

《网络与设备》章节练习

1. Linux设备驱动的主要抽象是那些？请列举sysfs文件系统中子项，并且指出他们之间的关系。

- Device (设备): 用于抽象系统中所有的硬件
 - 包括CPU和内存
- Bus (总线): CPU连接Device的通道
 - 所有的Device都通过bus相连
- Class (分类): 具有相似功能或属性的设备集合
 - 类似面向对象程序设计中的Class
 - 抽象出一套可以在多个设备之间共享的数据结构和接口
 - 从属于相同Class的设备驱动程序，直接继承

2. ARM 通用中断控制器（GIC）指出了中断完成需要经过两个阶段：优先级下陷（priority dropping）以及中断未激活（interrupt deactive）？他们和Linux中的中断机制有什么关系（上半部和下半部）。

- GIC将中断完成分两步走，是为了使响应和完成可以异步进行
- Linux通过将中断处理程序分为两半，使得中断处理时可以在handler中执行较长的任务，并且不会长时间保持中断阻塞。top half将请求放入队列（或设置flag），将其他处理推迟到bottom half。这使得top half在bottom half仍在工作时处理新的中断。

3. 在数据传输的场景中，如果接受者（receiver）无法跟上发送者（sender）发送数据包的速度，接受者需要如何处理？（提示选择轮询或者中断的方式）

- 轮询或者中断的方式都优缺点
 - 若使用轮询的方式，若收端来不及处理数据，等buffer满了之后会有数据丢失
 - 若使用中断的方式，收端在处理中断上下文中时，后来数据会收不到
- 应该采用
 - 静态配置法：UART驱动需要在初始化时指定“波特率”，限制发包速率
 - 动态协商法：TCP设计了流量控制机制，发端逐步试探出收端收包能力的上限、进行动态调整

4. 简单描述当接受者（receiver）从网卡接受到了网络包之后的数据传输流程。在真实情况下，Linux中会发生几次上下文的切换，ChCore中呢？

- 传输流程
 1. 网卡收到包后把包存入rx ring中
 2. 网卡发送中断，cpu收到后调用中断处理函数ISR，并发出一个软中断softirq

3. os申请sk_buffer
 4. 从rx ring将包读入sk_buffer
 5. softirq被调度
 6. ip层解析sk_buffer
 7. tcp层解析sk_buffer
 8. 将包从socket接收队列拷贝到用户空间
- 有2次上下文切换，因为有一次中断

5. Linux是如何使用skb机制在内核中实现零拷贝的？skb拷贝在什么时候会发生？请同时考虑浅拷贝（skb_clone）以及深拷贝（skb_copy）。

- sk_buff本身不存储报文，通过指针指向真正的报文内存空间，在各层传递时只需调整指针相应位置即可。
- 在tcpdump等抓包工具中会用到浅拷贝，克隆出新的sk_buff控制结构，和原来指向同一报文
- 修改报文如NAT地址转换中会用到深拷贝，需要同时复制sk_buff以及指向的报文

6. Intel DPDK使用了哪一种技术来提升网络性能？如果你是ChCore团队的网络设计者，你会选者什么方式来优化ChCore为内核的网络服务，你可以通过学习"Snap: a microkernel approach to host networking", Marty et al., SOSP'19来回答该问题

- DPDK使用的技术
 1. 抛弃中断，使用轮询模式
 2. 使用大页（2MB）减少TLB miss
 3. 控制平面与数据平面相分离
 4. 用多核编程代替多线程技术
 5. CPU 核尽量使用所在 NUMA 节点的内存，避免跨NUMA内存访问
- 我会使用大页来优化ChCore为内核的网络服务