下面分四个部分来分析：

virtio device注册

virtio driver注册

前端发送

前端接受

virtio规范：

<https://docs.oasis-open.org/virtio/virtio/v1.1/virtio-v1.1.pdf>

内核中virtio子系统（前端）与后端的交互都遵守这个协议。

下面以mmio类型的virtio为例进行分析。

下面是协议中规定的后端virtio设备的寄存器布局信息（4.2.2）：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 偏移 | 读写 | 功能 | 备注 |
| Magic Value（幻数） | 0x0 | 只读 | 固定为0x74726976，小端格式存储，等价于”virt”字符串 | 前端代码初始化时回检查该值 |
| Version（版本） | 0x4 | 只读 | 设备版本号，最新的是2，老的设备是1 |  |
| Device ID （设备ID） | 0x8 | 只读 | Virtio子系统设备ID，表示不同类型的virtio设备，比如network card，block device，console，rpmsg，SCSI host等，参考 |  |
| Vendor IDI  （厂商ID） | 0xC | 只读 | Virtio子系统厂商ID |  |
| DeviceFeatures  （设备特性） | 0x10 | 只读 | 返回一个32位的flag标志位，可以通过DeviceFeaturesSel控制输出哪一组特性集：DeviceFeaturesSel\*32~ DeviceFeaturesSel\*32+31 |  |
| DeviceFeaturesSel  （设备特性选择） | 0x14 | 只写 | 设备（宿主机）特性集选择 |  |
| DriverFeatures  （驱动特性） | 0x20 | 只写 | 能够被前端驱动支持并激活的设备特性，通过DriverFeaturesSel可以控制当前配置的是那一组特性集 |  |
| DriverFeaturesSel  （驱动特性选择） | 0x24 | 只写 | 激活的（客户机）特性集选择 |  |
| QueueSel  （队列选择） | 0x30 | 只写 | 虚拟队列索引，从0开始 |  |
| QueueNumMax | 0x34 | 只读 | 被QueueSel选中的虚拟队列中元素的最大个数 |  |
| QueueNum | 0x38 | 只写 | 驱动实际使用的虚拟队列（QueueSel）中元素的个数 |  |
| QueueReady | 0x44 | 读写 | 驱动向这个bit写1来通知设备它可以从这个虚拟队列（QueueSel）上执行请求 |  |
| QueueNotify | 0x50 | 只写 | 驱动向这个寄存器写值表示虚拟队列有新的buffer需要处理。如果还没有协商VIRTIO\_F\_NOTIFICATION\_DATA，那么需要写入虚拟队列的编号，否则需要写入如下的数据结构：  le32 {  vqn:16; // 虚拟队列的编号  next\_off:15;  next\_wrap:1;  }，这个结构中包含的信息使得后端代码不需要查询内存中的virtqueue就可以知道虚拟队列中可用的数据，使效率得到提升 |  |
| InterruptStatus | 0x60 | 只读 | 驱动读写这个寄存器来获得设备中断状态，bit0表示设备使用了虚拟队列中的一个bufer，bit1表示设备的配置发生变化 |  |
| InterruptACK | 0x64 | 只写 | 写入InterruptStatus定义的值来通知设备对应的中断已经处理完毕 |  |
| Status | 0x70 | 读写 | 获取或者设置当前设备的状态，写入0会触发设备复位 |  |
| QueueDescLow | 0x80 | 只写 | Vring中desc的起始物理地址的低32位 | Payload |
| QueueDescHigh | 0x84 | 只写 | Vring中desc的起始物理地址的高32位 |  |
| QueueDriverLow | 0x90 | 只写 | Vring中avail的起始物理地址的低32位 | 前端->后端 |
| QueueDriverHigh | 0x94 | 只写 | Vring中avail的起始物理地址的高32位 |  |
| QueueDeviceLow | 0xa0 | 只写 | Vring中used的起始物理地址的低32位 | 后端->前端 |
| QueueDeviceHigh | 0xa4 | 只写 | Vring中used的起始物理地址的高32位 |  |
| ConfigGeneration | 0xfc | 只读 | 用于获取判断设备的config是否在driver配置期间发生变化，具体是驱动先读取该值，然后根据config的内容开始配置，配置结束后，再读写该值，如果两次读到的值相同的，配置完毕，否则还还需要从新读取config进行配置，知道配置前后从这个寄存器读到两个相同的值 |  |
| Config | 0x100+ | 读写 | 字节对齐，存放设备特定的配置，空间大小取决于设备和驱动 |  |

virtio device创建的创建