

介绍Linux操作系统下的U 硬件设备驱动 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/253/2021_2022__E4_BB_8B_E7_BB_8DLinu_c103_253019.htm USB 是英文"Universal Serial Bus"的缩写，意为"通用串行总线"。是由Compaq(康柏)、DEC、IBM、Intel、NEC、微软以及Northern Telecom (北方电讯) 等公司于1994年11月共同提出的，主要目的就是为了解决接口标准太多的弊端。USB使用一个4针插头作为标准插头，并通过这个标准接头，采用菊花瓣形式把所有外设连接起来，它采用串行方式传输数据，目前最大数据传输率为12Mbps, 支持多数据流和多个设备并行操作，允许外设热插拔。目前USB接口虽然只发展了2代 (USB1.0/1.1, USB2.0)，但是USB综合了一个多平台标准的所有优点 -- 包括降低成本，增加兼容性，可连接大量的外部设备，融合先进的功能和品质。使其逐步成为PC接口标准，进入了高速发展期。那么对于使用Linux系统，正确支持和配置常见的USB设备，就是其使用必不可少的一步。相关技术基础 模块 (驱动程序) 模块 (module) 是在内核空间运行的程序，实际上是一种目标对象文件，没有链接，不能独立运行，但是可以装载到系统中作为内核的一部分运行，从而可以动态扩充内核的功能。模块最主要的用处就是用来实现设备驱动程序。Linux下对于一个硬件的驱动，可以有两种方式：直接加载到内核代码中，启动内核时就会驱动此硬件设备。另一种就是以模块方式，编译生成一个.o文件。当应用程序需要时再加载进内核空间运行。所以我们所说的一个硬件的驱动程序，通常指的就是一个驱动模块。设备文件 对于一个设备，它可以在/dev

下面存在一个对应的逻辑设备节点，这个节点以文件的形式存在，但它不是普通意义上的文件，它是设备文件，更确切的说，它是设备节点。这个节点是通过mknod命令建立的，其中指定了主设备号和次设备号。主设备号表明了某一类设备，一般对应着确定的驱动程序；次设备号一般是区分不同属性，例如不同的使用方法，不同的位置，不同的操作。这个设备号是从/proc/devices文件中获得的，所以一般是先有驱动程序在内核中，才有设备节点在目录中。这个设备号（特指主设备号）的主要作用，就是声明设备所使用的驱动程序。驱动程序和设备号是一一对应的，当你打开一个设备文件时，操作系统就已经知道这个设备所对应的驱动程序。SCSI设备 SCSI 是有别于IDE的一个计算机标准接口。现在大部分平板式扫描仪、CD-R刻录机、MO光磁盘机等渐渐趋向使用SCSI接口，加之SCSI又能提供一个高速传送通道，所以，接触到SCSI设备的用户会越来越多。Linux支持很多种的SCSI设备，例如：SCSI硬盘、SCSI光驱、SCSI磁带机。更重要的是，Linux提供了IDE设备对SCSI的模拟（ide-scsi.o模块），我们通常会就把IDE光驱模拟为SCSI光驱进行访问。因为在Linux中很多软件都只能操作SCSI光驱。例如大多数刻录软件、一些媒体播放软件。通常我们的USB存储设备，也模拟为SCSI硬盘而进行访问。Linux硬件驱动架构 对于一个硬件，Linux是这样来进行驱动的：首先，我们必须提供一个.o的驱动模块文件（这里我们只说明模块方式，其实内核方式是类似的）。我们要使用这个驱动程序，首先要加载运行它（insmod *.o）。这样驱动就会根据自己的类型（字符设备类型或块设备类型，例如鼠标就是字符设备而硬盘就是块设备

