

Debian GNU/Linux 5.0 ("lenny"), ARM 的发行说明

Debian 文档项目 [<http://www.debian.org/doc/>]

Debian GNU/Linux 5.0 ("lenny"), ARM 的发行说明: Debian 文档项目 [<http://www.debian.org/doc/>]

出版日期 2009-02-14

本文档是自由软件；您可以在自由软件基金会发布的 GNU 公共许可证的条款下重新发布或修改它；您应当使用该许可证的第二版本。

本程序发布的目的是希望它对您有用，但没有任何担保，即使针对商业或其它特定应用目的。请查阅 GNU 公共许可证以获得更多细节。

您应当在收到本程序的同时也收到了一份 GNU 公共许可证的副本；如果没有收到，请给自由软件基金会写信。地址是：51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

协议文本可以在 <http://www.gnu.org/copyleft/gpl.html> 和 Debian GNU/Linux /usr/share/common-licenses/GPL-2 找到。

目录

- 1. 简介 xx
 - 报告本文档的 bug xx
 - 贡献升级报告 xx
 - 本文档的源文件 xx
- 2. Debian GNU/Linux 5.0 有什么新东西 xx
 - ARM 架构中有什么新东西? xx
 - 这次发布中有什么新东西? xx
 - 包管理 xx
 - proposed-updates 区 xx
 - 系统改进 xx
 - 内核相关的主要变化 xx
 - 内核打包的变化 xx
 - Emdebian 1.0(基于 Debian GNU/Linux lenny 5.0) xx
 - 上网本支持 xx
 - Debian 中的 Java xx
- 3. 安装系统 xx
 - 安装系统有哪些新特性? xx
 - 主要更新 xx
 - 自动安装 xx
- 4. 从以前的发行版升级 xx
 - 升级前的准备 xx
 - 备份数据和配置文件 xx
 - 提前告知用户 xx
 - 准备恢复 xx
 - 为升级准备安全环境 xx
 - 检查系统状态 xx
 - 复审包管理器中的未决操作 xx
 - 禁用 APT pinning 操作 xx
 - 检查包状态 xx
 - proposed-updates 区 xx
 - 非官方源和 backports xx
 - 手工删除软件包的标记 xx
 - 为 APT 准备源 xx
 - 添加互联网 APT 源 xx
 - 添加本地镜像 APT 源 xx
 - 从 CD-ROM 或 DVD 添加 APT 源 xx
 - 升级软件包 xx
 - 记录会话 xx
 - 更新包列表 xx
 - 确保您有足够的空间升级 xx
 - 先升级 apt 和/或 aptitude xx
 - 使用 aptitudes 记录的 apt 自动安装的软件包列表 xx
 - 最小系统升级 xx
 - 升级系统的其它部分 xx
 - 升级期间可能遇到的问题 xx
 - 升级内核与相关包 xx
 - 安装内核元数据包 xx
 - 设备枚举的顺序 xx
 - 引导期间的问题 xx
 - 在重启之前要做的事情 xx
 - 重新运行 lilo xx

- 系统引导时在显示 **Waiting for root file system** 后挂起 xx
 - 如何在升级前避免此问题 xx
 - 如何在升级后解决此问题 xx
- 为下个发行版做准备 xx
- 过时的包 xx
 - 哑包 xx
- 下个 **Debian** 发行版的计划 xx
 - 抛弃 **ARM ABI**, 而改用 **ARM EABI** xx
- 5. **lenny** 中需要注意的问题 xx
 - 潜在的问题 xx
 - 与 **udev** 相关的设备的问题 xx
 - 某些应用程序可能无法在 2.4 内核下工作 xx
 - 某些网站无法通过 **TCP** 进行连接 xx
 - 无法自动关机 xx
 - 异步网络初始化可能引起无法预测的情况 xx
 - 使用 **WPA** 保护无线网络的问题 xx
 - 文件名中使用非 **ASCII** 字符可能引起问题 xx
 - 声音停止工作 xx
 - NFS** 挂载现在由 **nfs-common** 来处理 xx
 - 罗马尼亚(ro)键盘布局的改变 xx
 - 升级 **apache2** xx
 - NIS** 和网络管理器 xx
 - Mozilla** 产品的安全状态 xx
 - Security status of OCS Inventory and SQL-Ledger** xx
 - KDE** 桌面 xx
 - GNOME** 桌面改变与支持 xx
 - Emacs21*** 默认不支持 **Unicode** xx
 - slurpd/replica** 已经被删除 xx
 - 桌面没有使用全部屏幕 xx
 - DHCP** 失败备选问题 xx
 - VServer Disk Limit** xx
- 6. 关于 **Debian GNU/Linux** 的更多信息 xx
 - 更多读物 xx
 - 获得帮助 xx
 - 邮件列表 xx
 - 互联网在线聊天系统 xx
 - 报告 **Bug** xx
 - 为 **Debian** 做贡献 xx
- A. 管理您的 **etch** 系统 xx
 - 升级您的 **etch** 系统 xx
 - 检查您的 **sources list** xx
- B. 发行注记的贡献者 xx
- C. **Lenny** 献给 **Thiemo Seufer** xx
- 索引 xx
- 术语表 xx

