LinuxUSB驱动框架分析Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022_LinuxUSB_E9_A9_c103_644723.htm

初次接触与OS相关的设备驱动编写，感觉还挺有意思的，为了不至于忘掉看过的东西，笔记跟总结当然不可缺，更何况我决定为嵌入式卖命了。好，言归正传，我说一说这段时间的收获，跟大家分享一下Linux的驱动开发。但这次只先针对Linux的USB子系统作分析，因为周五研讨老板催货。当然，还会顺带提一下其他的驱动程序写法。事实上，Linux的设备驱动都遵循一个惯例表征驱动程序(用driver更贴切一些，应该称为驱动器比较好吧)的结构体，结构体里面应该包含了驱动程序所需要的所有资源。用OO的术语来说，就是这个驱动器对象所拥有的属性及成员。由于Linux的内核用c来编写，所以我们也按照这种结构化的思想来分析代码，但我还是希望从OO的角度来阐述这些细节。这个结构体的名字有驱动开发人员决定，比如说，鼠标可能有一个叫做mouse_dev的struct，键盘可能由一个keyboard_dev的struct(dev for device，我们做的只是设备驱动)。而这次我们来分析一下Linux内核源码中的一个usb-skeleton(就是usb驱动的骨架咯)，自然，他定义的设备结构体就叫做usb-skel：

```
struct usb_skel { struct usb_device * udev. /* the usb device for this device */ struct usb_interface * interface. /* the interface for this device */ struct semaphore limit_sem. /* limiting the number of writes in progress */ unsigned char * bulk_in_buffer. /* the buffer to receive data */ size_t bulk_in_size. /* the size of the receive buffer */ __u8 bulk_in_endpointAddr. /* the address of the bulk in endpoint
```

```
*/ __u8 bulk_out_endpointAddr. /* the address of the bulk out
endpoint */ struct kref kref. }. 这里我们得补充说明一下一些USB
的协议规范细节。USB能够自动监测设备，并调用相应得驱
动程序处理设备，所以其规范实际上是相当复杂的，幸好，
我们不必理会大部分细节问题，因为Linux已经提供相应的解
决方案。就我现在的理解来说，USB的驱动分为两块，一块
是USB的bus驱动，这个东西，Linux内核已经做好了，我们可
以不管，但我们至少要了解他的功能。形象得说，USB的bus
驱动相当于铺出一条路来，让所有的信息都可以通过这
条USB通道到达该到的地方，这部分工作由usb_core来完成。
当设备接到USB接口是，usb_core就检测该设备的一些信息，
例如生产厂商ID和产品的ID，或者是设备所属的class
、subclass跟protocol，以便确定应该调用哪一个驱动处理该设
备。里面复杂细节我们不用管，我们要做的是另一块工作usb
的设备驱动。也就是说，我们就等着usb_core告诉我们要工作
了，我们才工作。对于usb规范定义的设备，他们有一个设备
的框架，对于开发人员来说，他大概如图所示：从开发人员的
角度看，每一个usb设备有若干个配置(configuration)组成，
每个配置又可以有多个接口(interface)，每个接口又有多个设
置(setting图中没有给出)，而接口本身可能没有端点或者多个
端点(end point)。USB的数据交换通过端点来进行，主机与各
个端点之间建立起单向的管道来传输数据。而这些接口可以
分为四类：控制(control)用于配置设备、获取设备信息、发
送命令或者获取设备的状态报告 中断(interrupt) 当USB宿主要
求设备传输数据时，中断端点会以一个固定的速率传送少量
数据，还用于发送数据到USB设备以控制设备，一般不用于
```

传送大量数据。 批量(bulk) 用于大量数据的可靠传输，如果总线上的空间不足以发送整个批量包，它会被分割成多个包传输。 等时(isochronous) 大量数据的不可靠传输，不保证数据的到达，但保证恒定的数据流，多用于数据采集。 Linux中用struct usb_host_endpoint来描述USB端点，每个usb_host_endpoint中包含一个struct usb_endpoint_descriptor结构体，当中包含该端点的信息以及设备自定义的各种信息，这些信息包括： bEndpointAddress(b for byte) 8位端点地址，其地址还隐藏了端点方向的信息(之前说过，端点是单向的)，可以用掩码USB_DIR_OUT和USB_DIR_IN来确定。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com