

# Linux培训系列

## 第六讲

将向您演示如何从源代码编译 Linux 内核。在演示过程中，我们将讨论各种重要的内核配置选项，更深入地介绍有关内核中 PCI 和 USB 支持的信息。在本系列教程（共 8 篇；本文是第 6 篇）结束时，您将具备成为 Linux 系统管理员所必需的知识。

红联Linux论坛是致力于Linux技术讨论的站点，目前网站收录的文章及教程基本能满足不同水平的朋友学习。

红联Linux门户：[www.linux110.com](http://www.linux110.com)

红联Linux论坛：[www.linuxdiyf.com/bbs](http://www.linuxdiyf.com/bbs)

下载:Linux电子书籍：

<http://www.linux286.com/linux/linuxdzsj.htm>

## 目录

### 介绍内核

- 介绍内核
- 与硬件进行相互操作
- CPU 抽象
- 抽象 IO
- 以网络为中心
- 联网的优点
- 引导回顾
- 介绍 ... 模块！
- 模块所在位置
- 模块 — 并非用于所有进程！

### 查找并下载源代码

- 查找并下载源代码 - 内核版本历史
- 使用哪些内核源代码
- 从源代码获得内核
- 解包内核

### 配置内核

- 配置内核
- 新的配置方法
- 配置技巧
- 代码成熟级别选项
- 模块以及与 CPU 相关的选项
- 常规和并行端口选项
- RAID 和 LVM
- 联网和相关设备
- IDE 支持
- SCSI 支持
- 各种字符设备
- 文件系统和控制台驱动程序

### 编译和安装内核

- 编译和安装内核 - make dep
- make bzImage
- 编译模块

### 引导配置

- 引导配置 - LILO 简介
- 配置 LILO
- LILO 代码
- 调整 LILO
- 最终的 lilo.conf
- LILO 配置的原因

### PCI 设备

- PCI 设备
- 检查 PCI 设备
- PCI 设备资源

### Linux USB

- Linux USB
- 启用 USB
- UHCI、OHCI 和 EHCI — 天啊！

*最后几步*

*挂装 usbdevfs*

**海量Linux文章**

*海量Linux技术文章*

介绍内核

## 介绍内核

发布时间 :2007-01-27 21:43:29

通常，“Linux”一词指的是完整的Linux分发版（distribution）和使分发版工作的所有协作运行程序。然而，您或许会惊奇地发现从技术角度讲Linux是并且只是一个内核。尽管一般所指的“Linux”的其它部分（如shell和编译器）是完整的操作环境的基本组成部分，但从技术角度讲，这些部分与Linux（内核）是分开的。尽管如此，人们仍用“Linux”来指“基于Linux的分发版”。不过，至少每个人都认同Linux内核是所有“Linux操作系统”的心脏。

## 与硬件进行相互操作

发布时间 :2007-01-27 21:43:57

Linux 内核的主要作用是直接与系统中的硬件进行相互操作。内核在原始硬件与应用程序之间提供了一个抽象层。例如，按此方式，程序本身无需知道特定主板芯片组或磁盘控制器的详细信息 — 而是可以在向磁盘读写文件的更高层次上进行操作。

## CPU 抽象

发布时间 :2007-01-27 21:44:27

Linux 内核还在系统的处理器（或多处理器）之上提供一个抽象层次 — 使多个程序看起来象是同时运行。内核负责使每个进程公平、分时共享处理器的计算资源。

若正在运行 Linux，那么正在使用的内核不是支持 UP（单处理器）的，就是支持 SMP（对称多处理器）的。如果您正好有一块 SMP 主板，但使用的是 UP 内核，那么 Linux 将不会“看见”额外的处理器！修正这一问题需要编译用于您硬件的特殊 SMP 内核。目前，SMP 内核也可运行在单处理器系统上，但性能略有下降。

