Linux E1000 网卡驱动在 Ucore+ 上的移植

2015 年操作系统课课程设计

计 24 田博

清华大学

2015 年 6 月 18 日

完成功能概要

- 参考 JOS lab 实现 E1000 驱动
- LWIP 支持（可以启动 HTTP 服务器访问根目录文件）
- 利用 Header-gen 移植 Linux E1000 驱动编译通过
- E1000 发包
- E1000 中断收包

展示

前期工作：自制 E1000 驱动和 PCI 支持

主要参考：

- E1000 驱动（简单）：`https://github.com/bcalmeida/JOS`
- LWIP 移植（复杂）：

`https://code.google.com/p/os-xv6-network/`

LWIP 移植

- 不能照搬 JOS!
- 两部分
  - semaphore 和 message box
  - ethernet interface
- LWIP 初始化：tcpip_init 和 netif_add
- 收包：内核线程不断查看 E1000 的状态寄存器
- ucore+ 内核线程调度问题

Linux E1000 驱动移植

本质：通过 PCI 获取的 MMIO 地址，驱动程序对 E1000 进行监控、控制

主要需要实现以下种类的接口函数：

- PCI 的支持
- net_device 的创建、配置、操作
- DMA 地址的分配、映射
- ioremap 为物理地址分配虚拟地址
- NAPI 相关函数处理收包
- skbuff 的创建、操作
- 中断注册

PCI 支持

之前实现的 PCI 代码全部废弃，参考 drivers/pci 重新实现

- pci bar 的读取，参考 access.c
- pci_dev 结构体的填充
  - 关键函数：pci_setup_device 函数
  - 只依赖一些基本的 bar 访问函数

困难之处

- PCI 在 Linux 中十分复杂，例如 pci fixup
- header-gen 无法分析出响应代码
- 读《Linux 内核源代码情景分析》中 PCI 的部分
- 抽丝剥茧，定位到 pci_setup_device 这个函数

其它

- net_device 的创建主要是移植 alloc_netdev_mqs 函数
- DMA 的实现：简单
- ioremap 的实现是通过分配 KERN_TOP 以上没有使用的虚拟地址空间
- 收包会调用 NAPI 相关的函数，这一部分整体可以参考 Linux 的实现
- skbuff：收包
- 中断处理：比较粗糙，直接 hard code 了中断号，可以更细致

Header-gen 的思考

- 作用:
  - 提取最小化的头文件
  - 生成函数原型
- 不足:
  - 最小化的头文件?
  - 缺乏对函数输入的分析