**VirtIO Block**

**1 背景**

见[VirtIO介绍](https://xiaomi.f.mioffice.cn/wiki/wikk49h3wuek9HIAixufJC85yMf) [VirtIO框架](https://xiaomi.f.mioffice.cn/wiki/wikk4Rawgsw80QqpbiBxypTxdLd)

**2 使用方法**

见[VirtIO-Block](https://xiaomi.f.mioffice.cn/wiki/wikk409u9B5RxhqoEoYyjZAPtjh)

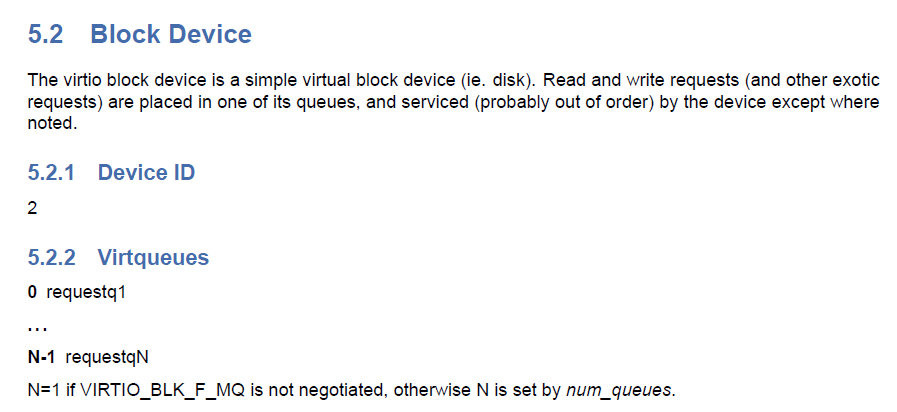
**3 代码实现**

**3.1 Virtio Spec**

**3.1.1 设备的ID和Virtqueue的数量**

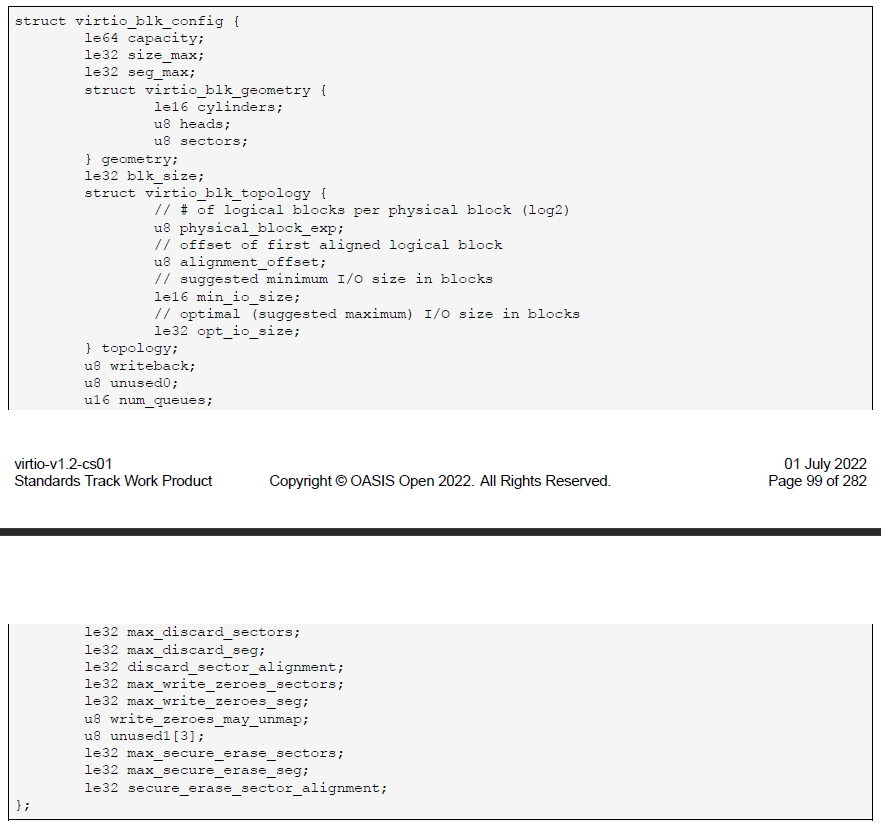
实现virtio driver的第一步是根据virtio的spec: https://docs.oasis-open.org/virtio/virtio/v1.2/csd01/virtio-v1.2-csd01.pdf

