### DPDK 网络数据收发编程与协议栈

Dpdk提供三组数据交互方式，igb\_uio，VFIO，KNI。

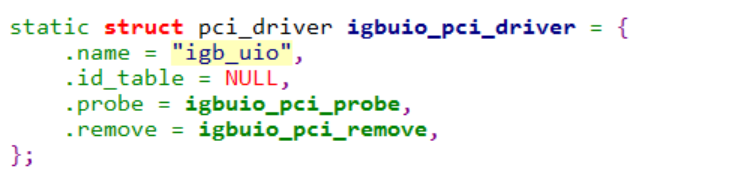
#### Igb\_uio

Igb\_uio分三个部分，igb\_uio内核模块，内核uio框架，uio用户接口

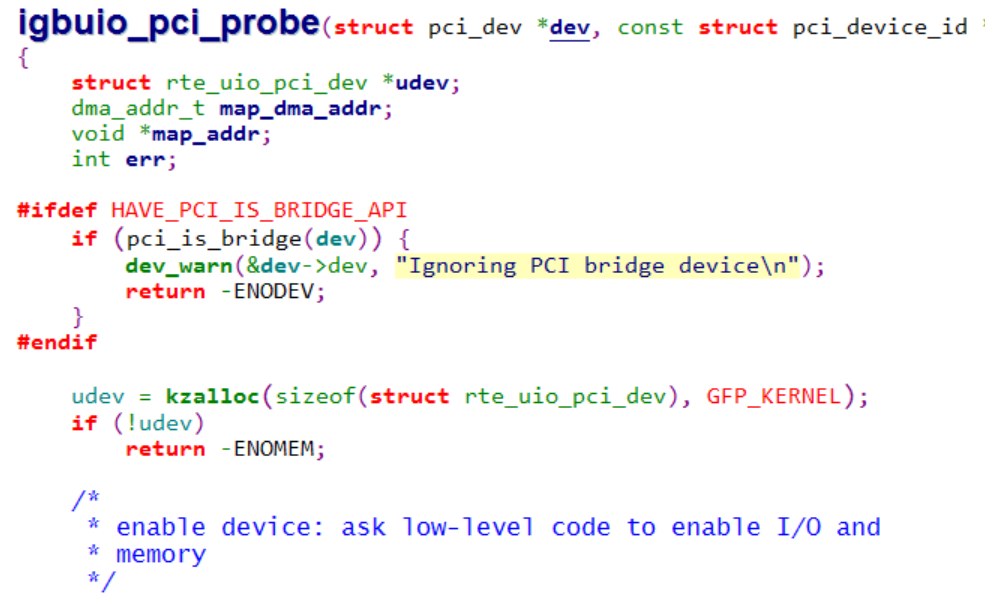
##### igb\_uio内核模块

igb\_uio驱动主要做的就是注册一个pci设备，在DPDK工具dpdk\_nic\_bind.py绑定NIC的时候这个驱动会probe到这个设备，进行相关配置。之后会注册一个UIO设备，probe函数会将记录设备的资源比如PCI设备BAR空间的物理地址、大小等信息记录下来传给用户态。注册的UIO设备名为igb\_uio，内核态中断处理函数为igbuio\_pci\_irqhandler，中断控制函数igbuio\_pci\_irqcontrol。注册的主要工作如下：

1. 初始化uio\_device结构体指针，主要包括等待队列wait、中断事件计数event、次设备号minor等。
2. 在/dev目录下创建了一个uio设备，设备名为uioX，X对应的就是次设备号minor。
3. 在/sys/class/uio/uioX/目录下创建maps和portio接口。
4. 注册中断和中断处理函数uio\_interrupt







