kprobes in rCore
展示报告

彭淳毅

总体进展

- Kernel
  - 实现了内核间的中断处理
- Kprobes
  - 在rCore上实现了一个简易的kprobes处理函数

Kernel部分

Kernel部分

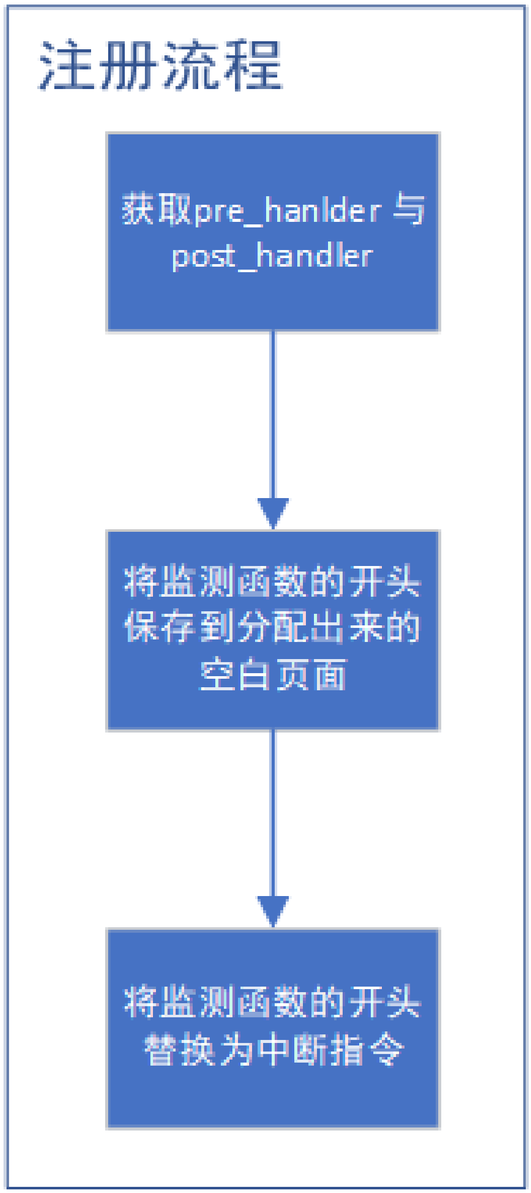
- 内核间中断处理的支持

Kprobes部分

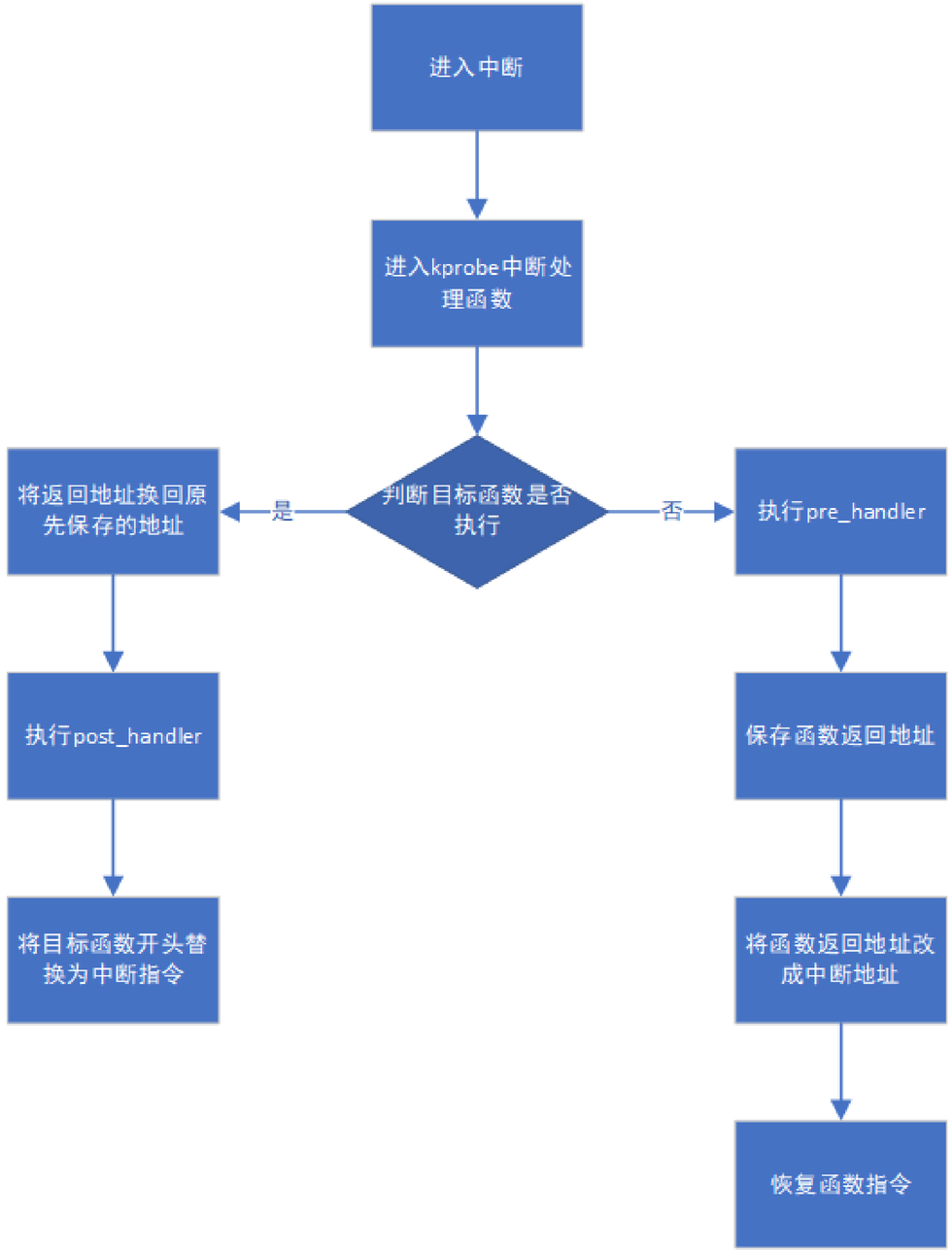
数据结构

- pre_handler : fn(),
- post_handler : fn(),
- insn_back : usize,
- kprobe_status : KprobesStatus,

注册阶段


注册流程
获取pre_hanlder 与 post_handler
将监测函数的开头保存到分配出来的空白页面
将监测函数的开头替换为中断指令

中断处理


进入中断
进入kprobe中断处理函数
判断目标函数是否执行
是
否
将返回地址换回原先保存的地址
执行post_handler
将目标函数开头替换为中断指令
执行pre_handler
保存函数返回地址
将函数返回地址改成中断地址
恢复函数指令

目前的问题

问题

- 局限性
  - 返回到原指令地址的函数
    - 函数执行过程中修改返回地址的 ×
    - 发散函数 ×
- 开销可能会很大
  - 涉及两次内存写操作

ToDo

ToDo

- ELF
  - 获取内核中的符号表信息
  - 通过函数名获取函数地址
- 减小开销
  - 重新设计中断处理的过程
- SBI移植
  - 在可以通过函数名获取地址后，尝试将功能移植进入RustSBI中

谢谢！

2021.11.04