## 抽象 IO

发布时间 :2007-01-27 21:44:55

内核还处理大量需要的任务：抽象所有文件输入输出格式。设想一下，如果每个程序都必须直接与特定的磁盘硬件打交道，会发生什么 — 如果您更换磁盘控制器，所有的程序都会停止运行！幸运的是，Linux 内核遵循 UNIX 模型：提供所有程序都可以使用的简单数据存储和访问抽象。这样的话，您喜欢的数据库就不需要考虑数据是存储在 IDE 硬盘上、还是存储在 SCSI RAID 阵列上或是存储在挂装于网络的文件系统上。

## 以网络为中心

发布时间 :2007-01-27 21:45:25

Linux 赢得声誉的主要因素之一是其健壮的联网功能，尤其是 TCP/IP 支持。如果您猜想 TCP/IP 栈在 Linux 内核中，那么您猜对了！内核为在网络上发送数据的程序提供了符合标准的高层次接口。在幕后，Linux 内核直接与特定的以太网卡或 pppd 守护程序进行相互操作，并处理低层次的因特网通信详细信息。请注意：本系列的下一篇文章（第 7 部分）将讨论 TCP/IP 和联网。



## 联网的优点

发布时间 :2007-01-27 21:45:55

Linux 的优点之一是内核中有大量可用的可选特性，特别是与联网有关的特性。例如，可以将内核配置为允许整个内部网络经过 Linux 调制解调器访问因特网 — 这称为 IP 伪装 (IP Masquerading) 或 IP NAT。

此外，可以将 Linux 内核配置为导出或挂装基于网络的 NFS 文件系统，这是考虑到允许 LAN 上其它 UNIX 机器可以轻松地与 Linux 系统共享数据。内核中有许多好东西，一旦开始研究 Linux 内核的众多配置选项，您就会了解。

## 引导回顾

发布时间 :2007-01-27 21:46:24

现在是快速复习 Linux 引导过程的良好时机。启动基于 Linux 的系统时，内核映象（以单个二进制文件形式存储）由引导装入程序（如 LILO 或 GRUB）从磁盘装入内存。此时，内核接管系统。内核首先做的事情之一是检测并初始化所有它找到且配置成支持的硬件。硬件正确初始化之后，内核就准备启动常规用户空间程序（也称为“进程”）。

由内核运行的第一个进程是 /sbin/init。它依次启动 /etc/inittab 中指定的其它进程。几秒钟内，Linux 系统就启动并运行起来，准备好供您使用。虽然您从不直接与内核打交道，但 Linux 内核始终运行于所有常规进程“之上”，并提供各种程序和库运行所必需的虚拟与抽象。

## 介绍 ... 模块！

发布时间 :2007-01-27 21:46:52

所有较新的 Linux 内核都支持内核模块。内核模块是真正美妙的事物 — 它们是内核的组成部分，驻留在磁盘上相对较小的二进制文件中。每当内核需要一个特定模块的功能时，就会从磁盘装入这个特定模块并自动将它与自身结合，这样便动态扩展了内核的能力。

如果装入的内核模块的特性在几分钟内未被使用，则内核会自动将它与内核其余部分分离并从内存卸装它 — 这被称为自动清除 ( autocleaning )。若没有内核模块，则需要确保运行的内核 ( 作为单个二进制文件存在于磁盘上 ) 完全包含所有可能需要的功能。若没有模块，则需要构建一个全新内核，以将新的重要功能添加到其中。

通常，用户构建一个包含所有基本功能的内核映象，然后再构建一组模块，这些模块对应着用户未来可能需要的功能。以后要使用时，则按照需要，将适当的模块装入内核。这也有助于节约 RAM，因为模块只有从磁盘装入后才使用 RAM。当从内核除去模块后，其使用的内存就被释放并用于其它用途。

## 模块所在位置

发布时间 :2007-01-27 21:47:30

内核模块通常位于 /lib/modules/x.y.z (其中 x.y.z 是模块所兼容的内核版本) ; 每个模块在其名称末尾都有 “.o” , 表明它是包含机器指令的二进制文件。正如您猜想的那样, 每个单独的模块代表内核功能的一个特殊组。D 擊瞿? 榭贍芴崙 FAT 文件系统支持, 而另一个模块则可能支持一块特殊的 ISA 以太网卡。