第 1 章 简介

本文档告知 Debian GNU/Linux 发行版的用户，关于版本 5.0 (代号 "lenny") 的主要变化。

本发行说明提供如何安全的从版本 4.0 (代号 **etch**) 升级到当前版本的信息，并告知用户在升级到当前版本时可能会遇到已知的某些潜在问题。

在 <http://www.debian.org/releases/lenny/releasenotes> 可以获得本文档的最新版本。如有疑问，请检查首页的日期，以确定您是否正在阅读最新版本。

小心

注意列出所有已知的问题是不可能的，因此选择了那些普遍存在和造成影响的问题。

请注意我们只支持从 Debian 的前一版本 (现在要升级的版本是 4.0) 升级。如果您需要从更早的版本升级，建议您先查看前一版本的发行说明，先升级到 4.0。

报告本文档的 bug

我们已经试图测试在本文档中描述的所有不同的升级步骤并预测用户可能遇到的所有问题。

虽然如此，如果您认为您发现了 **Bug** (不正确或者缺失的信息)，请在 **Bug** 追踪系统 [<http://bugs.debian.org/>]提交一个关于 `release-notes` 软件包的 **bug**。

贡献升级报告

我们欢迎来自用户的关于从 **etch** 升级到 **lenny** 的任何信息。如果您愿意分享，请在 **Bug** 追踪系统 [<http://bugs.debian.org/>]填写一个关于 `release-notes` 软件包的 **Bug** 以及您所获得的结果。希望您能够(使用 **gzip**)压缩任何附件。

当您提交升级报告的时候，请包含以下信息：

在升级前后，软件包数据库的状态：**dpkg** 的数据库状态信息在 `/var/lib/dpkg/status`，**aptitude** 的数据库状态信息在 `/var/lib/aptitude/pkgstates`。在升级之前，您应该已经象“备份数据和配置文件”一节描述的那样备份，当然您也可以在 `/var/backups` 找到备份信息。

会话日志使用 **script** 来创建，它的描述在“记录会话”一节。

apt 的日志 `/var/log/apt/term.log`，或 **aptitude** 的日志 `/var/log/aptitude`。

注意

在提交 **bug** 报告之前，您应该花点时间从日志中检查和删除任何敏感和/或机密信息，因为这些信息都会被发布在公开的数据库中。

本文档的源文件

本文档的源文件格式是 DocBook XML。HTML 版本使用 `docbook-xsl` 和 `xsltproc` 生成。PDF 版本使用 `dblatex` 或 `xmllroff` 生成。可以从 *Debian* 文档计划的 SVN 仓库获得本文档的源文件。您可以通过网页 [<http://svn.debian.org/viewsvn/ddp/manuals/trunk/release-notes/>]分别访问这些文件，以及查看它们的变更。请参考 *Debian* 文档计划的 SVN 页面 [<http://www.debian.org/doc/cvs/>]，以获取如何访问 SVN 仓库的信息。

第 2 章 **Debian GNU/Linux 5.0** 有什么新东西

Wiki [<http://wiki.debian.org/NewInLenny>] 有更多关于这个主题的信息。

这个版本中加入了对于 ARM EABI (armel) 的官方支持。

下面是 Debian GNU/Linux lenny 官方支持的架构:

- Intel x86 ('i386')
- Alpha ('alpha')
- SPARC ('sparc')
- PowerPC ('powerpc')
- ARM ('arm')
- MIPS ('mips' (big-endian) and 'mipsel' (little-endian))
- Intel Itanium ('ia64')
- HP PA-RISC ('hppa')
- S/390 ('s390')
- AMD64 ('amd64')
- ARM EABI ('armel')

您可以在 Debian 移植页面 [<http://www.debian.org/ports/>], 读到更多的关于您的架构的移植状态和移植细节的信息。

ARM 架构中有什么新东西?

