

# 有线网卡驱动的可加载内核模块改造

李智康

# 实验目标描述

- 重现已有的可加载模块相关工作；
- 代码合并；
- 可加载模块的接口支持规范化：支持一个实际可用的驱动（有线网卡）；
- 可加载模块的自动升级



# 已有相关工作介绍

- 2015/ucore4edison 已经完成在内核 e1000 driver 移植
- 2015/KernelModule 实现了一个简单的虚拟字符设备驱动的可加载
- 2014/RaspberryPi 有一个简单的中断处理框架
- 2015/ucore4edison 提供了针对 pci 设备的 I/O ioremap 函数及 pci 支持

# 实现方案

- 参考 ldd3(linux device driver) ，主要思想就是围绕 e1000 设备运行，通过最少限度的对 linux 必要头函数的实现，同时在 ucore 中添加对应的支持，并通过在中间添加一层函数来完成 linux 函数和 ucore 函数之间的转换
- e1000 是一个 pci 设备，因此涉及到的主要内容有 lkm ， pci ，中断

# 主要代码修改描述

- loadable kernel module 的支持
- 在 linux header 和 ucore header 函数命名 conflict 如何解决？
- 参考 2015/KernelModule

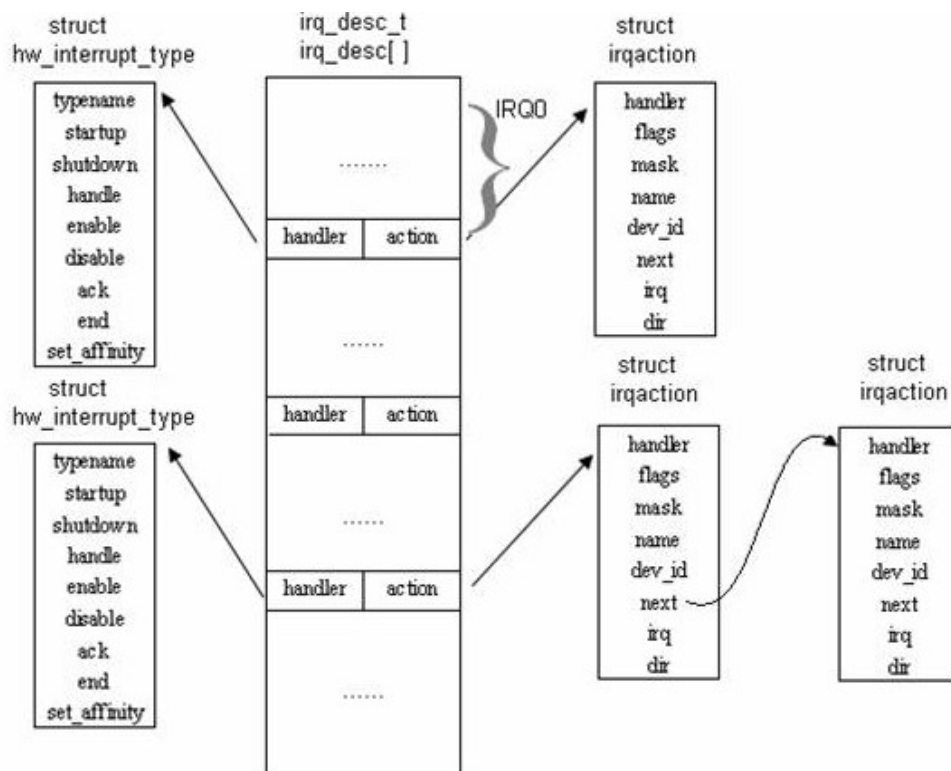
比如 linux 和 ucore 中都有 memset 这个函数，现在形成的调用关系为 memset(linux)-->ucore\_memset(ucoreFunctions)-->memset(ucore) 同时在 kmodule/Makefile 中添加 -IlinuxAdapter/include，通过这两种方式，实现了 ucore func 和 linux func 的隔离。

```
- LinuxAdapter
|
| - ucoreFunctions
| (只能引用ucore的header,
| 设计ucore_*函数,
| 内部调用ucore中的函数)
|
| - linuxFunctions
| (只能引用linux的header,
| 一些linux常用函数的实现,
| 调用uCoreFunctions中的函数)
|
| - include
| (一个通过header-gen生成的linux header组)
|
- modules
|
| - mod-A
|
|   | - A.c
|   | - A.dummy.c
|   | (linux针对A特有函数的实现
|   | 作为module的一部分 )
|   |
|   | - mod-A.dep
|   | (各模块依赖关系)
|
|
|- ...
```