##### Uio框架

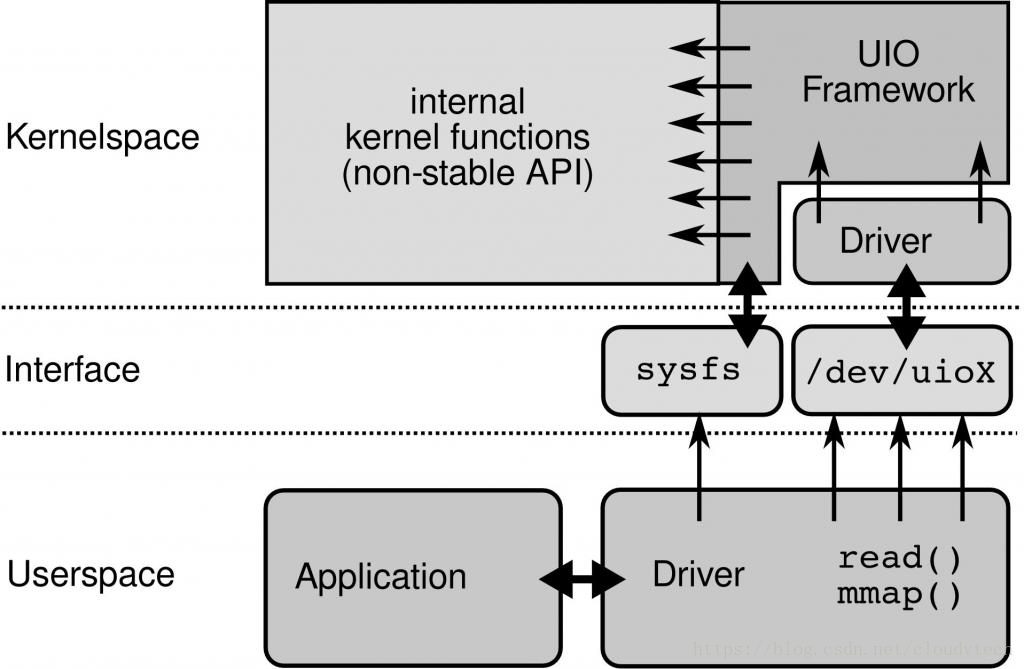
在UIO中，使用read/mmap在user space存取设备对应的内存区域；但是UIO还是有一小部分中断处理在内核中，这个个中断处理的主要职责是开关中断，并将中断计数值加一。户空间驱动要监测一个设备中断，它只需阻塞在对/dev/uioX的read()操作上， 当设备产生中断时，read()操作立即返回。

**内核态的职责：**

1. 分配和记录设备需要的资源和注册uio设备
2. 使能设备
3. 申请资源
4. 读取并记录配置信息
5. 注册uio设备
6. 必须\*在内核空间实现的小部分中断应答函数

**用户态职责：**

1. 获取中断事件(read/poll)
2. 处理中断(读写数据)