获取对应device的ID，virtqueue的个数和交互的格式，如下：



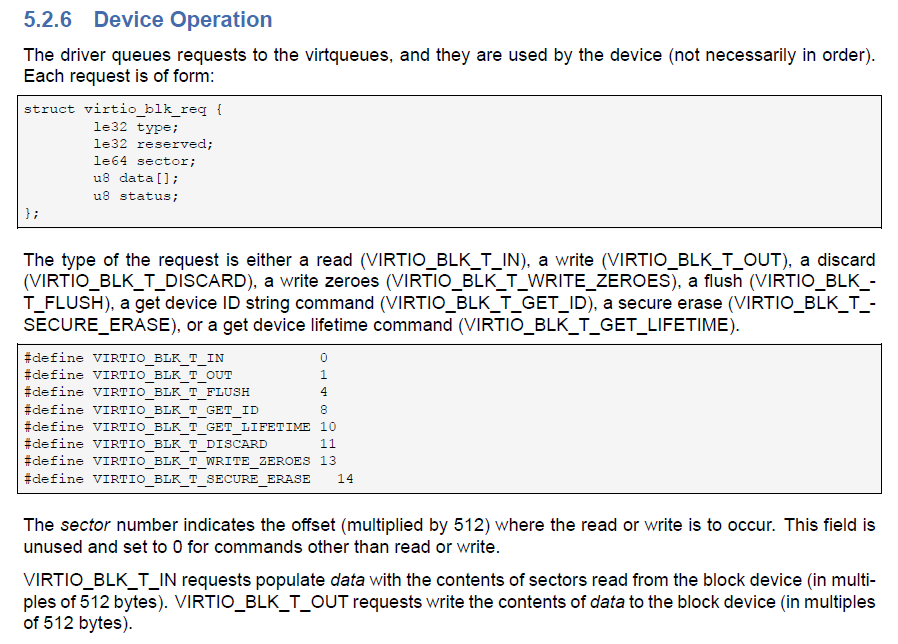
可以得到Block Device的Device ID是2，Virtqueue最小支持0，最大数量N由virtio\_blk\_config中的num\_queues来确定，见下方

**3.1.2 Virtio Block Config**



**3.1.3 交互方式**

5.2.6这一章有说明具体的交互方式，driver通过发送一个virtio\_blk\_req到device侧，然后device收到request后，将对应的状态和数据填充到buffer中后，返回给driver



**3.2 驱动注册**

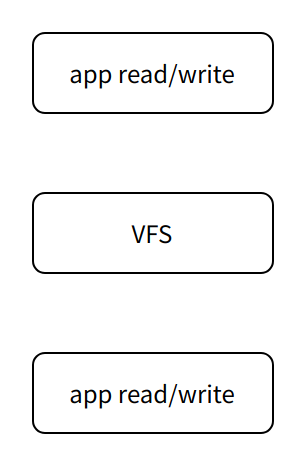
实现如下的struct virtio\_driver

|  |
| --- |
| C static struct virtio\_driver g\_virtio\_blk\_driver = {  LIST\_INITIAL\_VALUE(g\_virtio\_blk\_driver.node), /\* node \*/  VIRTIO\_ID\_BLOCK, /\* device id \*/  virtio\_blk\_probe, /\* probe \*/  virtio\_blk\_remove, /\* remove \*/ };  /\* virtio\_register\_blk\_driver()被virtio框架显式调用 \*/ int virtio\_register\_blk\_driver(void) {  return virtio\_register\_driver(&g\_virtio\_blk\_driver); } |

将driver注册到virtio-bus中，有对应的设备时，会自动匹配后回调驱动的virtio\_blk\_probe()函数

**3.2 工作流程**

这里简要描述virtio-blk driver和device的交互流程，主要针对读写过程，流程图如下：



rpmsg类似实现[rpmsg dev/mtd/blk](https://xiaomi.f.mioffice.cn/wiki/wikk4pW3HypPT0mXvUqSfCY7bMf)。