) 向系统注册，注册成功系统会反馈一个主设备号，这个主设备号就是系统对它的唯一标识（例如硬盘块设备在`/proc/devices`中显示的主设备号为3，我们用`ls -l /dev/had`看到的主设备就肯定是3）。驱动就是根据此主设备号来创建一个一般放置在`/dev`目录下的设备文件（`mknod`命令用来创建它，它必须用主设备号这个参数）。在我们要访问此硬件时，就可以对设备文件通过`open`、`read`、`write`等命令进行。而驱动就会接收到相应的`read`、`write`操作而根据自己的模块中的相应函数进行了。其中还有几个比较有关系的東西：一个是`/lib/modules/2.4.XX`目录，它下面就是针对当前内核版本的模块。只要你的模块依赖关系正确（可以通过`depmod`设置），你就可以通过`modprobe`命令加载而不需要知道具体模块文件位置。另一个是`/etc/modules.conf`文件，它定义了一些常用设备的别名。系统就可以在需要此设备支持时，正确寻找驱动模块。例如`alias eth0 e100`，就代表第一块网卡的驱动模块为`e100.o`。他们的关系图如下：配置USB设备 内核中配置. 要启用 Linux USB 支持，首先进入"USB support"节并启用"Support for USB"选项（对应模块为`usbcore.o`）。尽管这个步骤相当直观明了，但接下来的 Linux USB 设置步骤则会让人感到糊涂。特别地，现在需要选择用于系统的正确 USB 主控制器驱动程序。选项是"EHCI"（对应模块为`ehci-hcd.o`）、"UHCI"（对应模块为`usb-uhci.o`）、"UHCI (alternate driver)"和"OHCI"（对应模块为`usb-ohci.o`）。这是许多人对 Linux 的 USB 开始感到困惑的地方。要理解"EHCI"及其同类是什么，首先要知道每块支持插入 USB 设备的主板或 PCI 卡都需要有 USB 主控制器芯片组。这个特别的芯片组与插入系

统的 USB 设备进行相互操作，并负责处理允许 USB 设备与系统其它部分通信所必需的所有低层次细节。Linux USB 驱动程序有三种不同的 USB 主控制器选项是因为在主板和 PCI 卡上有三种不同类型的 USB 芯片。"EHCI"驱动程序设计成为实现新的高速 USB 2.0 协议的芯片提供支持。"OHCI"驱动程序用来为非 PC 系统上的（以及带有 SiS 和 ALi 芯片组的 PC 主板上的）USB 芯片提供支持。"UHCI"驱动程序用来为大多数其它 PC 主板（包括 Intel 和 Via）上的 USB 实现提供支持。只需选择与希望启用的 USB 支持的类型对应的"?HCI"驱动程序即可。如有疑问，为保险起见，可以启用"EHCI"、"UHCI"（两者中任选一种，它们之间没有明显的区别）和"OHCI"。（赵明注：根据文档，EHCI已经包含了UHCI和OHCI，但目前就我个人的测试，单独加EHCI是不行的，通常我的做法是根据主板类型加载UHCI或OHCI后，再加载EHCI这样才可以支持USB2.0设备）。启用了"USB support"和适当的"?HCI"USB 主控制器驱动程序后，使 USB 启动并运行只需再进行几个步骤。应该启用"Preliminary USB device filesystem"，然后确保启用所有特定于将与 Linux 一起使用的实际 USB 外围设备的驱动程序。例如，为了启用对 USB 游戏控制器的支持，我启用了"USB Human Interface Device (full HID) support"。我还启用了主"Input core support"节下的"Input core support"和"Joystick support"。一旦用新的已启用 USB 的内核重新引导后，若/proc/bus/usb下没有相应USB设备信息，应输入以下命令将 USB 设备文件系统手动挂装到 /proc/bus/usb：

```
# mount -t usbdevfs none /proc/bus/usb
```

为了在系统引导时自动挂装 USB 设备文件系统，请将下面一行添加到 /etc/fstab 中的 /proc 挂装

行之后：`none /proc/bus/usb usbdevfs defaults 0 0` 模块的配置方法。在很多时候，我们的USB设备驱动并不包含在内核中。其实我们只要根据它所需要使用的模块，逐一加载。就可以使它起作用。首先要确保在内核编译时以模块方式选择了相应支持。这样我们就应该可以在`/lib/modules/2.4.XX`目录看到相应.o文件。在加载模块时，我们只需要运行`modprobe xxx.o`就可以了（`modprobe`主要加载系统已经通过`depmod`登记过的模块，`insmod`一般是针对具体.o文件进行加载）对应USB设备下面一些模块是关键。 `usbcore.o`要支持usb所需要的最基础模块 `usb-uhci.o`（已经提过） `usb-ohci.o`（已经提过） `uhci.o`另一个uhci驱动程序，我也不知道有什么用，一般不要加载，会死机的 `ehci-hcd.o`（已经提过 `usb2.0`） `hid.o`USB人机界面设备，像鼠标呀、键盘呀都需要 `usb-storage.o`USB存储设备，U盘等用到 相关模块 `ide-disk.o`IDE硬盘 `ide-scsi.o`把IDE设备模拟SCSI接口 `scsi_mod.o`SCSI支持 注意kernel config其中一项：`Probe all LUNs on each SCSI device 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