

1. stream buffer 中断. 传递信息

缓冲区  
队列 → 更小更快, 任务阻塞  
可用于

task notification

一个对象 信号间通信 → 信号量, 互斥量 二进制量

5个字节, 不需要中间的中转 → 32位的值和1个值

有写过移植的接口  
到板级

准备工程模板  
配置文件

2.

sel4 depends

介绍 capability

init 程序 分配  
进程 sel4 cap...

boot 程序

boot loader 有个进程

csiod → 装一个 capability, 要什么

cnode → 一个 csiod 数组

cspace → 给一个线程, 有很多 cnodes,  
↓  
树状

bits 定义有多少 capability

偏移量去找它

Linux 和 sel4 区别

相当于  
可加载

set uid

soacl 有  
capabilities 一种

3.1° untype memory

sel4 没有很充足虚拟内存

删一下文件

到底有几种 capability

tcb 线程控制块

untype capability 都是指向物理内存  
re-type

什么是 capability

device 决定可不可以被内核插入

2° (tcb) 用到  $r \rightarrow r$

调度算法

3°

buffer 传递数据和 capability

调用比较快的 只能传递数据

call  
resume

3. uc/os 2 小型嵌入式系统 嵌入式系统 C语言汇编

从单核到多核, 嵌入式处理系统

为什么要级联多核?

无循环代码

通过提高主频提高性能, 达到瓶颈

再构多核 smp 对称多处理

1° 同步原语的扩展方式, 每个核上都有一个相同的操作数  
扩展多核不用多操作

2° UNIX, 分布式RTOS, 每个核都是相对独立的

amp 和 smp 是 bmp

3° 模仿 linux 设计多核系统时操作数

1° 多个处理器核的 <sup>启动</sup> 顺序问题 { 先启动核 bp

2° 多核的同步和互斥问题 互锁锁 解决资源竞争

3° 任务调度算法设计

软件结构  
与处理器无关的代码  
基于处理器的代码

同级的多线程 同级的时间片轮转

时间管理模块, 内存管理模块,

多核调度算法      静态分配策略  
动态分配策略

局部就绪队列

全局就绪队列 — 更容易实现负载均衡

对称性

时间复杂度低任务 O(1) 级别

抢占式 — 任务切换      保证切换次数最少

4<sup>th</sup> 内核以轮转工作

处理和处理器有关的参数 SMP

可用 SMP  
比较简单

多核子核

★ 和老平衡一下

1+1  
运维

Capability

+

FreeRTOS

+

多核

可行性验证

可行性报告:

capability 可行性验证怎么验证

三模冗余

FreeRTOS

代码 - 一定要重了

跨核通信

se14代码怎么测试

Rust里

性能指标

VPI 接口

调研:

1. 怎么验证 capability 的安全性

在 se14 里面的 完整性说明  
se14 和 FreeRTOS 相似性

2. FreeRTOS 为什么没有通过安全性

↓  
证明过程

3. FreeRTOS 已经可以在多核上跑

实时操作系统, 嵌入式操作系统  
放在

调研

*Thank you*