增加了对 Marvell Orion 平台的支持。明确的, Debian GNU/Linux 5.0 支持 Orion 平台的下述设备: QNAP Turbo Station (TS-109 [<http://www.cyrius.com/debian/orion/qnap/ts-109/>], TS-209 [<http://www.cyrius.com/debian/orion/qnap/ts-209/>], TS-409 [<http://www.cyrius.com/debian/orion/qnap/ts-409/>]), HP mv2120 [<http://www.cyrius.com/debian/orion/hp/mv2120/>], 和 Buffalo Kurobox Pro [<http://www.cyrius.com/debian/orion/buffalo/kuroboxpro/>]。

IXP4xx 平台 (例如 Linksys NSLU2) 的以太网驱动已经集成到基本内核中, 所以 lenny 中的 Debian 内核使用此驱动, 替代以前的 Debian 版本使用的非官方驱动。

内置以太网卡需要的 IXP4xx 私有微码由 non-free 中的 `ixp4xx-microcode` 提供。包含此微码的 Debian 安装程序映像位于 slug-firmware.net。

这次发布中有什么新东西?

Debian 的这次发行再次带来了比它前个发行版本 `etch` 更多的软件; 包括 7700 个新软件包, 共计 23200 个。在这个发行版中, 多数软件包得到了更新: 有 13400 个(占 `etch` 软件包总数的 72%)。而且, 由于各种原因, 有相当数量的软件包(3100 个, 占 `etch` 软件包总数的 17%) 从这次发行中被删除了。您将会看不到这些包有任何更新, 而且在包管理软件中它们会被标记为“过时的”。

在这次发行中，Debian GNU/Linux 从 X.Org 7.1 更新到 X.Org 7.3。

Debian GNU/Linux 再次与几个桌面程序和环境一起发布。和其它发行版一样，它现在包含 GNOME 2.22 桌面环境，KDE 3.5.10，以及 Xfce 4.4.2，以及 LXDE 0.3.2.1+svn20080509。还有一些能提高生产效率的软件，包含办公室套件 OpenOffice.org 2.4.1，KOffice 1.6.3，GNUCash 2.2.6，GNUMERIC 1.8.3，以及 Abiword 2.6.4。

其它的桌面应用程序也被更新了，包含 Evolution 2.22.3 和 Pidgin 2.4.3 (即以前的 Gaim)。Mozilla 家族也被更新了：iceweasel (3.0.6 版) 是无商标的 Firefox 网页浏览器，icedove (2.0.0.19 版) 则为无商标的 Thunderbird 邮件客户端。

和过去一样，这次发行还包含了以下软件的更新：

软件包	在版本 4.0(etch) 中的版本	在版本 5.0(lenny) 中的版本
Apache	2.2.3	2.2.9
BIND DNS 服务器	9.3.4	9.5.0
Cherokee 网络服务器	0.5.5	0.7.2
Courier MTA	0.53.3	0.60.0
Dia	0.95.0	0.96.1
Ekiga VoIP 客户端	2.0.3	2.0.12
Exim 默认邮件服务器	4.63	4.69
GNU 编译器套件(默认的编译器)	4.1.1	4.3.2
GIMP	2.2.13	2.4.7
GNU C 函数库	2.3.6	2.7
lighttpd	1.4.13	1.4.19
maradns	1.2.12.04	1.3.07.09
MySQL	5.0.32	5.0.51a
OpenLDAP	2.3.30	2.4.11
OpenSSH	4.3	5.1p1
PHP	5.2.0	5.2.6
Postfix MTA	2.3.8	2.5.5
PostgreSQL	8.1.15	8.3.5
Python	2.4.4	2.5.2
Tomcat	5.5.20	5.5.26

官方 Debian GNU/Linux 发行版现在包含 4 到 5 张二进制 DVD 或 28 到 32 张二进制 CD (依赖于架构)，和 4 张源代码 DVD 或 28 张源代码 CD。此外，还有一个多架构 DVD，包含 amd64 和 i386 架构的子集，和对应的源代码。Debian GNU/Linux 首次发行蓝光镜像，针对 amd64 和 i386 架构，以及源代码。

Debian 现在支持 Linux 标准基础(LSB) 3.2。Debian 4.0 支持的是 3.1 版本。

包管理

aptitude 是从命令行进行包管理的首选软件，它实现了与 **apt-get** 相同的包管理功能，并且被证明比 **apt-get** 能更好地解决依赖关系。如果您依然在使用 **dselect**，您应该把 **aptitude** 作为正式的包管理前端软件。

proposed-updates 区

所有针对 **stable** (或 **oldstable**) 发行版的更改，在它们能被放入发行版之前，都要经过一个外围测试阶段。每一次那样的 **stable** (**oldstable**) 发行版更新被称为是一个发行点。通过 **proposed-updates** 这个机制来完成发行点的准备工作。

