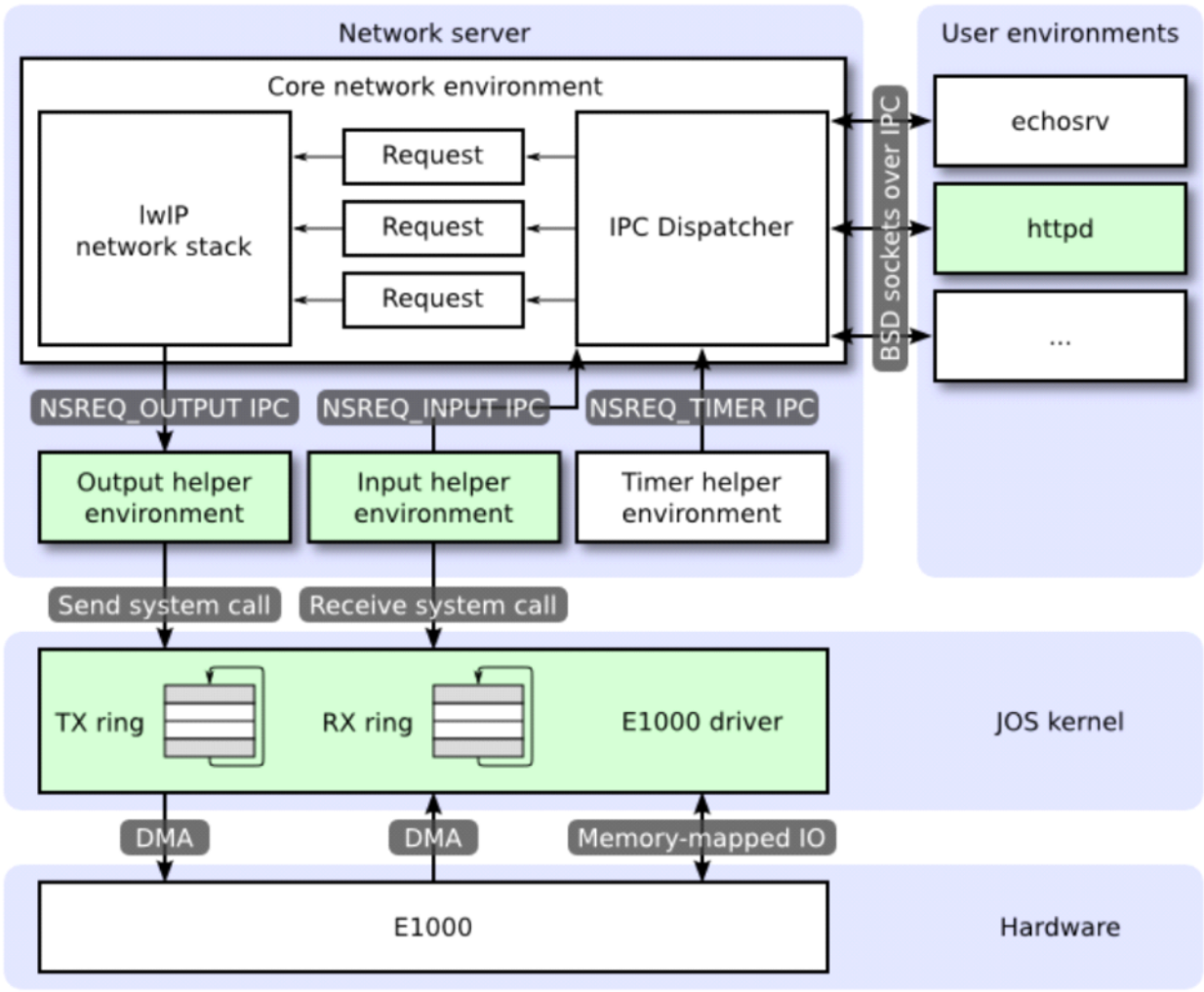
Lab6 笔记

2019年6月5日　　14:36

这个lab需要实现网卡驱动，网络服务端有4部分组成：
Core network server environment
Input environment
Output environment
Timer environment
我们需要实现的是下图高亮部分



partA

Lab5实现的OS没有时间的概念。所以我们需要添加它。现在有硬件产生的时钟中断，每10ms发送一次；每次时钟中断我们都增加一个变量的值来表示，这个在kern/time.c文件内实现。

Exercise1

- 修改trap.c中的trap_dispatch函数，每次clock interrupt都调用time_tick函数
- 实现kern/syscall.c中的sys_time_msec，调用time_msec，time_msec为时钟变量值+1
- 在syscall函数中分发sys_time_msec。

Exercise3

- 在kern/pci.c的pci_attach_vendor数组中添加e1000的entry，
- 在kern/e1000.c中的pci_e1000_attach函数中对E1000调用pci_func_enavle

Memory-mapped I/O

软件通过mmio和E1000进行通信，mmio在比较高的地址，用KADDR无法访问。因此需要创建新的你内存映射

Execise4

这个exercise主要是在e1000初始化的时候为e1000映射指定的内存，并把内存赋值给base变量
用lab4完成的函数mmio_map_region映射内存

DMA

E1000使用内存直接访问(DMA)来的读写数据包，驱动程序需要为发送和接收队列分配内存。
用一个循环队列来表示，分别有头指针和尾指针，两个指针之间为可用的描述符。这些数组的指针和描述符中的缓冲区地址都必须是物理地址，因为迎接直接冲物理地址读取，不经过MMU

C structure

Exercise2中的手册给出了描述符结构和transmit初始化的步骤

Exercise5

在初始化5个描述descripter数组的寄存器，e1000.h中有很多有用的宏定义
初始化TCTL寄存器

Exercise6

- 实现e1000_tx函数传输数据包
- 检查下一个描述符是否free，设置下一个描述符，更新TDT

Exercise7

- 完成kern/syscall.c中的sys_net_send函数
- User_mem_check检查传入的地址权限是否符合，调用e1000_tx函数

Exercise8

- 实现net/output函数
- 用ipc_rev读入packet请求
- 用sys_net_send发送包到设备驱动

Exercise10

- 根据手册14.4设置receive队列并配置E1000
- 用card的mac地址设置RAL和RAH，可以硬编码52:54:00:12:34:56
- 注意字节顺序，MAC地址是从低地址到高地址表示的
- 设置buffer的大小的2048

Exercie11

- 完成e1000_rx和sys_net_recv函数
- E1000_rx
    - 检查DD位
    - 拷贝缓冲区数据到参数buf中
    - 重置DD和EOP位
    - 返回拷贝长度
- Sys_net_recv，检查权限为并调用e1000_rx函数

Exercie12

- 实现input.c
- 用sys_net_recv接受网络包，如果没有网络包，调用sys_yield释放cpu资源
- 如果有包，把包长度赋值给nsipcbuf中的pkt的jp_data变量
- 并拷贝网络包内容到nsipcbuf.pkt.jp_data
- 调用sys_ipc_try_send发送该内容到ns_envid环境

Exercise13

- 实现user/httpd.c中的send_file和send_data函数
- Send_file
    - 打开请求的url，并返回fd描述符
    - 用fstat获取该描述符基本信息
    - 如果是目录，返回404错误
    - 之后就是给定的代码依次发送header, size, content_type和header_fin
- Send_data
    - 检查数据包大小
    - 读取文件内容到buf中
    - 把buf发送到请求指定的sock