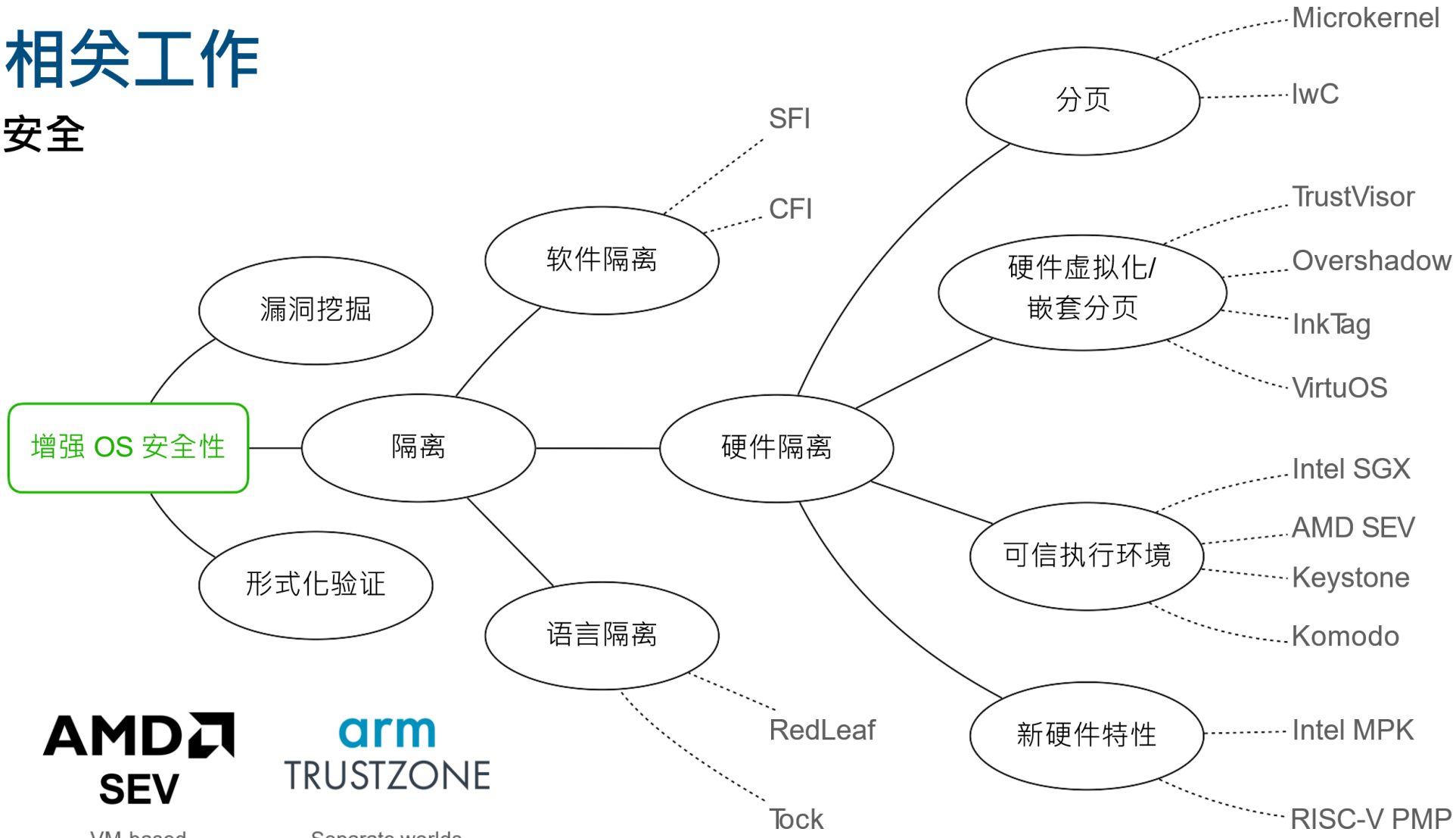
面向智能设备，高安全、高性能、应用生态好的操作系统是新生态（智能网联汽车）发展的必需

研究现状

操作系统架构　编程语言　资源管理

相关工作

安全

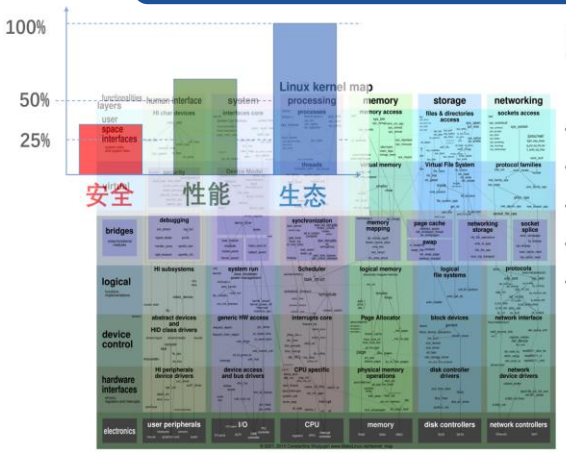


Process-based　VM-based　Separate worlds

拟解决的关键科学问题

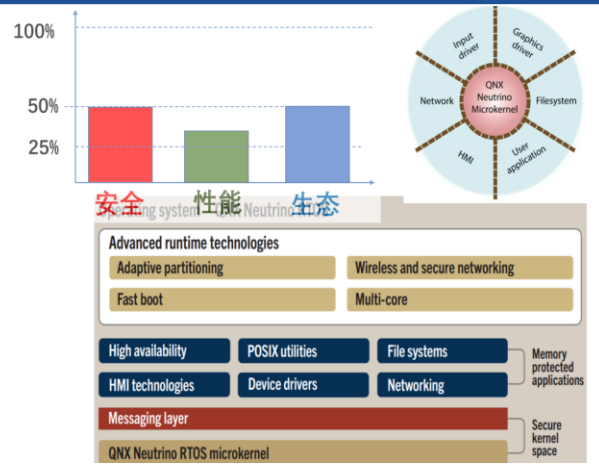
- **重点提出基于Rust的可扩展操作系统安全架构，较现有OS提供更多层次的安全保障能力，为构建自主可控IT生态奠定OS基础。**

一、如何开发与设计可灵活组合、支持持续适应优化和可持续演化的操作系统安全架构?



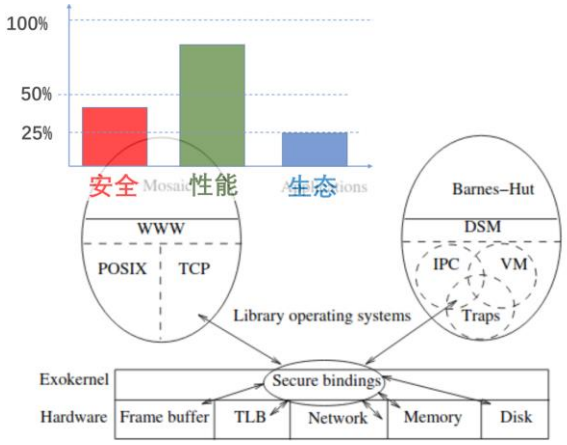
宏内核架构 monolithic-kernel

- 伪模块化：任意两个模块都可有调用关系
- 高复杂化：模块之间存在上百个的关联性
- 历史遗留：老代码难以移除，成为包袱
- 安全漏洞：C语言是多数典型漏洞的根源
- 使用场景：移动端、桌面、服务器领域



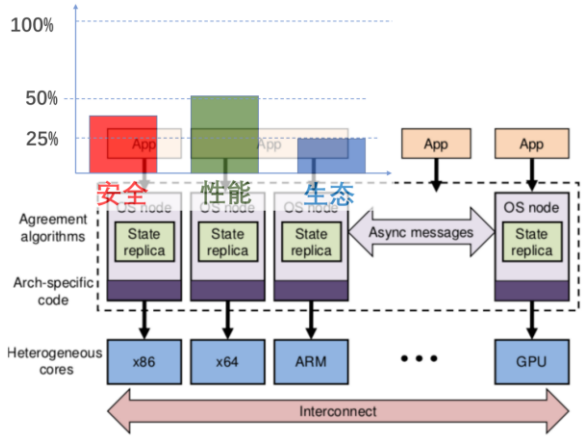
微内核架构 micro-kernel

- 应用生态：应用范围小，安全检查不足
- 高复杂化：模块之间存在上百个的关联性
- 安全漏洞：C语言是多数典型漏洞的根源
- IPC隔离：安全缺陷依然不容忽视
- 使用场景：嵌入式系统



