

Lab6 说明文档

5130309029 沈匡济

完成内容说明

由于时间和难度问题，这个 lab 自己只做了一部分，助教运行 `make grade` 可以看到通过了前两个点。这个 lab 要我们写的是一个网卡驱动，个人感觉对 `partA` 要做的内容有了一个大概的了解，只是实现起来可能十分复杂，具体详见下文。

Part A: Initialization and transmitting packets

Exercise1

在 `trap.c` 的每个时钟中断里加一句 `time_tick()` 调用，然后在系统中断路由里设置好 `SYS_time_msec` 的 `handler` 函数，在函数里直接调用 `time_msec`。

Exercise2

这里需要阅读大量材料，首先是 intel 的材料，这里稍微阐述一下 `chapter3` 和 `chapter14` 的内容。`chapter3` 描述了 `packet` 的接收、传递、描述符结构和 `TCP` 段及校验和，整个接收的过程分为确认包的存在，地址过滤，然后存储在 `FIFO` 队列中，并转换为本机的数据流。`chapter14` 讲到了 `PCI` 设备的初始化和重置，`PCI` 全称是 `Peripheral Component Interconnect`，在使用前我们通过总线查找到每一个 `PCI` 设备，为其分配 `I/O` 和内存映射。这部分代码已经帮我们实现好，通过观察 `pci` 结构体，我们可以发现每个 `pci` 设备由一个 `vendorID` 和一个 `deviceID` 组成的两个 `key` 来标识，当 `attach` 到一个设备的时候，lab 中提到用 `pci function` 来进行初始化，这一过程不是很明确，于是我看了 `pci_scan_bus` 中的代码：

```
for (df.dev = 0; df.dev < 32; df.dev++) {
    uint32_t bhlc = pci_conf_read(&df, PCI_BHLC_REG);
    if (PCI_HDRTYPE_TYPE(bhlc) > 1) // Unsupported or no device
        continue;

    totaldev++;

    struct pci_func f = df;
    for (f.func = 0; f.func < (PCI_HDRTYPE_MULTIFN(bhlc) ? 8 : 1);
         f.func++) {
        struct pci_func af = f;

        af.dev_id = pci_conf_read(&af, PCI_ID_REG);
        if (PCI_VENDOR(af.dev_id) == 0xffff)
            continue;

        uint32_t intr = pci_conf_read(&af, PCI_INTERRUPT_REG);
        af.irq_line = PCI_INTERRUPT_LINE(intr);

        af.dev_class = pci_conf_read(&af, PCI_CLASS_REG);
        if (pci_show_devs)
            pci_print_func(&af);
        pci_attach(&af);
    }
}
```

可以看到，lab 中列出了 `pci function` 的结构，`pci_conf_read` 应该是读出一个 `pci` 的信息，这里用一个循环变量来遍历，然后每次新生成一个结构体，设置好 `PCI` 的 `ID`、`INTERRUPT`、

CLASS 寄存器，CLASS 是指出该 PCI 是用来干什么的，是存储控制或是网络控制等，这里还是有很多疑问，比如 PCI_HDRTYPE_MULTIFN 的意义。经过思考，我认为是指出一个有多个功能的 device。

PCI Interface

Exercise3

这里让我们初始化一个 e1000 网卡，目前只需要通过 `pci_func_enable` 函数来激活它，不用做其它任何操作。

Memory-mapped I/O

为 E1000 的 BAR0 提供映射，按照题目中说的将它映射到了 KSTACKTOP 和 KERNBASE 之间，打印寄存器得到了正确的值：

```
device status register: 80080783
```

Exercise5

这里开始要为发送信息的描述符表做初始化，具体做法是分配一段双字对齐的内存，然后再在上面设置好寄存器，包括 TDLEN（长度）、TDH（头）、TDT（尾）、TCTL（传送控制）等，之后的 14.5.1 节涉及到信号，根据 lab 的描述子章节是不在要求范围内的。

这里的阅读量依然很大，之前的几个 exercise 都很简单，但是看了下这个 lab 后面需要自己写的东西太多了，时间不剩多少，没有了完善的注释自认为没有能力实现全部功能，因此终止了这个 lab 的工作，望谅解！