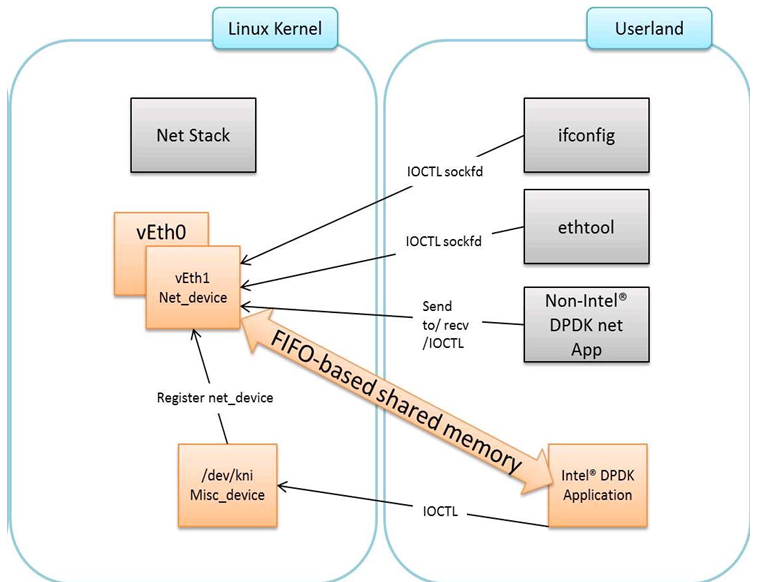
##### Uio用户接口





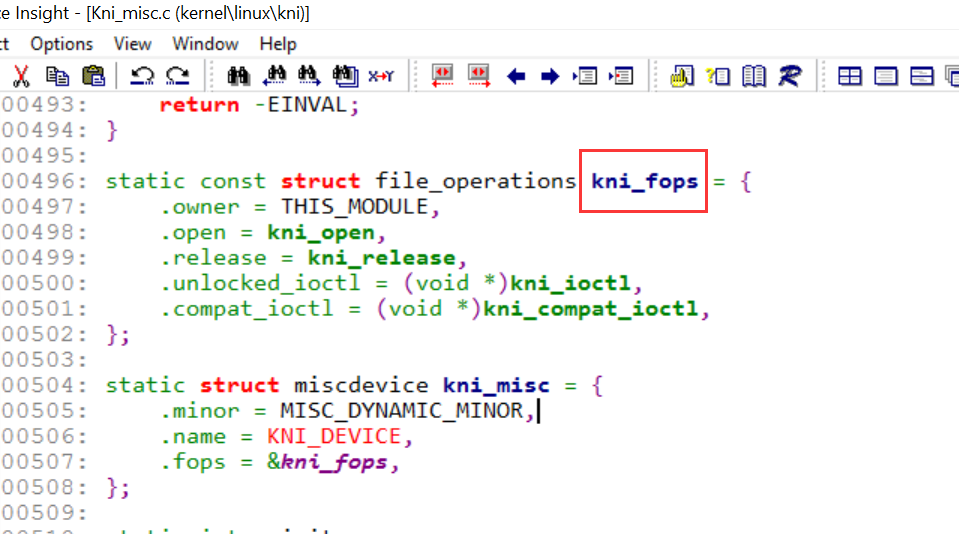


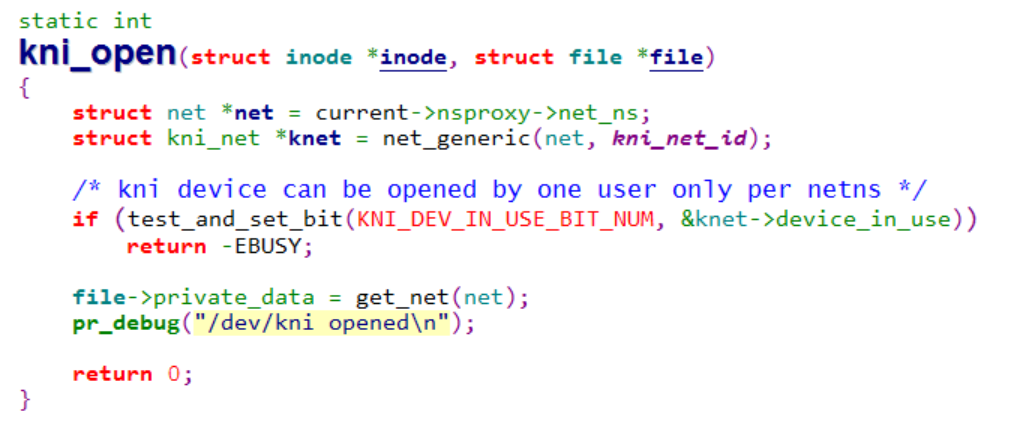
#### KNI原理

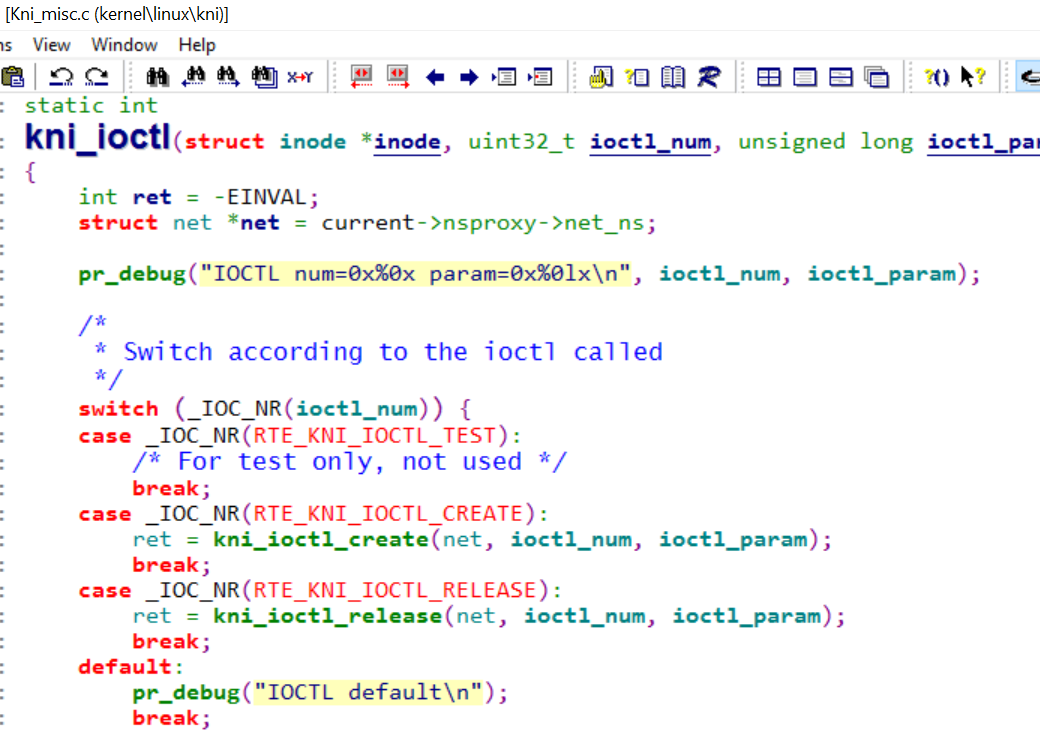


KNI必然要也需要内核模块的支持，即rte\_kni.ko。其共有三个参数，分别是lo\_mode，kthread\_mode和carrier。**lo\_mode**可配置为lo\_mode\_none，lo\_mode\_fifo，和lo\_mode\_fifo\_skb，默认为lo\_mode\_none。另外两个在实际产品中基本不会用到。**kthread\_mode**可配置为single和multiple，默认为single。**carrier**可配置为off和on，默认为off。模块初始化函数kni\_init也非常简单。除了解析上面的参数配置外，比较重要的就是注册misc设备和配置lo\_mode。