外核架构 exo-kernel

- 开发难度：不同应用需要不同的LibOS
- 安全漏洞：C语言是多数典型漏洞的根源
- 生态劣势：缺少应用生态，安全检查不足
- 使用场景：Hypervisor/VMM



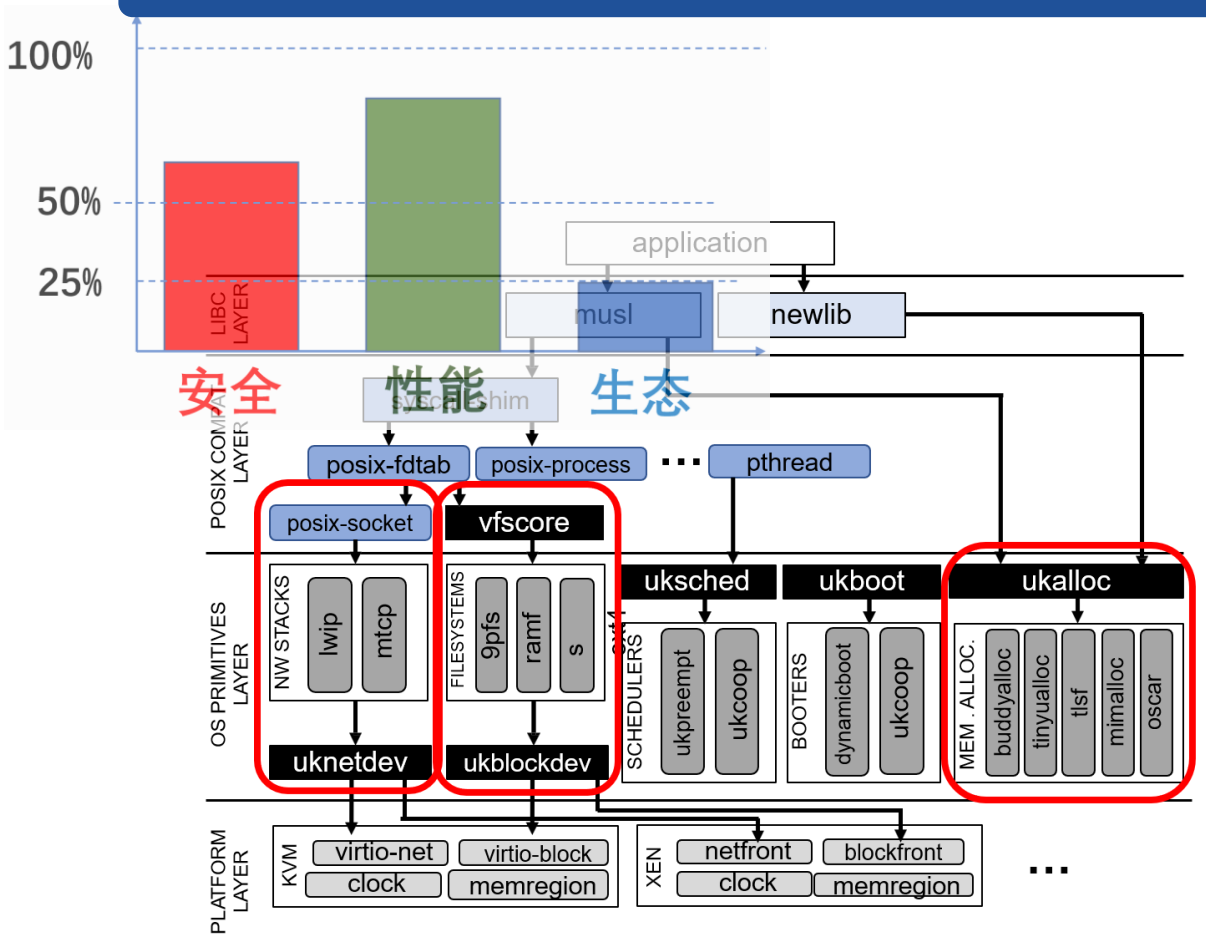
多核架构 multi-kernel

- 开发难度：分布式内核开发
- 安全漏洞：C语言是多数典型漏洞的根源
- 生态劣势：缺少应用生态，安全检查不足
- 使用场景：适合未来的异构架构

拟解决的关键科学问题

- **重点提出基于Rust的可扩展操作系统安全架构，较现有OS提供更多层次的安全保障能力，为构建自主可控IT生态奠定OS基础。**

一、如何开发与设计可灵活组合、支持持续适应优化和可持续演化的操作系统安全架构?



单核架构　uni-kernel

- 开发难度：APP与内核在同一地址空间
- 安全漏洞：C语言是多数典型漏洞的根源
- 通用需求：难以通用化
- 生态劣势：缺少应用生态
- 使用场景：高性能的特定领域场景

MIRAGE OS　RUSTY HERMIT　Unikraft

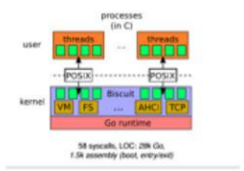
拟解决的关键科学问题

- **重点提出基于Rust的可扩展操作系统安全架构，较现有OS提供更多层次的安全保障能力，为构建自主可控IT生态奠定OS基础。**

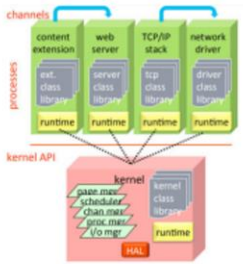
一、如何开发与设计可灵活组合、支持持续适应优化和可持续演化的操作系统安全架构?

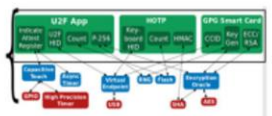
二十一世纪以来，尝试用新语言重构操作系统的探索一直在进行，其中Ocaml，C#，Go，Rust等新一代编程语言在学术界和产业界初露头角

安全是新一代系统语言试图代替C语言的重要因素



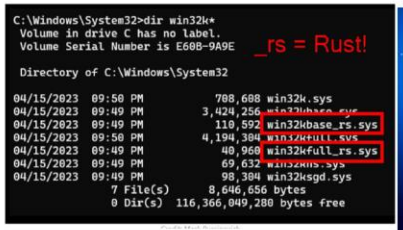
MIT用Go语言开发了Biscuit OS OSDI'2018





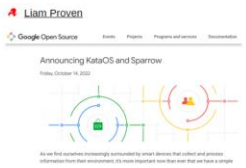
Stanford用RUST语言开发了tock OS，SOSP'2017

Replacing C++! Microsoft's Win11 kernel rewritten in Rust is here!



Google reveals another experimental operating system: KataOS

Based on Rust, on top of seL4 – a big deal in the microkernel world



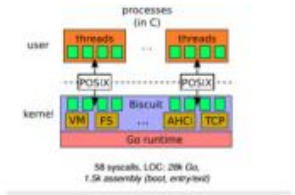
seL4 Foundation

Linus Torvalds: Rust will go into Linux 6.1

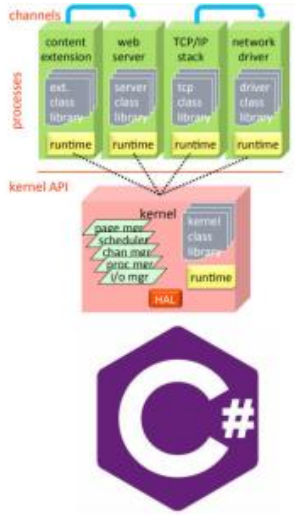
At the Kernel Maintainers Summit, the question wasn't, "Would Rust make it into Linux?" Instead, it was, "What to do about its compilers?"