## 模块 — 并非用于所有进程！

发布时间 :2007-01-27 21:48:02

值得一提的是：不能将所有东西都放入模块中。因为模块存储于磁盘上，可引导的内核映象需要拥有对磁盘控制器、驱动器和根文件系统的内编译（compiled-in）支持。如果没有将这些基本组件编译到内核映象中 — 也就是说，如果试图将它们作为模块编译 — 那么内核将没有从磁盘装入这些模块所必需的能力，这会产生一个很令人讨厌的“先有鸡还是先有蛋”的问题，结果是一个不能引导系统的内核！

查找并下载源代码

## 查找并下载源代码 - 内核版本历史

发布时间 :2007-01-27 21:49:45

目前最新的内核是2.6.20。也是较稳定的2.6内核系列。

可用的内核是 2.4.18。2.4.18 内核是 2.4 稳定内核系列的一部分。这一系列的内核发行版打算用于生产系统。

还有几个 2.5 系列内核可供使用，但不应当在生产系统上使用它们。“2.5”中的“5”是奇数，表示这些内核本质上是实验性的，并且打算供内核开发人员使用。当“2.5”内核准备好用于生产使用时，就会出现“2.6”（第二个数为偶数）系列。

## 使用哪些内核源代码

发布时间 :2007-01-27 21:50:13

若只想编译当前已安装内核的新版本（例如，要启用 SMP 支持），则最好的方法是安装分发版的内核源代码包。这样做之后，您应该在 `/usr/src/linux` 中发现一组新文件。

然而，有时候您想安装新内核。通常，最好的方法是只安装分发版内核源代码包的新版本或更新版本。该包将包含内核源代码，这些内核源代码经过补丁程序的修正，并且经过调整以便能最优地运行在 Linux 系统上。

## 从源代码获得内核

发布时间 :2007-01-27 21:50:44

如果您具有冒险精神，则可以从 <http://www.kernel.org/pub/linux/kernel> 抓取一个“主线”内核源代码 tar 包。在这个目录中，您可以找到由 Linus 或 Marcelo 发布的正式内核源代码。这些源代码可能没有分发版内核源代码包中的所有特性，因此，在知道自己正在做什么以前 ... 或在有“额外的”机器和大量空闲时间以前，通常最好不要使用主线内核:)

在 kernel.org，您会发现内核源代码根据内核版本（v2.2、v2.4 等）被组织到几个不同的目录中。在每个目录中，您会发现文件被标为“linux-x.y.z.tar.gz”和“linux-x.y.z.tar.bz2”。这些文件是 Linux 内核源代码 tar 包。您还会看到标为“patch-x.y.z.gz”和“patch-x.y.z.bz2”的文件。这些文件是补丁程序，可以用来更新完整内核源代码的较早版本。如果要编译新的内核发行版，则需下载这些“linux”文件之一。



## 解包内核

发布时间 :2007-01-27 21:51:16

如果您从 kernel.org 下载了新内核，则现在可以将它解包。要这么做，用 cd 命令进入 /usr/src。如果在那里有一个现成的“linux”目录，则将它改成“linux.old”（以 root 用户身份执行 mv linux linux.old）。

现在，可以解压缩新内核了。若仍处在 /usr/src 中，则输入 tar xzvf /path/to/my/kernel-x.y.z.tar.gz 或 cat /path/to/my/kernel-x.y.z.tar.bz2 | bzip2 -d | tar xvf -，这取决于源代码是用 gzip 还是用 bzip2 压缩的。输入这一命令后，新的内核源代码将被解压缩到新的“linux”目录中。注意 — 完整的内核源代码通常占据 50 MB 以上的磁盘空间！

配置内核

## 配置内核

发布时间 :2007-01-27 21:51:50

编译内核以前，需要配置它。可以用配置来精确控制在新内核中启用（以及禁用）哪些内核特性。还可以控制将哪些部分编译到内核二进制映像（将在引导时装入）中，而将哪些部分编译到“按需装入”的内核模块文件中。