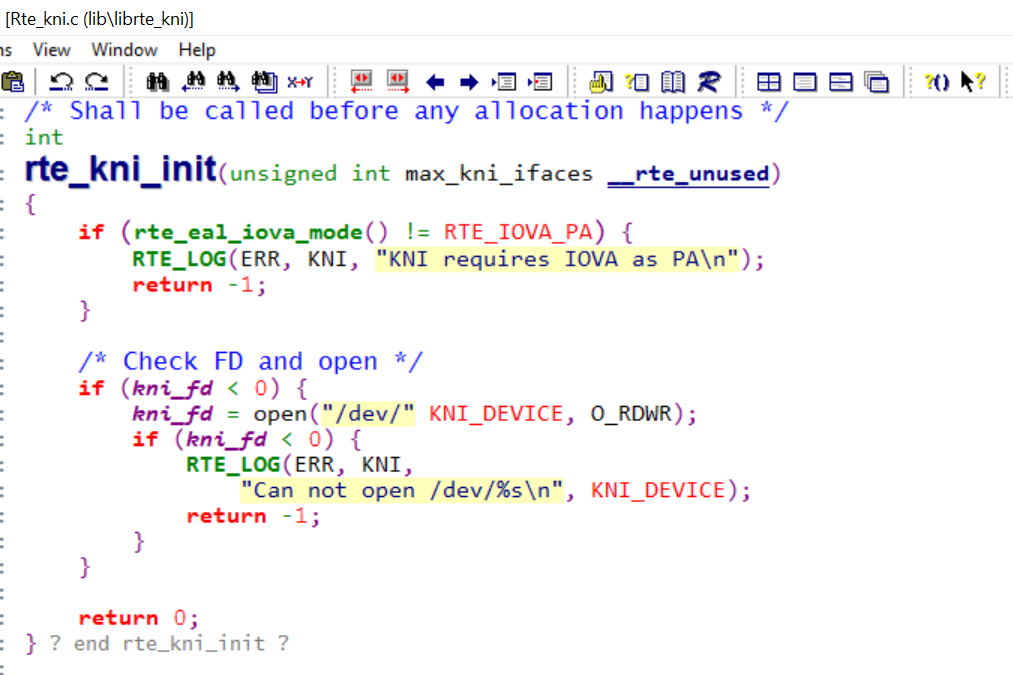
##### KNI内核模块实现

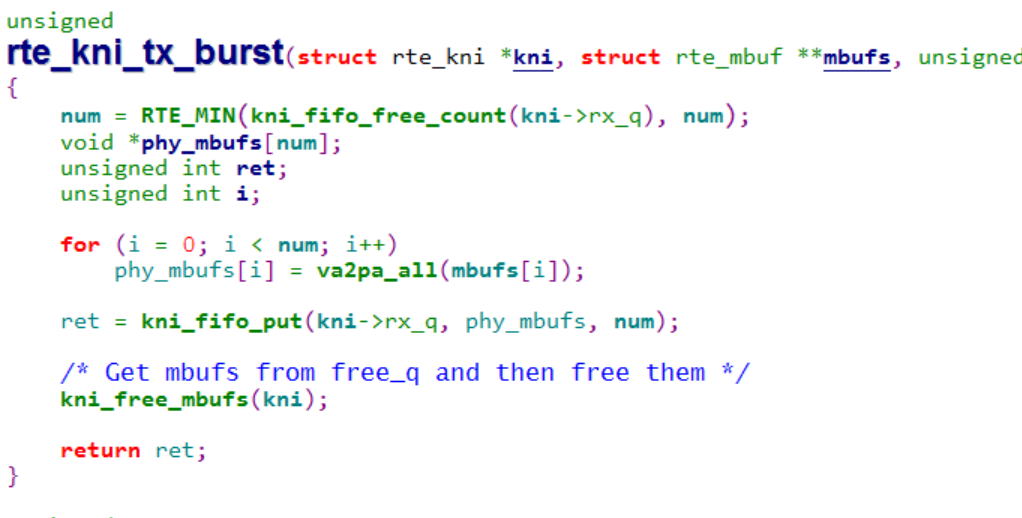


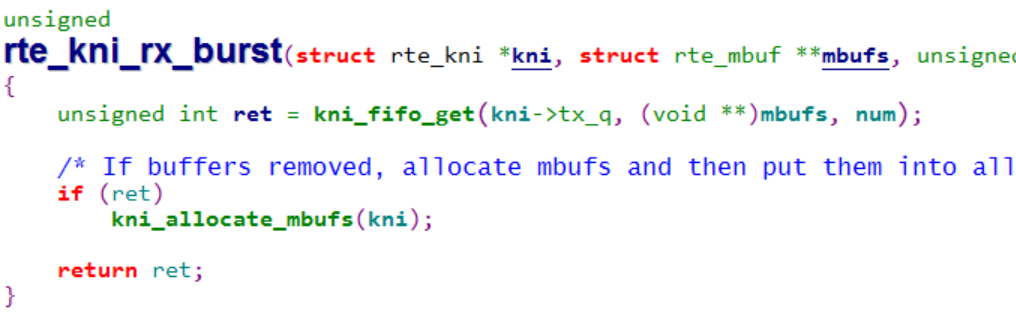




##### KNI用户接口







#### VFIO

##### VFIO原理

VFIO就是内核针对IOMMU提供的软件框架，支持DMA Remapping和Interrupt Remapping，这里只讲DMA Remapping。VFIO利用IOMMU这个特性，可以屏蔽物理地址对上层的可见性，可以用来开发用户态驱动，也可以实现设备透传。