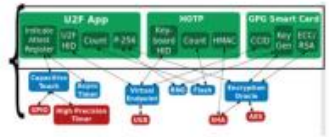
Written by Steven Vaughan-Nichols, Senior Contributing Editor on Sept. 19, 2022

目 录



MIT用Go语言开发了Biscuit OS
OSDI'2018





Stanford用RUST语言开发了
tock OS，SOSP'2017

主要研究内容及技术架构

■理论基础：

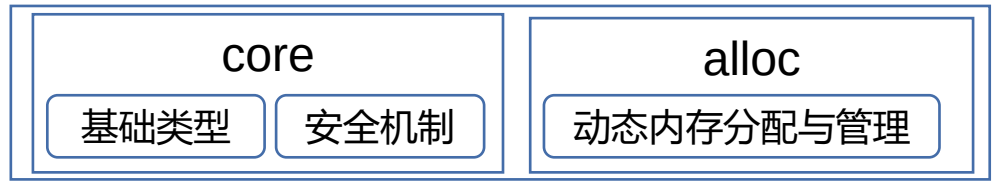
扩展Rust语言的所有权(Ownership)机制

到操作系统资源管理中，原理基于：

- **类型系统的Affine types(仿射类型)**
- **值/资源拥有者唯一/所有权转移/借用**

基于Affine Type的Ownership安全机制

Rust核心库



核心规则：

- **所有权规则**：每个值/资源在任意时刻只能由一个所有者拥有
- **借用规则**：临时地借用值而不获取所有权
- **生命周期规则**：验证引用或借用的有效范围

	Exchange	Weakening	Contraction	Use
Ordered	—	—	—	Exactly once in order
Linear	Allowed	—	—	Exactly once
Affine	Allowed	Allowed	—	At most once
Relevant	Allowed	—	Allowed	At least once
Normal	Allowed	Allowed	Allowed	Arbitrarily

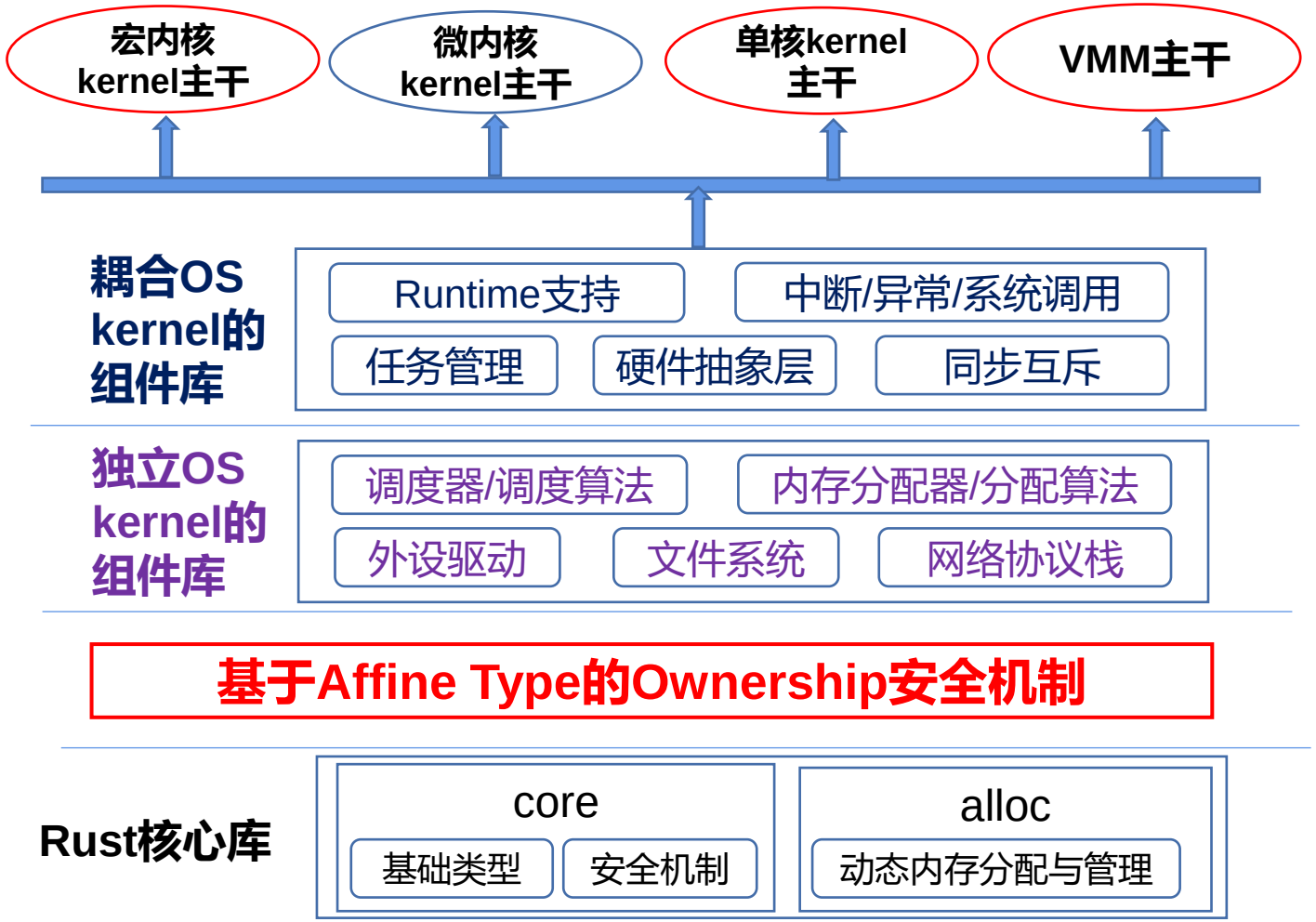
Substructural type system

主要研究内容及技术架构

■操作系统架构设计：

细化内核模块属性，形成单向依赖，独立存在的内核模块

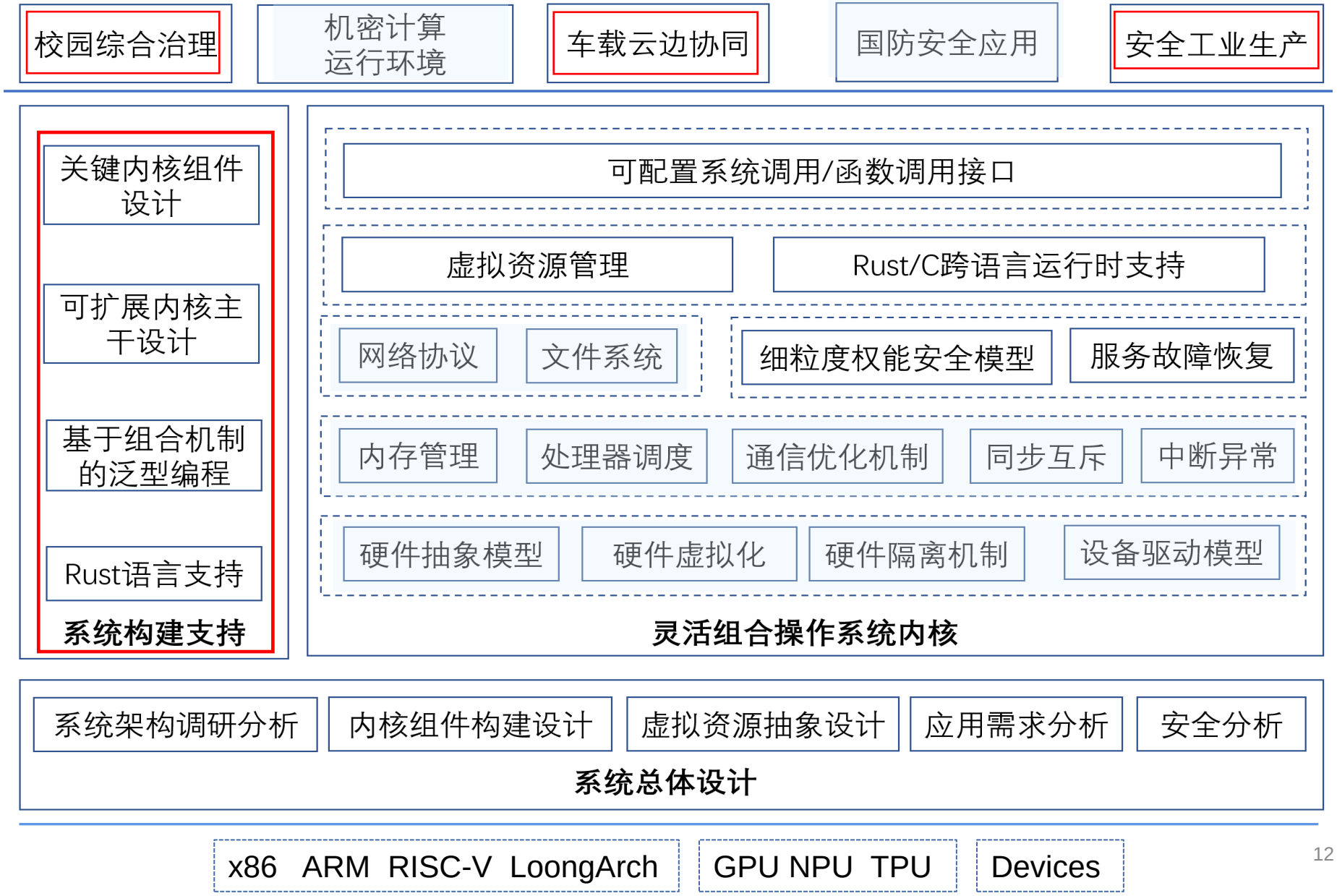
- 与OS无关的语言级核心库
- 独立OS的组件库
- 耦合OS的组件库
- 挂接组件的内核主干
- 独立OS的组件库无紧耦合依赖关系
- 基于配置和静态分析进行架构优化