软件可以通过两种途径进入 **proposed-updates** 区。首先，加入 **security.debian.org** 的软件包安全更新会被自动加入 **proposed-updates** 区。其次，%**debian**; 开发者可以将新的软件包直接上传至 **proposed-updates** 区。该区最新软件包列表可以在 <http://ftp-master.debian.org/proposed-updates.html> 处看到。

如果您希望在软件包正式加入发行点之前帮忙测试包的更新，您可以通过将 **proposed-updates** 加入 **sources.list** 中进行：

```
deb      http://mirrors.kernel.org/debian lenny-proposed-updates
main contrib
deb-src  http://mirrors.kernel.org/debian lenny-proposed-updates
main contrib
```

下一次您运行 **aptitude update** 命令时，系统将识别 **proposed-updates** 区中的软件包，并在升级包时查找这一区域。

严格意义上来说，这并不是 **Debian** 的新特性，而是一个以前未被太多关注过的特性。

系统改进

这次发行版中的大量改变能使全新安装的 **lenny** 受益，但是可能不会自动启用在从 **etch** 升级上来的系统中。这一节给出大部分相关改变的概述。

SELinux 优先级为
标准，但是默认不启
用






支持 **SELinux**(安全增强型 Linux) 所需的软件包已经被提升为**标准**优先级。这意味着全新安装时它们将会被默认安装。对于已安装好的系统，您可以用下述命令安装 **SELinux**:

```
# aptitude install selinux-basics
```

注意对 **SELinux** 支持不是默认启用的。有关设置和启用 **SELinux** 的信息可以在 **Debian Wiki** [<http://wiki.debian.org/SELinux>] 找到。

新的默认 **syslog** 守
护程序

rsyslog 现在作为 **Debian 5.0** 的默认系统及内核日志记录守护进程，代替了 **syslogd** 和 **klogd**。对于默认日志规则，可以直接使用无需替换。如果有自定义的日志规则，您应该将其迁移到新的配置文件 `/etc/rsyslog.conf`。

从 **etch** 升级的用户需要手工安装 **rsyslog**，删除 **syslogd**。在升级到 **lenny** 的过程中，并不会自动替换系统日志服务程序。

更好地支持 UTF-8

许多软件将被设置成默认使用 UTF-8 或比以前更好地支持 UTF-8。请参阅 <http://wiki.debian.org/UTF8BrokenApps>，查看处理 UTF-8 仍然有问题的应用程序信息。

Identification of
the release's
revision

Starting from Lenny, `/etc/debian_version` will indicate the revision number of the debian release (5.0, then 5.0.1, etc.)

This also means that you should not expect this file to be constant through the release lifetime.

Debian Wiki [<http://wiki.debian.org/Etch2LennyUpgrade>] 有些关于 **etch** 和 **lenny** 之间变更的额外信息。

内核相关的主要变化

Debian GNU/Linux 5.0 对所有架构均使用内核版本 2.6.26。

内核本身以及内核的 **Debian** 打包都有较大的变化。一些变化使升级过程变得困难，而且在升级到 **lenny** 后重启系统时有潜在问题。本节给出大多数重要变化的概述；在后面的章节中包含有潜在问题的解决方法。

内核打包的变化

Binary firmware for some drivers moved to non-free

Some drivers load binary firmware into the device they are supporting at run time. While this firmware was included in the stock kernel in previous releases, it has now be separately packaged in the non-free section. If you want to continue to use these devices after reboot, make sure the required firmware is present on the installed system. See section 6.4 of the Installation Manual [<http://www.debian.org/releases/stable/installmanual>] for details.

新的 OpenVZ 内核标识

Debian GNU/Linux 5.0 现在提供预编译 OpenVZ 的内核镜像，相对于 etch 中的 Linux-VServer 支持来说这是另一种虚拟化解决方案。OpenVZ 以比正常运行稍高一点的开销提供比 Linux-VServer 更好的特性 (如实时迁移)。

X86 架构内核软件包名称已统一

在以前的发行版中，针对 32 位的 AMD Athlon/Duron/Sempron 处理器有一个特殊的 -k7 内核标识。这个标识已被去除并由一个统一的标识 -686 来处理所有 AMD/Intel/VIA 686 型的处理器。

在可能的情况下，依赖于新软件包的伪过渡包已经被提供，以取代过时的软件包。

Emdebian 1.0(基于 Debian GNU/Linux lenny 5.0)



