JOS Lab6 文档

杨健邦 515030910223

本文档描述了本次 LAB 各个 Exercise 实现的方法或者我自己的理解。至于对 Question 的解答以及以及 Challenge 的实现方法，请阅读 answers-lab6.txt。

Part A: Initialization and transmitting packets

Exercise 1.
在 trap.c 中，在处理时钟中断的时候，调用 time_tick()来计时，但是要注意，每一次时钟中断都会向所有 CPU 发送，为了防止重复计时，当且仅当 CPU0 处理时钟中断的时候调用 time_tick() 。

Exercise 3.
查阅 E1000 的手册，得到相关网卡设备的 DEVICE ID 和 VENDOR ID，加到数组里面，在 e1000 的 attach 函数中，调用一下 pci_func_enable 即可 。

Exercise 4.
在 e1000 的 attach 函数中，将 bar 0(物理地址)映射到 KSTACKTOP 的地方，权限位应该置为 PTE_P | PTE_W | PTE_PCD | PTE_PWT，因为这里 KSTACKTOP 的地方之前没有映射，因此它不会出现在 TLB 之中，也不需要 invalidate TLB。用一个带有 volatile 关键字的全局变量 bar0 来记录其起始的虚拟地址，也就是 KSTACKTOP，当映射完成之后，assert(bar0[E1000_STATUS] == 0x80080783)来确保映射成功。

Exercise 5.
通过设置全局变量的方式，使得 descriptor ring buffer 所在的物理页在链接和加载程序的时候被自动分配在连续的物理地址空间，在设置全局变量的时候，加上 __attribute__((aligned(16)))保留字来使得链接器会将这个 buffer 放到 16 字节对齐的地址。

同时为了防止出现 cache 与 memory 之间存在不一致问题，在 attach 的时候，将 descriptor ring buffer 以及 packet data buffer 所在的物理页全部映射成 PTE_PCD 和 PTE_PWT 的形式(禁用缓存以及写穿)。

其它直接根据手册一步步初始化即可。

Exercise 6.
发送网络包的函数的函数名是 e1000_transmit，由于传入的指针参数和长度的有效性在 syscall 里面已经进行检验了，因此这个函数里面不再需要检验。通过检查 DD 是是否被置上来判断 ring buffer 是否已满(所有 desc 里面的 DD 位在 attach 初始化的时候就已经被清掉了，由于不支持长包，所以所有的 desc 在初始化的时候通过也把 EOP 位给置上了)。

Exercise 7.
与 sys_ipc_try_send 类似，我定义的新的系统调用的名称是 sys_net_try_transmit。这个系统调用首先使用 user_mem_assert 来确保用户传进来的指针是有效的，以及指针所指向的空间足够大道容纳一个网络数据包的内容，之后直接调用 e1000_transmit 函数即可 。

Exercise 8.
output 函数是一个死循环，首先使用之前的 ipc_recv 接口，不断尝试，接收一个 ipc 消息，当接收到 IPC 且 IPC 是来自 ns 这个 env 且 IPC 的值是 NSREQ_OUTPUT 的时候，才会进入另一个循环，不断调用 sys_net_try_transmit 来尝试发送网络包。

Part B: Receiving packets and the web server

Exercise 10.
对于 receive 所需要的物理空间分配，跟 transmit 的物理空间分配类似，这里就不多叙述。初始化工作同样是在 attach 函数中，根据手册一步步进行即可。

Exercise 11.
和 e1000_transmit 相似，receive 函数被命名为 e1000_receive，用户传入的指针的有效性检查已经在 syscall 层进行检验了。如果没有要接收的包，则返回一个-E_NO_RECV_PACKET 错误码让用户自行处理。

Exercise 12.
写一个 while 循环，首先通过 syscall 来接收一个网络包，如果没有收到网络包的话，则调用 sys_yield()来放掉 CPU 资源，等待下次被唤醒继续查看是否有收到网络包。如果有收到网络包，则新分配一个页，把网络包的数据拷到这个页之中，再通过 IPC 将数据发给 ns env，这里不能使用之前的页，因为当接收到一个新的网络包时，不知道 ns env 有没有处理上次通过 IPC 发给其的数据，重复使用同一个页会导致数据因覆盖而被丢失。

Exercise 13.
通过 open 来判断文件或目录是否存在，通过 fstat 得到一个 struct Stat，然后判断路径是否是一个目录，如果不存在或者是一个目录则调用 send_error 来返回错误。在 send_data 函数中，有个循环，每次从文件中读入至多 1518 个 Byte，然后写到 sock 对应的 fd 中，直到不能从文件中读入更多的 Byte 为止。