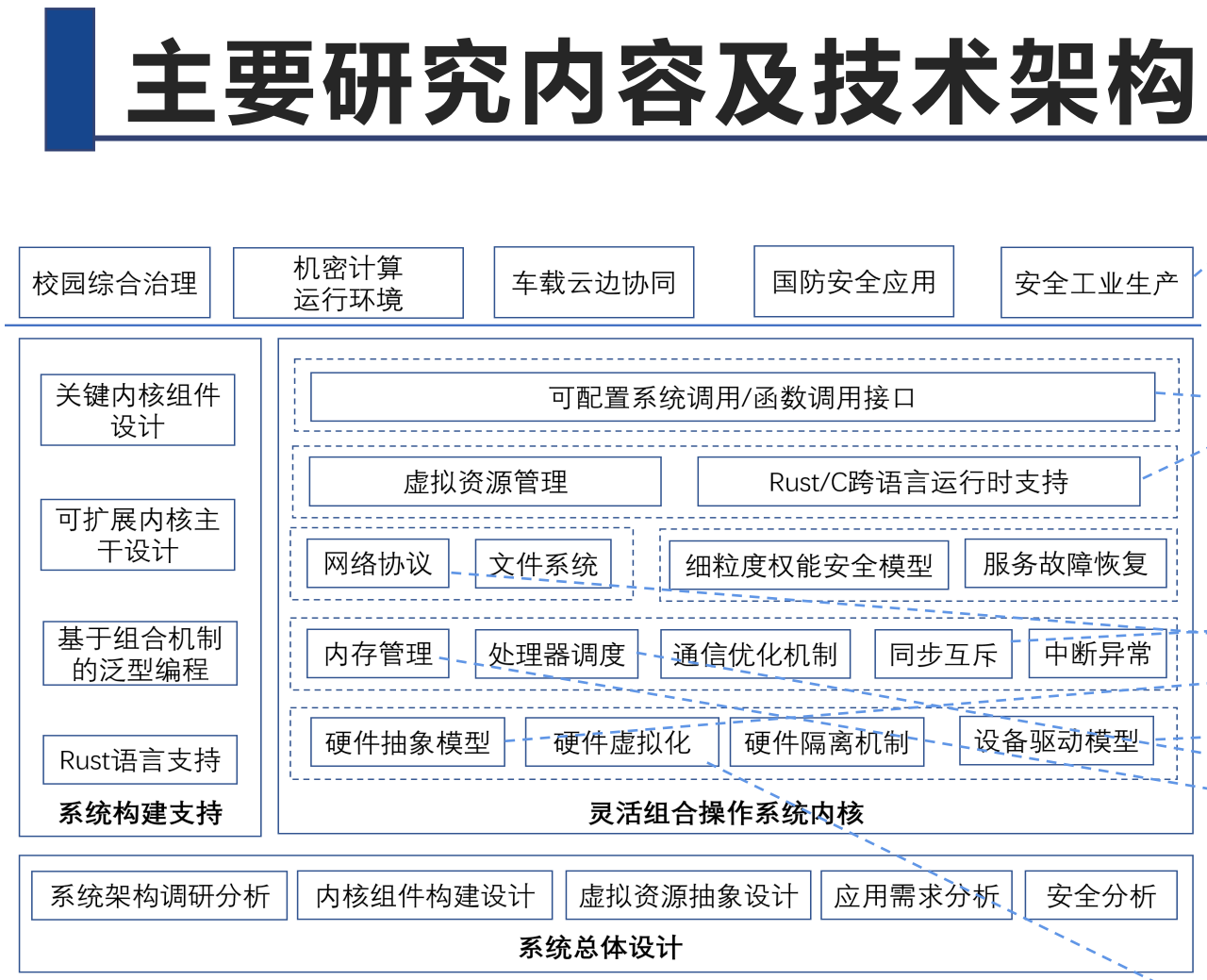
主要研究内容及技术架构

■新型架构操作系统内核实现

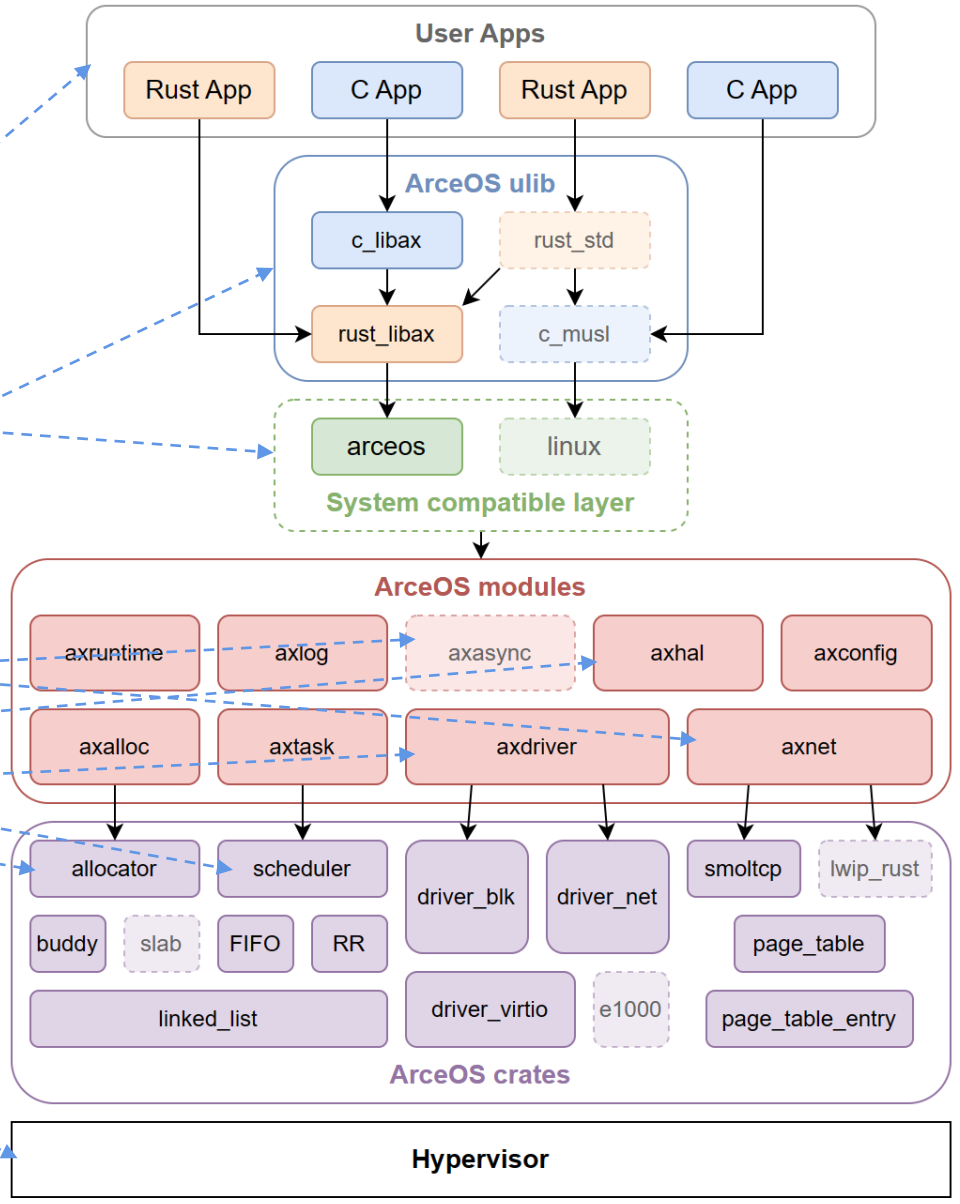
分别从关键系统组件、操作系统内核主干、内核组合机制等多个角度分别进行研究



主要研究内容及技术架构



Unikernel架构模型



Unikernel架构的ArceOS内核

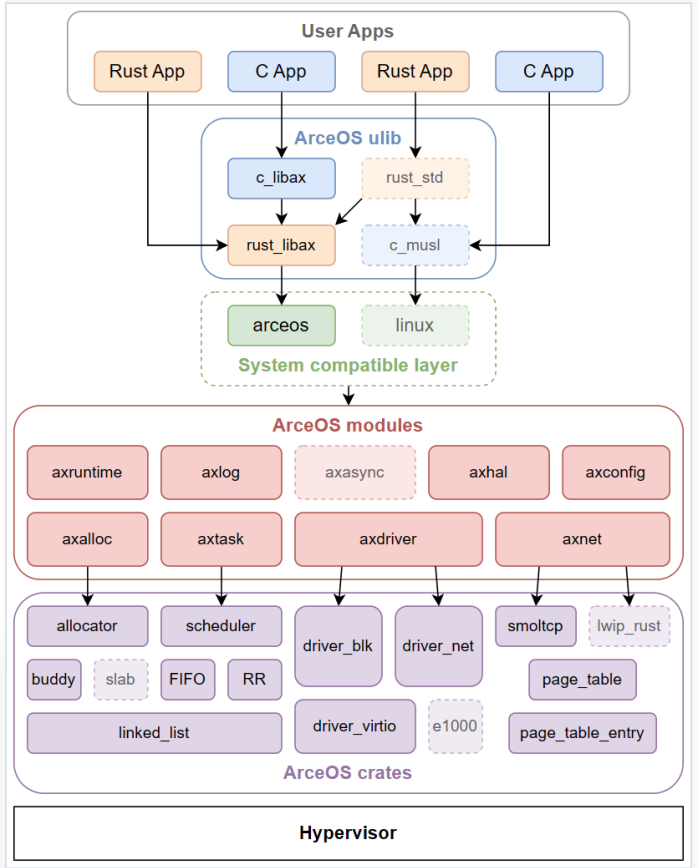
关键技术及指标

操作系统内核安全组合机制

支持最小权限的权能安全技术

- 基于组合模式和泛型模式进行功能扩展
- 特别是采用Rust语言的trait机制
- 实现操作系统内核组件在行为与数据上的分离
- 内核组件具有泛型编程的特征
- 通过Ownership机制的组合约束确保安全性

预期成果：
支持Hypervisor和kernel 形态，支持微内核、宏内核、Unikernel等操作系统架构，支持100个以上的Linux Syscall



ArceOS Modules

- axalloc: ArceOS global memory allocator.
- axconfig: Platform-specific constants and par
- axdisplay: ArceOS graphics module.
- axdriver: ArceOS device drivers.
- axfs: ArceOS filesystem module.
- axhal: ArceOS hardware abstraction layer, prc
- axlog: Macros for multi-level formatted loggi
- axnet: ArceOS network module.
- axruntime: Runtime library of ArceOS.
- axsync: ArceOS synchronization primitives.
- axtask: ArceOS task management module.

Crates

- allocator: Various allocator algorithms in a unified interface.
- arm_gic: ARM Generic Interrupt Controller (GIC) register definitions and bas
- axerrno: Error code definition used by ArceOS.
- axfs_devfs: Device filesystem used by ArceOS.
- axfs_vfs: Virtual filesystem interfaces used by ArceOS.
- axio: `std::io` -like I/O traits for `no_std` environment.
- capability: Provide basic capability-based security.
- crate_interface: Provides a way to define an interface (trait) in a crate, but c
- driver_block: Common traits and types for block storage drivers.
- driver_common: Device driver interfaces used by ArceOS.
- driver_display: Common traits and types for graphics device drivers.