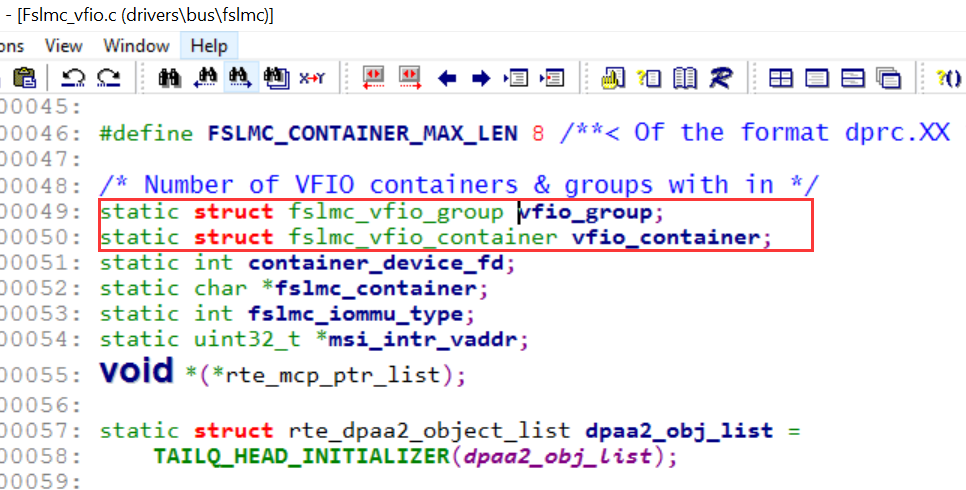
概念介绍

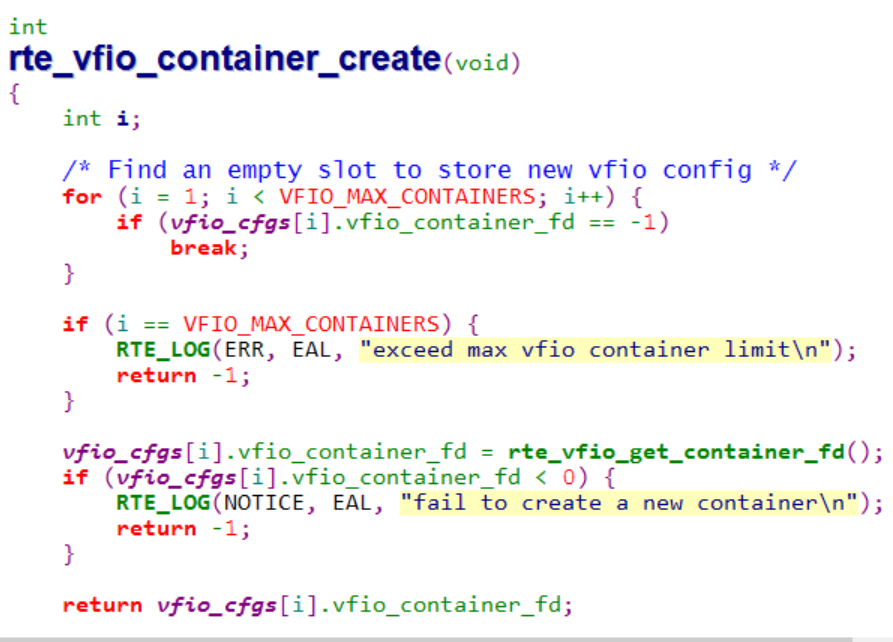
先介绍VFIO中的几个重要概念，主要包括Group和Container。

1) Group：group 是IOMMU能够进行DMA隔离的最小硬件单元，一个group内可能只有一个device，也可能有多个device，这取决于物理平台上硬件的IOMMU拓扑结构。 设备直通的时候一个group里面的设备必须都直通给一个虚拟机。 不能够让一个group里的多个device分别从属于2个不同的VM，也不允许部分device在host上而另一部分被分配到guest里， 因为就这样一个guest中的device可以利用DMA攻击获取另外一个guest里的数据，就无法做到物理上的DMA隔离。

2) Container：对于虚机，Container 这里可以简单理解为一个VM Domain的物理内存空间。对于用户态驱动，Container可以是多个Group的集合。

##### VFIO模块实现







##### VFIO应用程序接口