# 主要代码修改描述

- 动态中断添加删除
- 导入 2014/Raspberryl



# 主要代码修改描述

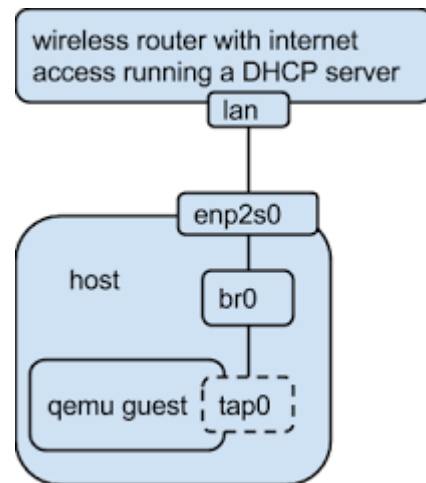
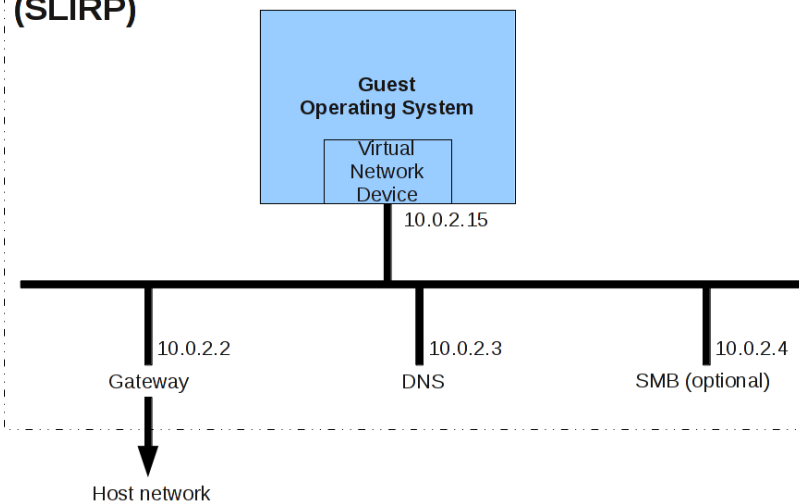
- e1000 dde 完善
- 目前是 pci 搜索到 e1000 物理设备后，在内核中新建一个 `pci\_func e1000\_func` 去存放 e1000 的相应信息
- 使用 `e1000\_netdev\_ops` 来提供 e1000 函数注册到内核使用的接口。
- 基本函数 register, unregister, open ,close ....



# 测试

在 Ubuntu14.04 下打开 qemu, 通过主机 (Ubuntu) 向 qemu 内 guest os 的网络地址 ping, 如果能收到正确返回, 说明 e1000 驱动的基本功能可以在 ucore 中使用, 即完成了基本移植工作。

## QEMU User Networking (SLIRP)



演示



# 其他

- 因为 ucore slab 最大支持 128K, 而 e1000 module > 128k, 故修改 mm/slab.c MAX\_SIZE\_ORDER
- 在加载 linux module 时, 因为 linux module 和 ucore module 大小不同, 所以会出现关键位置如 init, exit 位置不对, 所以要在 ucore module 结构中增加 padding 使之对齐。但因为每添加新模块后, kconfig.h 会有修改, 因此在移植新的 linux 模块的时候, 可能要对 padding 的大小进行修改
- 没有 device model



谢谢

# 主要代码修改描述

## ■ lkm 改造步骤

- 1. 定义 init , exit 函数及 struct module 等模块必备的函数和头文件
- 2. 补全 \\*.dummy.c 中的函数
- 3. 若模块需要使用内核中的全局变量或函数，需调用 EXPORT\_SYMBOL 函数将该符号导入符号表，同时在该模块的 .c 文件中声明该变量
- 4. 为该模块添加 makefile 及 \\*.dep 文件，并在 modules 目录下的 makefile 中添加