- driver_net: Common traits and types for network device (NIC) drivers.
- driver_pci: Structures and functions for PCI bus operations.
- driver_virtio: Wrappers of some devices in the `virtio-drivers` crate, that i
- handler_table: A lock-free table of event handlers. crates.io v0.1.0
- kernel_guard: RAII wrappers to create a critical section with local IRQs or pr
- lazy_init: A wrapper for lazy initialized values without concurrency safety bu
- linked_list: Linked lists that supports arbitrary removal in constant time.
- memory_addr: Wrappers and helper functions for physical and virtual addr
- page_table: Generic page table structures for various hardware architecture
- page_table_entry: Page table entry definition for various hardware architect
- percpu: Define and access per-CPU data structures.
- percpu_macros: Macros to define and access a per-CPU data structure.
- ratio: The type of ratios and related operations.
- scheduler: Various scheduler algorithms in a unified interface.
- slab_allocator: Slab allocator for `no_std` systems. Uses multiple slabs with
  4096 bytes.
- spinlock: `no_std` spin lock implementation that can disable kernel local IR
- timer_list: A list of timed events that will be triggered sequentially when the
- tuple_for_each: Provides macros and methods to iterate over the fields of a

关键技术及指标

多态硬件抽象管理
驱动开发模型

- 硬件抽象层设计
- 硬件隔离支持
- 设备驱动模型
- 硬件抽象管理和开发模型
- 驱动程序的可靠性
- Aarch64/x86-64/RISCV-64/LoongArch64

预期成果：
支持国产自主可控的CPU/RISC-V/ARM，支持网络/存储/AI加速芯片多种外设。

VirtIO-drivers-rs

VirtIO guest drivers in Rust. For **no_std** environment.

