Bridging Theory and Practice

Tips on Operating System Project

Presented by Alex Chi, Reviewed by Bugen Zhao

目录

- 如何完成这个 Project
- 常用工具介绍
- Linux 内核编程入门

如何完成 Project 2

简单三步

- 加一个 syscall。
- 根据内存占用杀进程。
- 确定杀进程的时机。

Project 1 和 2 的最大区别

- 在 Project 2 中，需要直接修改内核代码。

如何添加一个 syscall

- 在 Linux 内核中直接添加新的 syscall。
  http://blogsmayan.blogspot.com/p/adding-simple-system-call.html
- 在 Linux 启动过程中改写 syscall 数组。(没试过，应该可行)
- 通过内核模块改写 syscall。(同 Project 1)

如何杀进程

- oom_killer
  https://elixir.bootlin.com/linux/v3.4.113/source/mm/oom_kill.c#L716
- Android low-memory killer
  https://android.googlesource.com/kernel/arm64/+/android-9.0.0_r0.32/drivers/staging/android/lowmemorykiller.c

什么时候杀进程

- alloc_pages 加 hook
  - 扫描所有进程。(get_rss, handout 提供的方法)
  - 增量统计。(cgroup)
- 定时任务
  - 内核态：timer
  - 用户态：daemon + syscall

常用工具介绍

通过 git 跟踪修改的文件

Refer to “track-history-with-git.md”

Makefile

Refer to “os-project-makefile” folder

- make emulator 编译 kernel + 启动模拟器
- make test 自动编译 + push + 测试

Linux / C 内核编程入门

定义全局变量

Refer to “global-var” folder

初始化结构体

How and Where? Refer to “initialize-struct” folder

- 定义时初始化
- 在第一次使用时初始化
- 挂载 kernel module 时顺便初始化