<http://blog.chinaunix.net/uid-24227137-id-3266449.html>

Linux2.6以后的设备驱动，都是在设备模型的基础上构建的，因此，要编写linux下的设备驱动程序，不论是usb设备，pci设备等，都需要了解设备模型。

设备模型的基础结构体主要是kobject，kset这两个结构体：

struct kobject {

char \* k\_name;

char name[KOBJ\_NAME\_LEN];

struct kref kref;

struct list\_head entry;

struct kobject \* parent;

struct kset \* kset;

struct kobj\_type \* ktype;

struct dentry \* dentry;

};

struct kset {

struct subsystem \* subsys;

struct kobj\_type \* ktype;

struct list\_head list;

struct kobject kobj;

struct kset\_hotplug\_ops \* hotplug\_ops;

};

还有一个subsys结构体，但subsys结构体跟kset差不多，就多了一个互斥访问信号量，因此，就不需要列出了，另外还有一个结构体

struct kobj\_type {

void (\*release)(struct kobject \*);

struct sysfs\_ops \* sysfs\_ops;

struct attribute \*\* default\_attrs;

};

用来表示kobject，kset的类型。

一个kobject结构如下图的kobject 类型部分，而一个kset结构如下图的kset 类型部分，一个kobject加入一个kset，主要是kobject结构体中的相关字段记录了对应的kset信息，

1. 录了kobject所对应 kset，其所指向的是kset所包含的kobject的地址，
2. ②记录了kobject所对应的kset的kset指针，
3. ③记录了kobject的类 型，
4. ④记录了kset所有的kobject的链子，这个链子是一个双向链表，每当有一个kobject加入到当前的kset,就会调用 list\_add\_tail()函数，把要加入kset的kobject连入链表的结尾，最终形成一个链表。



当有另外一个kobject要加入当前的kset，其中的①②③步跟第一个加入当前kset的kobject是一样的，即把要加入 的kobject的成员设置，使之指向当前的kset对应数据，而④需要把kobject添加到kset的list的尾部，下图表示了kobject b加入到kset A的图示：



**当有一个kset，需要加入到当前的kset,其方法也跟一个kobject要加入到当前kset一样，即把要加入的kset中所 包含的kobject的成员设置，使这些成员指向对应的kset的对应数据。而当前kset要加入另一个kset，其方式也是跟一个kset加入到当前 kset一样，都是设备kset中的kobject，使kobject的成员指向要加入的kset的对应数据即可，下图显示了一个kset B加入到kset A中的图示。**