Device	Supported
Block	✅
Net	✅
GPU	✅
Input	✅
Console	✅
Socket	✅

Used by 332

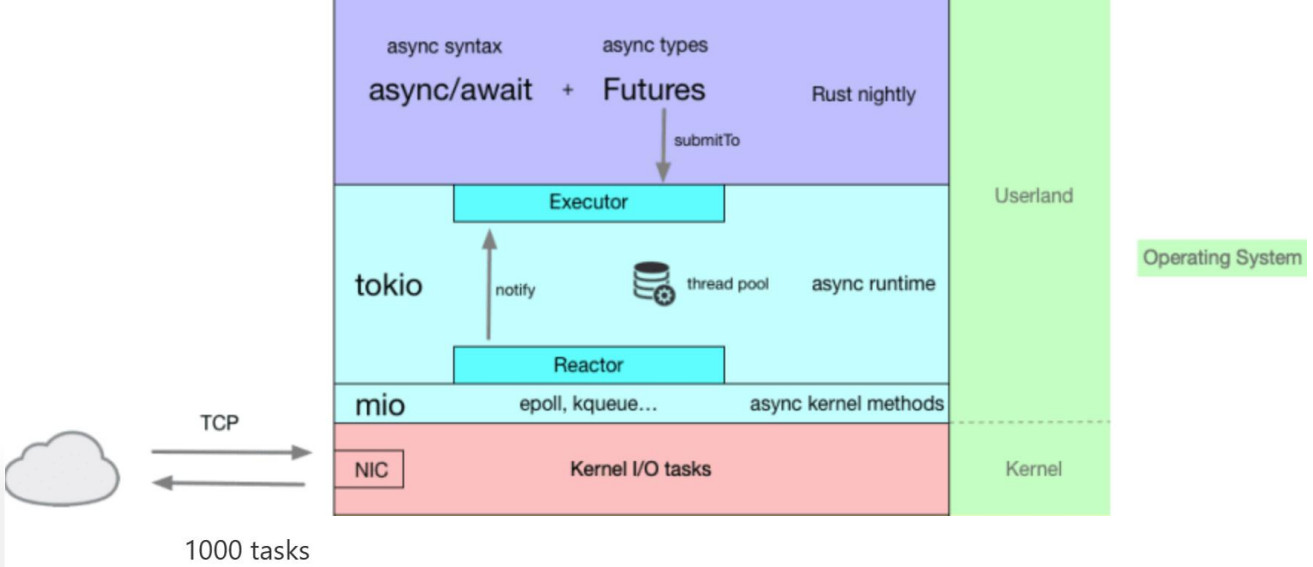
Contributors 12

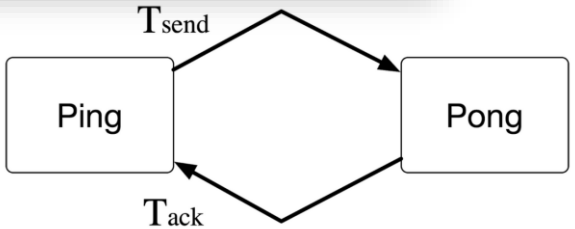
关键技术及指标

协程级异步调度机制

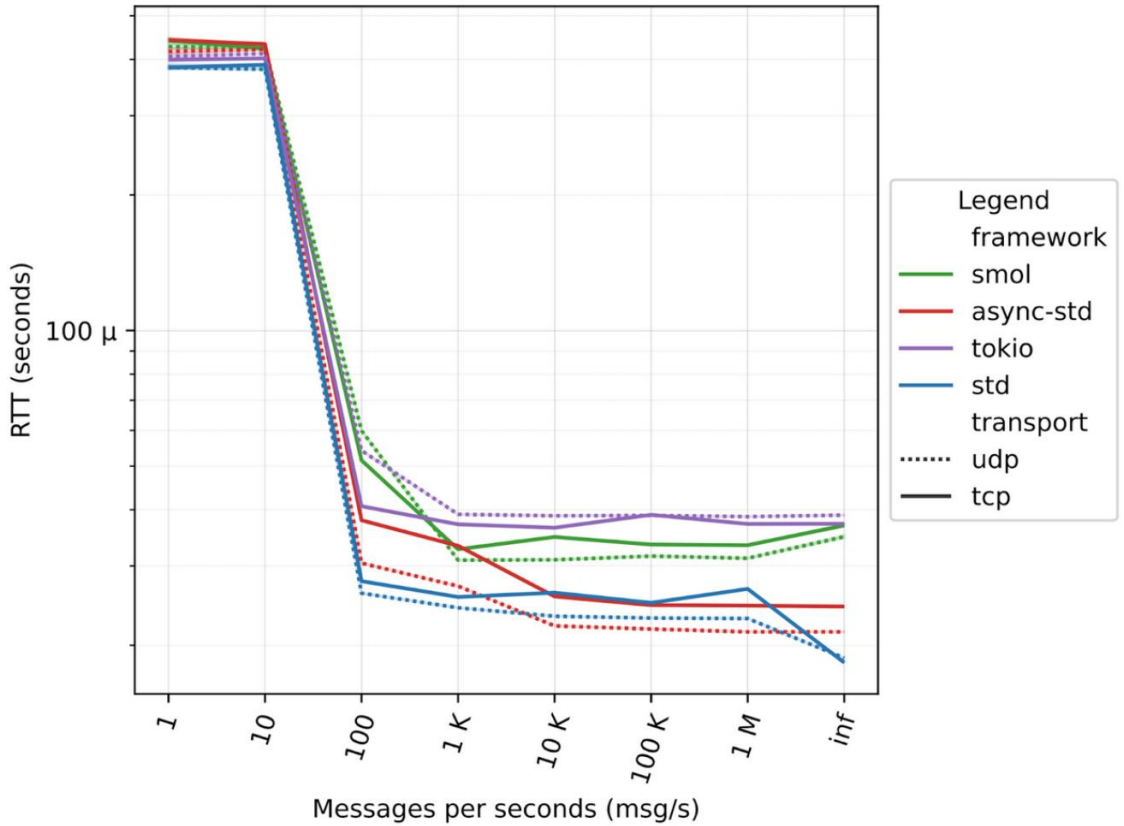
- 研究进程/线程/协程的嵌套调度机制
- 实现协程与线程之间的互协作机制
- 在内核协程设计方面，采用中断线程化设计
- 支持内核控制流在协程与线程之间进行切换