老式的配置内核的方法是非常令人痛苦的，包括进入 `/usr/src/linux` 并输入 `make config`。尽管 `make config` 仍然有效，但请不要尝试用这种方法配置内核 — 除非您喜欢在命令行回答数百个（是的，数百个！）yes/no 问题。

## 新的配置方法

发布时间 :2007-01-27 21:52:23

不用再输入“make config”，新的方法是输入“make menuconfig”或“make xconfig”来配置内核。如果输入“make menuconfig”，则会得到一个好看的基于控制台的彩色菜单系统，可以用它配置内核。如果输入“make xconfig”，则会得到一个非常好看的基于 X 的 GUI，可以用它配置各种内核选项。

使用“make menuconfig”时，左边有“<>”的选项可以被编译为模块。当选项被突出显示时，按空格键在取消该选项（“<>”）、选择将该选项编译到内核映像中（“<\*”）和选择编译为模块（“<M>”）三者间切换。也可以按“y”键启用选项，按“n”禁用该选项或按“m”键以在可能的情况下将它编译为模块。幸运的是，大多数内核配置选项都有详细的帮助信息，可以通过输入 h 查看这些信息。

## 配置技巧

发布时间 :2007-01-27 21:52:54

遗憾的是，内核配置选项太多了，根本没有足够篇幅在这里全部介绍它们（不过，如果有兴趣，可以查看options(4)手册页以获取更完整的选项清单）。

在下列几页中，将对执行“make menuconfig”或“make xconfig”命令时出现的重要类别进行概述，并同时指出基本的或重要的内核配置选项。

## 代码成熟级别选项

发布时间 :2007-01-27 21:53:22

现在，我们来看看不同的内核配置选项类别。下面简略概述了每个类别。我鼓励您通过在 `/usr/src/linux` 中输入 “`make menuconfig`” 或 “`make xconfig`” 来加深对这些类别的理解。

Code maturity level options：该配置类别包含单一选项：“Prompt for development and/or incomplete code/drivers”。如果启用这个选项，那么许多被认为还处在实验阶段的选项（如 ReiserFS、devfs 和其它选项）将在其它类别菜单下可见。如果不选中这个选项，可见的选项将只是那些被认为是“稳定的”选项。通常，启用这个选项是个好主意，这样可以看到内核必须提供的所有功能。

## 模块以及与 CPU 相关的选项

发布时间 :2007-01-27 21:53:51

Loadable module support : 该配置类别下是三个与内核对模块的支持相关的选项。通常，应该启用所有这三项。

Processor type and features : 这一部分包含各种特定于 CPU 的配置选项。“ Symmetric multiprocessing support option ” 特别重要，如果系统有一个以上的 CPU，则应启用该选项。否则，只能利用系统中的第一个 CPU。通常应该启用“ MTRR Support ” 选项，因为它可以在现代系统上的 X 中产生更好的性能。

## 常规和并行端口选项

发布时间 :2007-01-27 21:54:18

General setup：在这一节中，通常应启用联网和 PCI 支持选项，还应启用“Kernel support for ELF binaries”（将它构建到内核中，而不是构建为模块）。建议启用 a.out 和 MISC 二进制选项，不过将它们构建为内核模块通常更有意义。还要确保启用“System V IPC”和“Sysctl support”。请参阅内置帮助，以获取有关这些选项的更多信息。

Parallel port support 选项：拥有并行端口设备（包括打印机）的人都会对 Parallel port support 节感兴趣。请注意：为了获得完整的打印机支持，除了启用此处适当的并行端口支持以外，还必须启用“Character devices”节下的“Parallel printer support”。

## RAID 和 LVM

发布时间 :2007-01-27 21:54:46

Multi-device support (RAID and LVM) : 这包含与 Linux 软件 RAID 和逻辑卷管理有关的选项。软件 RAID 允许以冗余方式使用磁盘以提高可用性。您可以在 developerWorks 软件 RAID 系列中找到有关软件 RAID 的更多信息（请参阅本教程最后一节“参考资料”，获取相关链接）。