Lenny 现在包含 Emdebian 的制作工具。它允许 Debian 源文件包被交叉构建，并且裁剪，以适合嵌入式的 ARM 系统。

Emdebian 1.0 发布版包含了已经构建好的 ARM 包，足够为特殊的机器和机器变体创建定制的 root 文件系统。内核和内核模块需要单独提供。对 armel 和 i386 的支持正在开发中。请参见 Emdebian 网页 [<http://www.emdebian.org/>] 获得更详细的信息。

上网本支持

上网本，例如华硕的 Eee PC，Debian 是支持的。关于 Eee PC，请看 `eeepc-acpi-scripts`。还有，Debian 提供一个新的轻量级 X11 桌面环境，`lxde`，它有利于上网本或是其它性能相对较低的电脑。

Debian 中的 Java

OpenJDK Java 运行环境 `openjdk-6-jre` 和开发组件 `openjdk-6-jdk` 目前已经包含在 Debian 中了，在执行 Java 图形用户界面程序以及网络程序或是编译这些程序时需要用到它们。这些包编译时使用了 IcedTea 编译支持，以及来自 IcedTea 计划的补丁。

第 3 章 安装系统

Debian 安装程序是 Debian 官方的安装系统。它提供了 Debian 操作系统的各种安装方式。您适合用何种安装方式，取决于您的计算机硬件架构。

您可以借助 Debian 网站 [<http://www.debian.org/releases/stable/debian-installer/>] 上的安装指南找到 lenny 的安装程序映像。

安装手册也可以在 Debian 官方 CD/DVD 的第一张盘上找到，路径是：

```
/doc/install/manual/language/index.html
```


您也可以检查勘误表 [<http://www.debian.org/releases/stable/debian-installer/index#errata>]，以查看 Debian 安装程序的已知问题列表。

安装系统有哪些新特性？

自从在 Debian GNU/Linux 3.1 (sarge) 中第一次官方发布以来，Debian 安装程序有许多开发工作，在硬件支持上有显著改进，还增加了很多有趣的新功能。

在发布说明中，我们只列举安装程序的主要变更。如果您想对从 etch 开始各项变更的详细了解有个大致的了解，请阅读 lenny 之 beta 和 RC 版本的发布声明，它们位于 Debian 安装程序的新闻 [<http://www.debian.org/devel/debian-installer/News/>]网页上。

主要更新

支持在安装期间加载固件	当固件在 Debian 安装媒体之外提供时，现在可以从可移动媒体加载二进制文件。
支持从 Microsoft Windows 安装	现在安装媒体提供一个程序允许系统从 Microsoft Windows 环境中安装 Debian 。
支持 SATA RAID	
尽早升级有安全更新的包	当有网络访问能力时，安装程序会升级全部自发布 lenny 最初版本以来有更新的包。此升级在安装期间，系统重新引导之前进行。
支持 <i>volatile</i>	从而已安装的系统更难被在发布 lenny 到安装期间，已知且被修正的安全问题所影响。 现在安装程序可以选择从 volatile.debian.org 升级安装完毕的系统。此主机提供有规律更新的数据，例如时区定义，病毒特征码，等等。
新移植架构	现在支持 armel 架构。还提供了 i386 Xen 客户安装镜像。
支持硬件语音合成设备	现在安装程序支持几种硬件语音合成设备，因此提高了有视力障碍用户的易访问性。 
支持挂接选项 relatime	现在的安装程序能配置分区使用挂接选项 relatime ，允许仅在以前的读取时间早于当前的修改时间时才更新文件和目录的访问时间。
在安装期间执行 NTP 时钟同步	在安装期间将计算机时钟与 NTP 同步，所以安装好的系统立即有准确的时钟。
新增语言支持	得益于翻译人员的巨大努力， Debian 现在可以在文本界面下以 63 种语言安装(50 种可以在文本用户界面使用，13 中只能在图形用户界面使用)。这比 etch 版多了 5 种语言。此次增加的语言包括阿姆哈拉语、马拉地语、爱尔兰语、爱沙尼亚语以及塞尔维亚人语。由于缺少对应的翻译更新，爱沙尼亚语在本版本中不再被支持。在 etch 中禁用的威尔士语被重新激活。
简化的国家选择	由于它们的字符集无法在非图形界面显示，这些语言只有在图形安装程序中才可以被选择：阿姆哈拉语，孟加拉语，不丹语，古吉拉特语，印地语，格鲁吉亚语，高棉语，马拉雅拉姆语，马拉地语，尼泊尔语，旁遮普语，泰米尔语和泰国语。 现在国家列表按照大陆分组，当用户不想使用与选择的语言关联的国家时，选择国家更容易。