预期成果：
对redis/sqlite等典型Linux应用，比Linux Kernel在总体性能上超过50%。



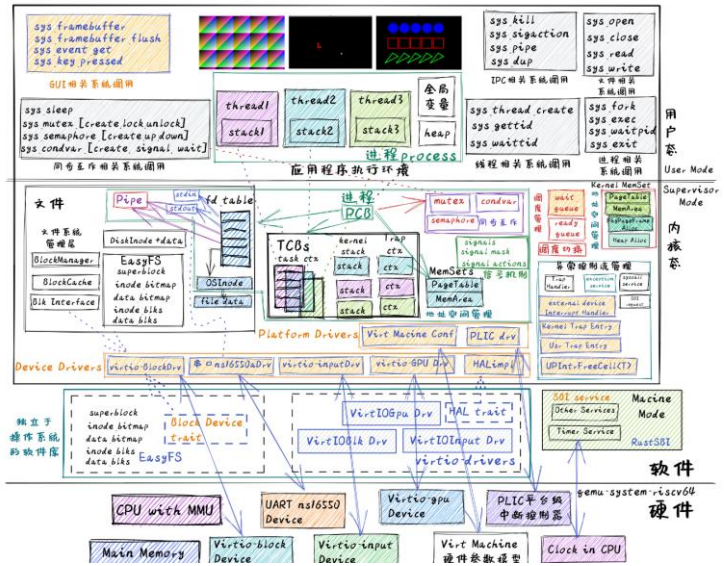


$$T_{rtt} = T_{ack} - T_{send}$$



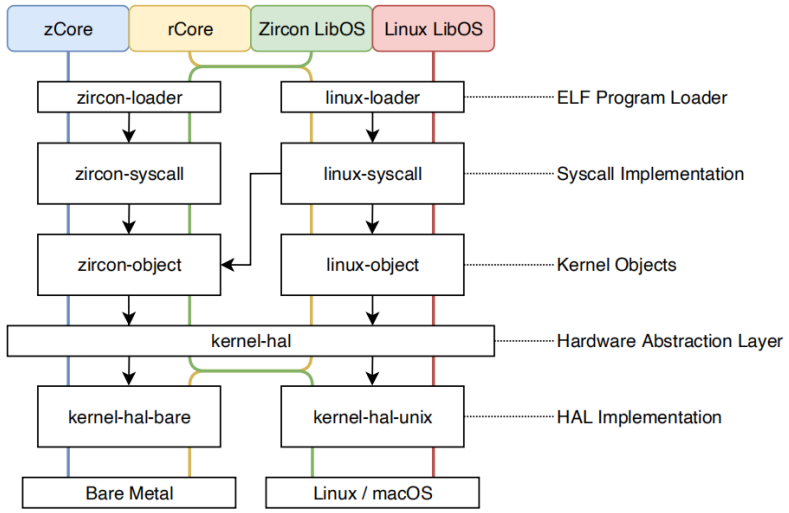
工作基础

uCore-tutorial →

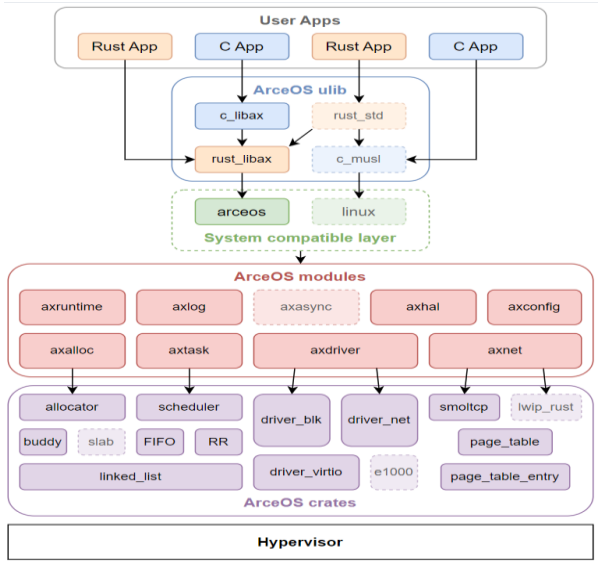


rCore-tutorial →

rCore→zCore→



unikernel → multikernel



2009 uCore-tutorial
教学操作系统

2018 rCore-tutorial
国内第一个教学
Rust-based OS

2019 rCore
国内第一个科研
Rust-basd OS

2020 Occlum LibOS
与蚂蚁金服合作，面向机密计算OS
zCore
多模态的Rust-basd OS

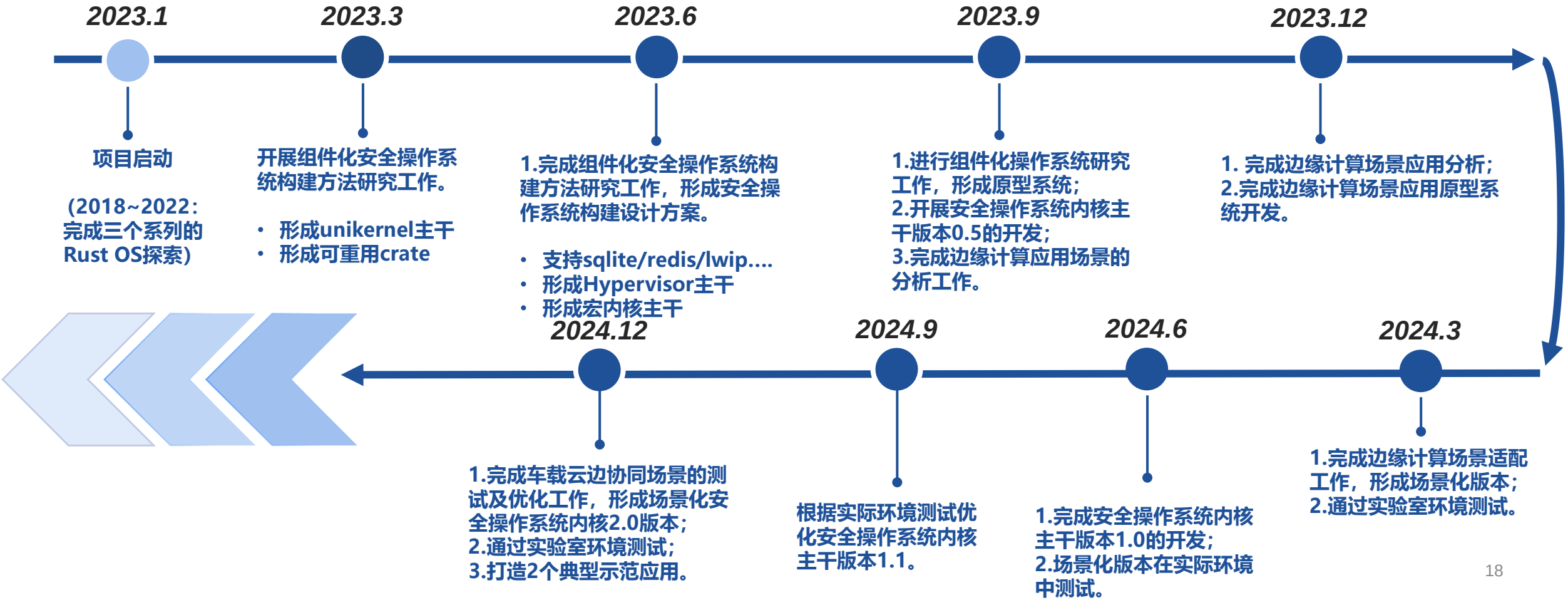
2022 HyperEnclave
与蚂蚁金服合作，机密计算VMM
RVM
面向实时系统的Hypervisor

2023 unikernel
泛在kernel
RVM-1.5
Hypervisor with linux
arceos
面向组合模式的OS框架

科研计划

清华大学　北京大学　中科院计算所　国汽智联创新中心 泉城实验室等

■ **以安全操作系统内核的研究和实现为主线，以应用场景的验证为辅助组织科研工作。**



目 录

重点应用场景：智能网联车用操作系统内核

场景描述

基于安全操作系统内核，支持一端连接智能网联汽车整车企业，确保整车应用层面自动驾驶、智能网联、信息娱乐等功能的实现；一端连接芯片企业，确保不同功能的硬件资源可以充分适配调用。

CSAE-CCF-CICV操作系统联合实验室

	2023	2024	2025 ...
近期目标	6月底基线1.0版本实车示范演示 Linux 基线1.0版本开源 Linux 6.1 内核安全加固、实时优化，基于Rust语言重写相关驱动		
中期目标	基于Rust语言的全新自动驾驶操作系统实车验证 复用基于Rust语言重写的相关驱动 复用基线1.0版本实时优化能力 全新自动驾驶操作系统、Rust语言、Unikernel架构		
远期目标		支持多核异构、多域融合汽车电子/电气架构 基于Rust语言的全新自动驾驶操作系统量产落地	

Rust for Linux　Unikernel　Multikernel　Hypervisor

1. **驱动代码移植&验证完成**
   - 从 Linux v5.10 到 v6.1
   - 总线及设备驱动40余种
2. **实车验证进行中**
   - 合众哪吒自动驾驶功能适配
3. **多单位联合开发**
   - 清华大学、电子科大
   - 黑芝麻、诚迈科技
   - 创新中心
4. **团队协同工作**
   - 建设代码GIT仓库
   - 编写开发文档
   - 技术经验分享