## 联网和相关设备

发布时间 :2007-01-27 21:55:13

Networking options : 您当然会猜到这包含与联网有关的选项！如果计划将 Linux 系统连接至一个典型网络，则应确保启用“Packet socket”、“Unix domain sockets”和“TCP/IP networking”。您可能会对各种其它选项感兴趣，其中包括“Network packet filtering”，它允许您使用 iptables 命令设置自己的有状态防火墙。有关这一操作的信息，请参阅 developerWorks 教程 Linux 2.4 stateful firewall design。

Network device support : 使 Linux 联网正常工作的第二个要求是将对特别的联网硬件的支持编译进来。应选择对您希望内核支持的网卡（或多块网卡）的支持。需要的选项最有可能位于“Ethernet (10 or 100Mbit)”子类别下。

## IDE 支持

发布时间 :2007-01-27 21:55:43

ATA/IDE/MFM/RLL support : 对使用 IDE 驱动器、CD-ROM、DVD-ROM 和其它外围设备的系统而言，这一节包含重要的选项。如果系统有 IDE 磁盘，则请确保启用“Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support”、“Include IDE/ATA-2 DISK support”以及适合特殊主板的芯片组（构建到内核中，而不是构建为模块——这样系统才能引导！）。如果有 IDE CD-ROM，则请确保还启用了“Include IDE/ATAPI CD-ROM support”。注：若没有特定芯片组支持，IDE 仍可工作，但可能无法利用特别的主板的所有性能增强功能。

还请注意：对几乎所有的系统，都极力推荐“Enable PCI DMA by default if available”选项。若没有缺省地启用 DMA（直接内存存取），则 IDE 外围设备将以 PIO 方式运行，并且性能可能比正常情况下慢 15 倍！通过以 root 用户身份在 shell 提示符上输入 `hdparm -d 1 /dev/hdx`，可以验证是否在特别的磁盘上启用了 DMA，其中 `/dev/hdx` 是与要启用 DMA 的磁盘对应的块操作特殊设备。

## SCSI 支持

发布时间 :2007-01-27 21:56:12

SCSI support：这一节包含与 SCSI 磁盘和外围设备有关的所有选项。如果有基于 SCSI 的系统，请务必根据需要启用“SCSI support”、“SCSI disk support”、“SCSI CD-ROM support”和“SCSI tape support”。如果正从 SCSI 磁盘引导，则请确保将“SCSI support”和“SCSI disk support”都内编译到内核中，而不是选择将它们编译成可装入的模块。为使 SCSI 正确地工作，还需要执行额外的步骤：进入“SCSI low-level drivers”子类别，确保启用了特别的 SCSI 卡的支持并将其配置为直接编译到内核中。

## 各种字符设备

发布时间 :2007-01-27 21:56:38

Character devices：这一节包含各种内核驱动程序的大杂烩。请确保启用“Virtual terminal”和“Support for console on virtual terminal”，内核引导后出现的标准的基于文本的控制台需要它们。很有可能还需要启用“Unix98 PTY support”。如果您希望使用并行打印机，请记住还要启用“Parallel printer support”。其它所有选项通常是可选的。推荐采用“Enhanced real-time clock support”；通常需要“/dev/agpgart (AGP support)”和“Direct Rendering Manager”以利用 X 下丰富的 Linux 3D 加速功能（特别当您有 Voodoo3+、ATI Rage 128、ATI Radeon 或 Matrox 等显卡时）。要想使 X 在加速方式中工作，除了简单地启用这些选项外还需要额外的配置步骤。