自动安装

前一节中所提到的一些变更，也意味着安装程序使用预配置文件进行自动安装的调整。这意味着，如果您有能在 **etch** 安装程序中正常使用的预配置文件，您不能指望它能在新安装程序中正常使用，而无需修改。

安装指南 [<http://www.debian.org/releases/stable/installmanual>]现在有一个单独附录，用来描述使用预配置文件的方法。

第 4 章 从以前的发行版升级

升级前的准备

我们建议您在升级前阅读第 5 章 *lenny* 中需要注意的问题中的信息。它描述了与升级过程有间接联系的潜在问题，但在升级前了解这些信息对您来说还是很重要的。

备份数据和配置文件

在升级系统前，强烈建议您进行一次完整的备份，或者至少备份您不愿意丢失的数据和配置信息。升级工具和流程是非常可靠的，但升级过程中的硬件错误会对您的系统造成严重损害。

您需要备份的内容包括 `/etc`、`/var/lib/dpkg`、`/var/lib/aptitude/pkgstates` 下的文件以及 `dpkg --get-selections "*"` 命令的输出(命令行中的引号必须要有)。

升级过程本身不会修改 `/home` 下的任何内容。但某些程序(比如部分 **Mozilla** 套件，以及 **GNOME** 和 **KDE** 桌面环境)会在新版本的程序第一次启动时用新的默认值来覆盖现有的用户设置。做为一种预防措施，您也许想备份用户目录下的隐藏文件和目录 (“**dotfiles**”)。该备份有助于您恢复或者重建原有设置。另外，您可能还需要把该事项通知 `apt-get`。

任何软件包安装操作都必须以超级用户权限执行，您可以以 `root` 身份登录，或使用 `su` 或者 `sudo` 来获取必要的访问权限。

升级需要几个前提条件，您应当在实际升级前检查它们。

确保您在使用合适的内核

lenny 的 `glibc` 版本在所有架构将无法与旧于 `2.6.8` 内核工作，而且有些架构还有更高的要求。我们强烈建议您在升级过程开始前，升级到 `etch 2.6.18` 或 `2.6.24` 内核或者最低 `2.6.18` 版本的自定义内核并测试。

提前告知用户

提前通知所有用户您正在计划的任何升级将是明智的决定，但是通过 `ssh` 连接访问您系统的用户可能在升级过程中不会注意到什么东西，而且应该可以继续正常工作。

如果您希望采取更多的预防措施，请在升级前备份或者卸载 `/home` 分区。

当升级到 *lenny* 时您可能需要进行内核升级，所以通常需要重启。一般说来，这将在升级完成后进行。

准备恢复

由于 `etch` 和 *lenny* 的内核在驱动、硬件探测与命名以及设备文件的命名和排序等方面有着诸多变化，在更新后您可能在重启系统时遇到很大的问题。本章和发行注记的下一章记录了很多已知的潜在问题。

由于这个原因，当您的系统重启失败(或对于远程管理系统来说，无法从网络唤醒)时，请确保您能够将其恢复。

如果您通过 `ssh` 远程连接进行升级，则强烈建议您采取必要的预防措施以便通过远程串行终端来访问该服务器。因为在升级内核并重启后，某些设备可能会被改名(如在“设备枚举的顺序”一节中所描述的情况)，而您将不得不通过本地控制台修复系统配置。还有，如果系统在升级过程中意外重启您可能需要使用本地控制台进行修复。

很明显首先应尝试的就是以您原来的内核重启。然而，由于本文档其它地方所描述的各种各样的问题，并不保证可以成功重启。

如果重启失败了，您需要采取替代方案来启动您的系统以便访问和修复它。其中一种选择是使用特制的急救盘或者 **Linux live CD**。以这些介质引导后，您应该可以挂载您的根文件系统并 `chroot` 到它里面来检查并修复问题。

我们推荐的另一种选择是使用 **lenny Debian** 安装程序的拯救模式。使用安装程序的优点是，您可以从众多安装方式中选择最适合您情况的一种。更多信息请查询安装手册 [<http://www.debian.org/releases/stable/installmanual>]第 8 章的“修复损坏的系统”一节以及 Debian 安装程序常见问题 [<http://wiki.debian.org/DebianInstaller/FAQ>]。

initrd 引导时使用调试环境

initramfs-tools 在其生成的 **initrd** 中包含一个调试 **shell**。如果 **initrd** 无法挂载您的根文件系统，您将会进入此调试 **shell**，里面可以使用基本的命令以帮助跟踪问题甚至有可能解决之。