天目全数字实时仿真软件 SkyEye

www.digiproto.com 迪捷软件 DIGIPROTO

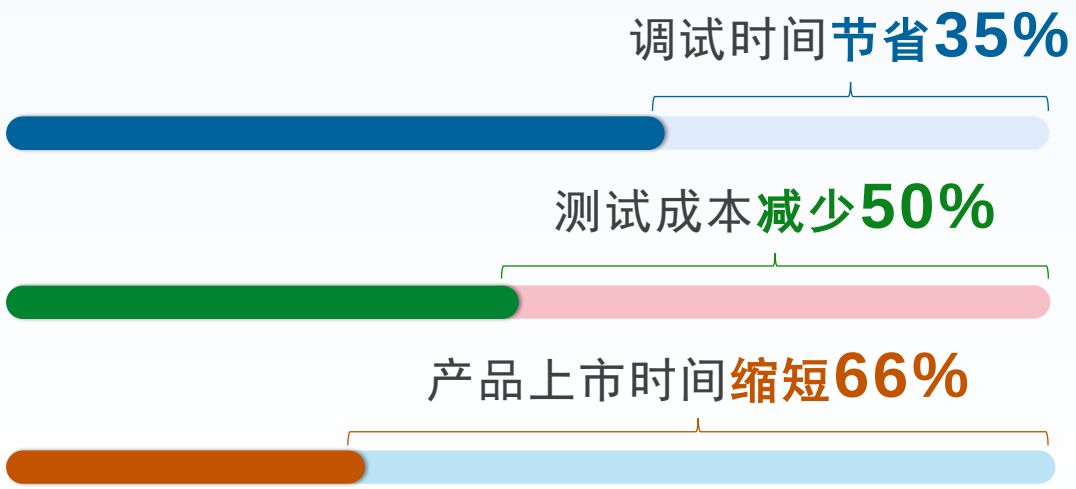
SkyEye起源

源自于清华大学**2003年陈渝老师**发起的开源软件，被国内外个人爱好者和高校**大量使用**，且有相关操作系统书籍以及高校课程。

目前由**康烁老师&迪捷软件**负责商业化开发，广泛用于飞机、导弹、卫星和汽车等安全关键领域。

SkyEye介绍

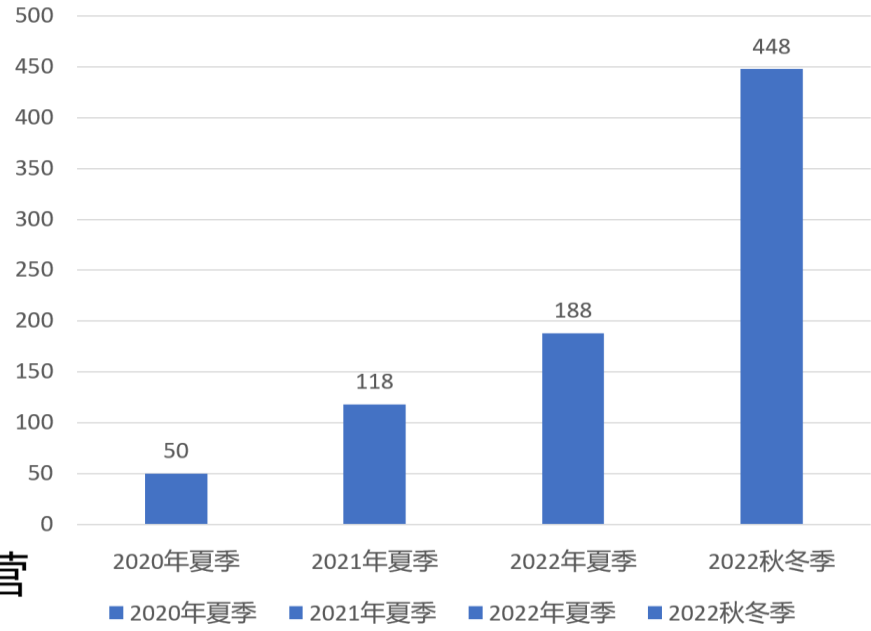
- 基于**可视化建模**的硬件行为级仿真平台，支持用户通过拖拽的方式对**硬件**进行行为级别的**仿真和建模**。
- 仿真**ARM、PowerPC、X86、DSP、Sparc、MIPS、龙芯、飞腾、国微**等主流嵌入式硬件平台。
- 适配国内自主研发的操作系统**天脉、锐华、翼辉、RT-Thread等**。
- 通过利用**基于LLVM的动态二进制翻译**技术，使虚拟处理器运行速度可以达到**2000MIPS**以上。



人才培养与团队建设

系统软件比赛 & 开源操作系统训练营

- 2020年
  - 2020.7.1 ~ 8.31 夏季训练营
- 2021年 CSCC 全国大学生计算机系统能力大赛
  - 2021.7.1 ~ 7.31 夏季训练营
- 2022年 第五届中国软件开源创新大赛 CSCC 全国大学生计算机系统能力大赛
  - 2022.7.3 ~ 8.31 夏季训练营
  - 2022.10.16 ~ 2023.2.1 秋冬季训练营



rCoreOS开源社区

探索基于Rust语言开发各种架构/形态的操作系统内核框架和可重用的内核组件，帮助应用和内核开发者快速搭建满足特定应用需求的高性能高安全操作系统内核。

- 想法来源：基于泛型独立组件构建各种领域OS
- ArceOS: 组件化操作系统的初步探索

开源操作系统年度技术会议 OS2ATC

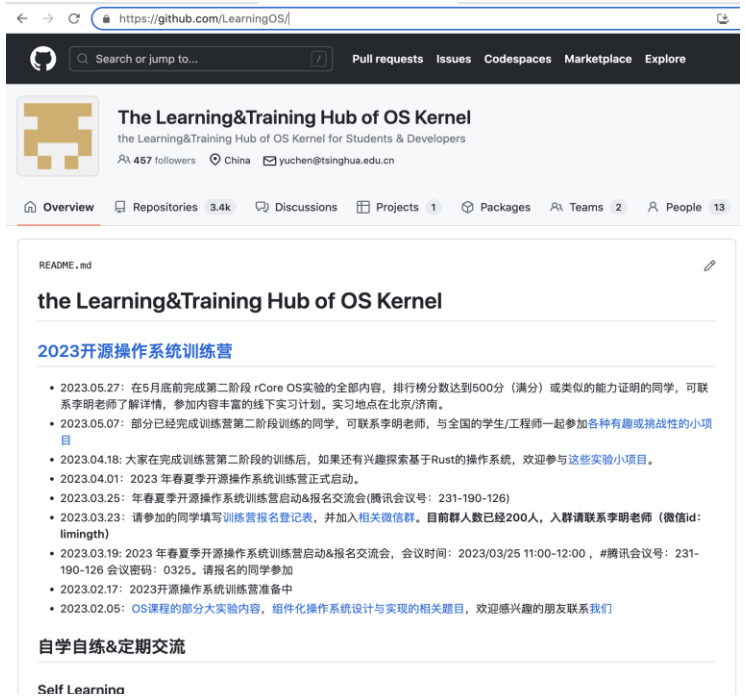
中国Linux内核开发者大会

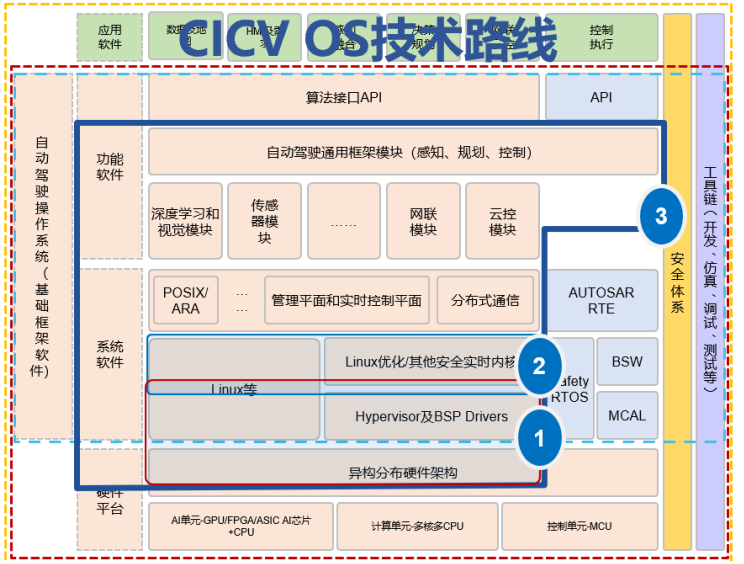
人才培养与团队建设

- **开源OS训练营：** https://github.com/LearningOS
- **开源OS科研项目：** https://github.com/rcore-os
- **清华OS课件：** https://learningos.github.io/os-lectures
- **全国大学生OS比赛:** https://os.educg.net

产	国汽智联，合众汽车，黑芝麻智能，蚂蚁，地平线等
学	清华大学，北京大学，中科院计算所，北航等
研	鹏城实验室，启元实验室，中关村实验室，泉城实验室，车用泛在操作系统联合实验室等
用	智能网联汽车操作系统，工业安全操作系统，机器人操作系统，物联网AIoT操作系统等

- **自动驾驶OS开发训练营：** https://github.com/cicvedu





谢谢！
请专家批评指正