## 文件系统和控制台驱动程序

发布时间 :2007-01-27 21:57:08

File systems：正象您猜想的那样，这包含与文件系统驱动程序相关的选项。要确保将用于“/”（根目录）的文件系统编译到内核中。这一文件系统通常是 ext2，但也可能是 ext3、JFS、XFS 或 ReiserFS。请确保还启用了“/proc file system support”选项，因为大多数分发版都需要它。除非计划使用“/dev file system support”（在这种情况下则应将“/dev/pts”选项置为禁用），否则，通常还应启用“/dev/pts file system support for Unix98 PTYs”。

Console drivers：大多数人通常会启用“VGA text console”（x86 系统上通常需要）以及可选的“Video mode selection support”。也有可能使用“Frame-buffer support”，它将使文本控制台以图形显示，而不再是文本屏幕。这些驱动程序中的一些会对 X 有负面影响，因此最好是坚持使用 VGA 文本控制台，至少在开始阶段是如此。

编译和安装内核

## 编译和安装内核 - make dep

发布时间 :2007-01-27 21:57:53

一旦配置了内核，就到了编译它的时候了。但能够编译它之前，需要生成相关性信息。在 /usr/src/linux 中可通过输入 make dep 做到这一点。

## make bzImage

发布时间 :2007-01-27 21:58:19

现在是编译实际二进制内核映象的时候了。输入 `make bzImage`。几分钟后，将完成编译，并且您将在 `/usr/src/linux/arch/i386/boot`（用于 x86 PC 内核）中找到 `bzImage` 文件。很快您将看到如何安装新内核映象，不过现在该讨论模块了。

## 编译模块

发布时间 :2007-01-27 21:58:48

既然已生成了 bzImage，那么是编译模块的时候了。即使在配置内核时没有启用任何模块，也不要省略这一步——养成编译 bzImage 后立即编译模块的习惯很有益处。而且，如果确实无模块启用编译——这个步骤会一闪而过。输入 `make modules && make modules_install`。这将编译模块并随后将其安装至 `/usr/lib/<kernelversion>` 中。

祝贺您！内核现已完全编译，而且模块全都被编译并安装。现在是重新配置 LILO 的时候了，这样您就可以引导新内核。



引导配置

## 引导配置 - LILO 简介

发布时间 :2007-01-27 21:59:30

终于到了重新配置 LILO 的时候了，这样它就可以装入新内核。LILO 是最流行的 Linux 引导装入程序，而且所有流行的 Linux 分发版都使用它。首先要做的是查看 `/etc/lilo.conf` 文件。该文件包含类似于 “`image=vmlinuz`” 的一行内容。这一行告诉 LILO 在哪里寻找内核。

## 配置 LILO

发布时间 :2007-01-27 22:00:04

要配置 LILO 以引导新内核，有两种选择。第一种是覆盖现有的内核 — 如果没有某种紧急引导方法（如带有这一特别的内核的引导磁盘），这样做就有些冒险。

较安全的选择是配置 LILO，以便它既可以引导新内核也可以引导旧内核。LILO 可以配置为在缺省情况下引导新内核，而当遇到问题时仍可选择较旧的内核。这是推荐的选择，我们会向您演示如何执行。

## LILO 代码

发布时间 :2007-01-27 22:00:36

lilo.conf 可能看起来如下：

```
boot=/dev/hda
delay=20
vga=normal
root=/dev/hda1
read-only
```

```
image=/vmlinuz
label=linux
```

要向 lilo.conf 添加新的引导项，请执行以下步骤。首先，将 /usr/src/linux/arch/i386/boot/bzImage 复制到 root 分区上的一个文件，如 /vmlinuz2。复制该文件后，再复制 lilo.conf 的最后三行内容，并将它们添加到该文件的末尾 ... 我们几乎完成了 ...

