

Project 5 Device Driver 设计文档

中国科学院大学

段江飞

2018 年 12 月 18 日星期二

1. 网卡驱动

(1) 任务一实现时，你初始化了几个接收描述符 (RDES)，每个接收描述符中设置了哪几个域的值，所设置的域的各自含义是什么？

任务一中初始化了 64 个接收描述符，对每个接收描述符，`rdes0` 设置为 0，第 32 位为 0 表示未被 DMA 拥有，还没开始接收数据；`rdes1` 设置第 24 位表示 `buff2` 中的地址指向下一个描述符的地址，第 10 到 0 位设置为 0x400 表示 `buff1` 的大小为 1KB，其余位设为 0，最后一个接收描述符的 `rdes1` 的第 25 位也设为 1，表示环形描述符链表的最后一个，其 `buff2` 域指向第一个接收描述表符；`rdes2` 设置为接收 `buff` 的物理地址，是在 `unmap` 空间的一段内存，将高 3 位抹掉；`reds3` 则设置为下一个接收描述表符的物理地址，最后一个接收描述表符的 `rdes3` 设置为第一个接收描述表符的物理地址。

(2) 任务一实现时，每个发送描述符中设置了哪几个域的值，所设置的域的各自含义是什么？

任务一中初始化了 64 个发送描述符，对每个发送描述符，`tdes0` 设置为 0，第 32 位为 0 表示未被 DMA 拥有，还没开始传输数据；`tdes1` 设置第 24 位表示 `buff2` 中的地址指向下一个描述符的地址，第 10 到 0 位设置为 0x400 表示 `buff1` 的大小为 1KB，其余位设为 0，最后一个接受描述符的 `tdes1` 的第 25 位也设为 1，表示环形描述符链表的最后一个，其 `buff2` 域指向第一个发送描述表符；`tdes2` 设置为发送 `buff` 的物理地址，是在 `unmap` 空间的一段内存，将高 3 位抹掉；`teds3` 则设置为下一个发送描述表符的物理地址，最后一个发送描述表符的 `tdes3` 设置为第一个发送描述表符的物理地址。

(3) 任务二实现时，检查是否有数据包到达网卡这一操作是在哪个流程中执行的？例如是时钟处理流程，还是接收线程本身，或者是其他流程中？

都有，首先在调用 `sys_net_recv` 之后，接收线程先检查第 64 个接收描述表是否已经接收到，如果没有，则阻塞等待，之后会在时钟中断流程中检查是否有接收到包，如果已经接收到了，则会调用 `check_recv`，检查接收到的包。

(4) 任务三实现时，你的设计中，每接收到几个网络包时会产生一次中断？

任务三中，每接收 64 个包会产生一次中断。

(5) DMA 接收和发送描述符采用环形链表和链型链表都是可以的，你认为使用环形链表

和使用链型链表有什么区别？

使用环形链表能够用比较少的描述表符一次接收到更多的包，重复利用。

2. Bonus 设计

（1）相比较任务三而言，在 Bonus 中你是否有新增设计，以满足 Bonus 对网卡接收性能的要求？若有，请说明你的新增设计和用途。

将接受描述表符增加到了 256 个，这样接受 256 个包才会触发一次中断，减少了处理时间，这样在 10s 测试，能受到 19403 个包，达到 1Mb/s 的收包速度。

（2）设计或实现过程中遇到的问题和得到的经验（如果有的话可以写下来，不是必需项）

本来是连续调用 task3，但是性能很低，后来改成在接收包的系统调用里面添加了连续接收包，一次接收完成之后，就发起另一次接收，但这样在不太符合要求，后来又改成了增大接收描述表符。

3. 关键函数功能

请列出上述各项功能设计里，你觉得关键的函数，及其作用

初始化描述表符

```
send_desc_init(mac_t *mac)
recv_desc_init(mac_t *mac)
```

接收发送

```
uint32_t do_net_recv(uint32_t rd, uint32_t rd_phy, uint32_t daddr)
void do_net_send(uint32_t td, uint32_t td_phy)
```