需要检查的几个基本项目：**/dev** 下是否存在正确的设备文件；加载了什么模块(**cat /proc/modules**)；**dmesg** 的输出中有关驱动加载的错误信息。**dmesg** 的输出还会显示哪个设备文件被指定给哪个磁盘；您还应该检查 **echo \$ROOT** 的输出，以确保根文件系统在预想的设备上。

如果您确实修复了问题，输入 **exit** 会退出调试 **shell**，并从先前失败的地方继续启动进程。当然您还需要修复更深层次的问题，并重新生成 **initrd**，以使下次引导不再失败。

为升级准备安全环境

发行版升级应该从本机的文本模式虚拟控制台(或者直连串行终端)，或者通过 **ssh** 远程连接进行。

为了在远程升级时获得额外的可靠性保障，我们建议您在由 **screen** 程序生成的虚拟控制台中执行升级过程。它能进行可靠的重新连接并确保，即使远程连接失败，升级过程也不会被打断。

重要

您不应该在您要升级的机器上使用 **telnet**、**rlogin**、**rsh** 之类的东西，或者从 **xdm**、**gdm**、**kdm** 之类管理的 X 会话中升级。因为那些服务中的任意一个在升级中都有可能被中止，进而导致出现一个不可访问的，只升级了一半的系统。

检查系统状态

此章所述的升级过程是为了从“纯粹的”**etch** 升级而安排的，即系统中没有第三方的软件。为使升级过程最可靠，您可能会希望在升级前，从系统中移除第三方的软件包。

此过程也假定您的系统已经更新至 **etch** 的最新发行点。如果您还未做或是不确定，请按照“升级您的 **etch** 系统”一节中的说明做。

复审包管理器中的未决操作

在某些情况下，使用 **apt-get** 代替 **aptitude** 安装软件包时，可能会让 **aptitude** 认为包“未被使用”，并安排它被移除。一般来说，在正式升级前您应该确保系统是最新且“干净的”。

有鉴于此，您应该复查包管理器 **aptitude** 中是否有未决操作。如果包管理器中有软件包被安排为移除或更新，那么这可能对升级过程有负面影响。注意，要更正这一错误只能是，您的 **sources.list** 仍然指向 **etch** 而非 **stable** 或是 **lenny**。参看“检查您的 **sources list**”一节。

要执行复查的话，您必须以“图形模式”运行 **aptitude** 并按下 **g** (“Go”)。如果显示有任何未决操作，您都应该复查它们，然后要么修复，要么执行建议的操作。如果没有建议的操作，会有一条信息“没有软件包需要安装、删除或者升级”出现。

禁用 **APT pinning** 操作

如果已经设置 **APT** 从一个非 **stable**(如 **testing**)版安装特定软件包，您可能必须改变 **APT pinning** 设置(保存在 **/etc/apt/preferences** 中)以允许升级至新的 **stable** 版中的包。更多有关 **APT pinning** 的信息可以在 **apt_preferences(5)** 中找到。

检查包状态

不管用什么方法升级，我们都建议您先检查所有软件包的状态，并验证所有包都处于可升级状态。以下命令会显示任何半安装或是配置失败的包，还有那些有任何错误状态的包。

```
# dpkg --audit
```

您也可以用 **dselect**、**aptitude** 来审查系统中的所有包的状态，也可以用如下的命令来审查

```
# dpkg -l | pager
```

或是

```
# dpkg --get-selections "*" > ~/curr-pkgs.txt
```

在升级前移除所有的 **hold** 状态是很有必要的。如果有任何对升级而言有重大影响的包处于 **hold** 状态，升级会失败。

注意 **aptitude** 使用一种与 **apt-get** 及 **dselect** 不同的方法来注册 **hold** 状态的包。您可以以下命令来确认 **aptitude** 中处于 **hold** 状态的包。

```
# aptitude search "~ahold" | grep "^h"
```

如果您想检查 **apt-get** 中哪些包设置了 **hold** 状态，可以用

```
# dpkg --get-selections | grep hold
```

如果您在本地更改并重新编译了一个包，且并未重命名，或是在版本号中放入新的纪元号，您就必须将它设为 **hold** 状态，以防止它被升级。

aptitude 中的 “**hold**” 状态的包能用以下命令更改：

```
# aptitude hold package_name
```

用 **unhold** 代替 **hold** 即可清除 “**hold**” 状态。

如果有任何需要修复的东西，您最好确保 **sources.list** 仍然指向 **etch**，就像“检查您的 **sources list**”一节中所说的那样。

proposed-updates

如果您已经把 **proposed-updates** 区加入您的 **/etc/apt/sources.list** 文件，那么您应当在升级前移除它，这样可以降低冲突发生的概率。