## 调整 LILO

发布时间 :2007-01-27 22:01:09

现在，lilo.conf 看起来如下：

```
boot=/dev/hda
delay=20
vga=normal
root=/dev/hda1
read-only
```

```
image=/vmlinuz
label=linux
```

```
image=/vmlinuz
label=linux
```

现在，将第一个“image=”行改为读作image=/vmlinuz2。接下来，将第二个“label”行改为读作label=oldlinux。此外，请确保在靠近文件顶部有一行“delay=20”——若没有，则添加一行。如果有，则要确保数字至少是20。

## 最终的 lilo.conf

发布时间 :2007-01-27 22:01:39

最终的 lilo.conf 文件看起来如下：

```
boot=/dev/hda
delay=20
vga=normal
root=/dev/hda1
read-only

image=/vmlinuz2
label=linux

image=/vmlinuz
label=oldlinux
```

完成上述操作后，需要以 root 身份运行“lilo”。这非常重要！如果不这么做，引导过程将无法进行。运行“lilo”可以给它更新引导映射的机会。

## LILO 配置的原因

发布时间 :2007-01-27 22:02:05

现在对所做的更改加以说明。我们对 lilo.conf 文件进行了设置以允许引导两个不同的内核。该文件允许引导位于 /vmlinuz 的原始内核。它也允许引导位于 /vmlinuz2 的新内核。缺省情况下，它会尝试引导新内核（因为新内核的 image/label 行在配置文件中首先出现）。

如果出于某种原因需要引导旧内核，只需重新引导计算机并按住 shift 键。LILO 会检测到这一操作，并且允许输入希望引导的映象的标号。要引导旧内核，需要输入 oldlinux，然后按 Enter 键。要查看可能存在的标号的列表，需要按 TAB 键。

PCI 设备

## PCI 设备

发布时间 :2007-01-27 22:02:40

这一节将较仔细地研究在 Linux 下处理 PCI 设备的更详细的信息。在 Linux 下启用对 PCI 设备的支持非常简单。只需确保在“General Setup”内核配置类别下启用“PCI support”。还推荐采用“PCI device name database”选项，因为它允许查看 Linux 可以看到的 PCI 设备的实际英文名（而不只是它们的正式 PCI 设备标识号码）。确保启用以上选项后，Linux 就做好了支持 PCI 的准备。

唯一的额外步骤是：针对安装在系统中的卡的类型，启用特定的驱动程序。例如，若要安装 SoundBlaster Live! 卡，则需启用“Creative SBLive!”支持（位于“Sound”类别下），而要安装 3Com 3c905c Fast Ethernet 卡的话，则需启用“Network device support/Ethernet (10 or 100Mbit)”类别 / 子类别下的“3c590/3c900 series (592/595/597) "Vortex/Boomerang" support”。

## 检查 PCI 设备

发布时间 :2007-01-27 22:03:12

要查看有关已安装的 PCI 设备的信息，可以输入 `cat /proc/pci` 以查看精简的（且略带神秘的）信息 — 或输入 `lspci -v` 获取更详细和更易于理解的输出信息。“`lspci`”是 `pciutils` 包的一部分，它的源代码可以从 <http://atrey.karlin.mff.cuni.cz/~mj/pciutils.html> 获得。通常，使用与特别的分发版一起提供的 `pciutils` 版本就足够了。当输入 `lspci -v` 时，可能会看到许多以前从不知道其存在的 PCI 设备。通常，这样的设备是众多内置于计算机主板的、基于 PCI 的外围设备之一。可以在计算机 BIOS 中禁用（如果设备当前为不可见，则也可启用）这些设备，通常在“Integrated peripherals”节下进行设置。通常可以在系统引导期间按 `Delete` 键或 `F2` 键来访问计算机的 BIOS。

`pciutils` 包还包含名为“`setpci`”的程序，可以用这个程序更改各种 PCI 设备设置（包括 PCI 设备等待时间）。要了解有关 PCI 设备等待时间以及它对系统所产生的影响的更多信息，请参阅 `developerWorks` 文章 `Linux hardware stability guide, Part 2`。



