

1. CS61 做 WeensyOS

OS161 做 chickadee 基于 WeensyOS

可以在树莓派上跑

教学型 OS

2016年版 起了操作系统设计和层次
比较简单

18年之前叫 OS161, 后面叫 chickadee

有一个小框架, 完善 但库没有做完全 不会管 C

2. FreeRTOS

嵌入式操作系统

必须
任务调度器, 提供可预测的执行模式
行为是固定的

目前: 用户提供调度方式

任务优先级优化

调度算法

deadline first

调用

先进的调度方式
考虑方向

除了算法

还有什么 架构

$U < 1$

Developer Docs

有所做的优化 (rust 改型)

核心还是 C, 只改了封装

(FFI) 外部函数接口

FreeRTOS Kernel Developer Docs

- Tasks
- Queues, Mutexes, Semaphores...
- Direct To Task Notifications
- Stream & Message Buffers
- Software Timers
- Event Groups (or "Flags")
- Source Code Organization
- FreeRTOSConfig.h
- Static vs Dynamic Memory
- Heap Memory Management
- Stack Overflow Protection
- Creating a New Project
- Co-Routines (deprecated)

stm32 是一个单片机

学习 net5

3. se14

微内核 几千行, 几百行

进程间通信 IPC

capability

10 kernel objects

他们做的

1. rust 全部改型

2. 更加模块化

把采样的全部
用 rust 改写

⊗ 传统进程间通信

⊗ 用 rust 改写 sel4
把 sel4 移植到树莓派

4. 容器化 ~~传统容器~~ docker, kn... 不太行
docker 是容器化的一种实现, 一种软件, 是开源的

把环境打包,

必须
运行一个软件

用 rust 改写 ~~qvisor~~ 没人用

容器必须与原有环境有 - 定隔离性

Hyper 是一种虚拟化技术
硬件上的虚拟化

saas

要重新调研

KATA container

比较新型
容器 1 为了更好隔离性

它已经用 rust 改写了, 但可以再调研调研
看 issue

⊗ 对 agent 改动
↓
一个协议
改善 I/O

看 rust 中的关键组件

⊗ 问题: 开发者愿不愿意

Switch:

{ Hyper
KATA container

{ freeRTOS
sel 4

z Core r Core

~~ix~~