

建链成功后，创建一个IPv6相关的路由接口，可以创建，但是无法删除。且在执行删除操作后，Modem串口不能输入命令了。

unregister_netdevice: waiting for nbif2 to become free. Usage count = 2

如果直接从串口行中调用wbctl delif nas2 nbif2，也会出现相同的效果。

经过分析代码，在删除接口的时候，会调用netdev_wait_allrefs，等到所有的引用计数释放。如果dev->refcnt不归零，那么就会一直死循环等待。所以出现串口无法命令只能回车。

dev->refcnt是通过dev_hold和dev_put两个函数增加和减少的。问题肯定是由于dev_hold了一个dev，但是没有释放。

但是我们怎么知道是谁调用了dev_hold和dev_put呢？分析打印即可以知道结果。

```
#ifndef DEVPUT_DEBUG

static inline void dev_put(struct net_device *dev)
{
    atomic_dec(&dev->refcnt);
}

static inline void dev_hold(struct net_device *dev)
{
    atomic_inc(&dev->refcnt);
}

#else

#define dev_put(dev) \
    do{if((dev)->name[0] == 'w') \
        { \
            printk("%s[%d], dev_put dev[%s] refcnt=%d\n", __FUNCTION__, __LINE__, (dev)->name, atomic_read(&(dev)->refcnt)) ; atomic_dec(&(dev)->refcnt); \
            dump_stack(); \
        } \
    }while(0)

#define dev_hold(dev) \
    do{if((dev)->name[0] == 'w') \
        { \
            printk("%s[%d], dev_hold dev[%s] refcnt=%d\n", __FUNCTION__, __LINE__, (dev)->name, atomic_read(&(dev)->refcnt)) ; atomic_inc(&(dev)->refcnt); \
            dump_stack(); \
        } \
    }while(0)

#endif
```

南京那边IPv6的改动，很多文件的改动只是把什么dev_get_by_index改成CSP_dev_get_by_index。我求简单，把CSP_的前缀去掉了。

结果发现，平台中用到的CSP_dev_get_by_index，其实是__dev_get_by_index。汗~

上次去了一些，没有去干净。残留了几个。

update :

今天又爆出了wlan0接口无法删除的故障，经过分析发现又是引用计数的问题。

我把宏打开，也用了dump_stack()把函数调用关系打印出来，一个个地对比了一两个小时，没有分析出来哪个调用没有配对。

这个时候再回头看我当初分析故障的过程，真的是通过比对dev_hold和dev_put的调用关系查出来的吗？不是的。通过替换dev_hold和dev_put，比对后，只是发现问题与dev_get_by_index相关。之后再去比较CSP原始代码、我们修改的代码，才定位错误。

是不是上次没有修改干净？毕竟上次只改了IPv6目录。

进入到CSP原始目录，find . -name "" | xargs grep "CSP_dev_get_by_index"，以及find . -name "" | xargs grep "CSP_dev_get_by_name"，找到类似的错误。

还真找到一些。