## PCI 设备资源

发布时间 :2007-01-27 22:03:46

为了能正常工作，系统中的 PCI 设备需要利用系统的各种特定硬件资源，如中断等。在有数据等待处理时，许多 PCI 设备利用硬件中断向处理器发送信号。要了解哪些中断正在被各种硬件设备使用，可以通过输入 `cat /proc/interrupts` 查看 `/proc/interrupts` 文件。该命令的输出看起来如下：

```
      CPU0
0: 3493317      XT-PIC timer
1:  86405      XT-PIC keyboard
2:    0      XT-PIC cascade
5:    0      XT-PIC eth0
8:    2      XT-PIC rtc
9:  62653      XT-PIC usb-uhci, usb-uhci, eth1
10: 1550399      XT-PIC Audigy
12: 413422      XT-PIC PS/2 Mouse
14:  85418      XT-PIC ide0
15:    4      XT-PIC ide1
NMI:    0
ERR:    0
```

第一列显示的是 IRQ 号码；第二列显示的是内核对于这一特别的 IRQ 处理了多少中断；最后一列则显示与这个 IRQ 相关联的硬件设备（或多个设备）的“短名”。如您所见，如有必要，多个设备能够共享同一个 IRQ。

还可以通过输入 `cat /proc/ioports` 查看硬件设备正在使用的 IO 端口。

Linux USB

## Linux USB

发布时间 :2007-01-27 22:04:19

配置内核时，您可能注意到“USB support”节，这一节包含属于USB（也称为通用串行总线）的选项。USB是将外围设备连接到PC的相对较新的方法。目前，有USB鼠标、键盘、游戏控制器、打印机、调制解调器和更多其它设备。因为Linux USB的支持相对较新，许多Linux用户从未在其Linux系统上使用过USB设备，或可能还不完全熟悉Linux USB支持是如何工作的。以下几页将迅速地介绍Linux USB以帮助您入门。

## 启用 USB

发布时间 :2007-01-27 22:04:47

要启用 Linux USB 支持，首先进入“USB support”节并启用“Support for USB”选项。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。您的选项是“EHCI”、“UHCI”、“UHCI (alternate driver)”和“OHCI”。这是许多人对 Linux 的 USB 开始感到困惑的地方。

## UHCI、OHCI 和 EHCI — 天啊！

发布时间 :2007-01-27 22:05:19

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”。

## 最后几步

发布时间 :2007-01-27 22:05:49

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”。

## 挂装 usbdevfs

发布时间 :2007-01-27 22:06:21

一旦用新的已启用 USB 的内核重新引导后，应输入以下命令将 USB 设备文件系统挂装到 /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
```

有关 USB 的更多信息，请访问随后在“参考资料”中列出的 USB 站点。

海量Linux文章

## 海量 Linux 技术文章

发布时间 :2006-11-24 16:50:29

下面是linux技术文章快速入口。需要联网：

[Linux 技术交流](#)

<http://www.linuxdiyf.com/bbs/forum-3-1.html>

[Linux 应用](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=1>

[Linux 安装及学习指导](#)

<http://www.linuxdiyf.com/bbs/forum-45-1.html>

[Linux 系统安装](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=45&filter=type&typeid=11>

[Linux 学习指导](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=45&filter=type&typeid=12>

[Linux 软件安装](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=45&filter=type&typeid=13>

[shell](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=3>

[Linux 壁纸](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=4>

[红旗](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=5>

[Redhat](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=6>

[SuSE](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=7>

## Linux 认证

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=9>

## Linux 下载分享(酷件、书籍、视频分享)

<http://www.linuxdiyf.com/bbs/forum-6-1.html>

## 服务器应用

<http://www.linuxdiyf.com/bbs/forum-7-1.html>

## 数据库应用

<http://www.linuxdiyf.com/bbs/forum-8-1.html>

## Linux 编程与内核

<http://www.linuxdiyf.com/bbs/forum-9-1.html>

## UniX 技术文章

<http://www.linuxdiyf.com/bbs/forum-32-1.html>

## Linux 业界声音、新闻

<http://www.linuxdiyf.com/bbs/forum-11-1.html>

## Linux 人才招聘信息

<http://www.linuxdiyf.com/bbs/forum-46-1.html>



网络转载，感谢原作者！

制作：红联Linux论坛

祝您阅读愉快！