非官方源和 **backports**

如果有任何非 **Debian** 的软件包安装在您的系统中，您应该意识到它们可能会在升级期间因为依赖性冲突而被移除。如果这些包是通过在 `/etc/apt/source.list` 中添加一个额外的软件源而安装的，您应该检查那个软件源是否也提供 **lenny** 下编译好的包，并同时像您的 **Debian** 软件源那样更改相应的源。

某些用户可能有非官方的 **backported** “较新的”包版本存在于他们安装的 **Debian etch** 系统中。升级期间那样的包是最可能引起问题的，因为它们可能会导致文件冲突。“升级期间可能遇到的问题”一节已经有一些关于如何处理将会出现的文件冲突的信息了。

使用 **backports.org** 包

backports.org 是一个由 **Debian GNU/Linux** 开发者提供的半官方的仓库，它基于“testing”仓库重新编译，来为 **stable** 版提供新的软件包。

backports.org 仓库含有从“testing”来的包，使用一个比之略小的版本号，这样可以方便地从 **etch** 升级到 **lenny**。回溯至 **lenny** 仍然可用。另外，部分向后移植 (**backports**) 来自于不稳定版 (**unstable**) (包括安全升级及以下特殊程序: **Firefox**, **kernel**, **OpenOffice.org**, **X.org**)。

If you do not use one of these exceptions, you can safely upgrade to lenny. If you use one of these exceptions, set the Pin-Priority (see `apt_preferences(5)`) temporarily to 1001 for all packages from lenny, and you should be able to do a safe dist-upgrade too.

手工删除软件包的标记

要防止 **aptitude** 移除某些因依赖关系而安装的包，您需要手工删除这些包上的 **auto** 标记。对于桌面安装来说，包括 **OpenOffice** 和 **Vim**：

```
# aptitude unmarkauto openoffice.org vim
```

如果您已经用内核元数据包安装了 2.6 内核镜像，那还要包括它：

```
# aptitude unmarkauto $(dpkg-query -W 'linux-image-2.6.*' | cut -f1)
```

注意

您可以复查那些在 **aptitude** 中被标记为 **auto** 的包，执行：

```
# aptitude search '~i~M'
```

为 **APT** 准备源

在开始升级前，您必须在软件包源列表 `/etc/apt/sources.list` 中设定 **apt** 的配置文件。

apt 会查找所有通过任意“deb”行找到的包，并安装带最高版本号的包，给予该文件第一行以优先权(所以，如果有多个镜像地址，您最好先命名本地硬盘，然后是 **CD-ROM**，最后是 **HTTP/FTP** 镜像)。

提示

您可能需要为 **DVD** 和 **CD-ROM** 关闭 **GPG** 密钥检查。如果 `/etc/apt/apt.conf.d/00trustcdrom` 中还没有以下内容，则把下列内容添加至 `/etc/apt/apt.conf`。

```
APT::Authentication::TrustCDROM "true";
```

然而，这无法用于 DVD/CD-ROM 映像文件。

一个发行版通常既能通过它的代号(如: `etch`, `lenny`)引用，也可以用它的状态名引用(如: `oldstable`, `stable`, `testing`, `unstable`)。引用发行版的代号的好处在于，您绝对不会被新的发行版问题困扰且被带至此处。当然，这也意味着您不得不自己关注新版的发行。如果转而使用状态名，一旦有新版发行，您将只会看到一堆可用的软件包的更新。

添加互联网 **APT** 源

默认配置用于从主 Debian 网络服务器上安装，但您可能希望修改 `/etc/apt/sources.list` 以使用其它的镜像，离您最近的网络镜像是一种更好的选择。

Debian HTTP 或 FTP 镜像地址能在 <http://www.debian.org/distrib/ftplist> 找到(参看“Debian 镜像列表”一节)。HTTP 镜像通常比 FTP 镜像快。

例如，假设离您最近的 Debian 镜像是 <http://mirrors.kernel.org>。当用网络浏览器或是 FTP 程序检查那个镜像时，您会注意到主目录被组织成了如下情况：

```
http://mirrors.kernel.org/debian/dists/lenny/main/binary-arm/...
http://mirrors.kernel.org/debian/dists/lenny/contrib/binary-arm/...
```

要让 `apt` 使用这个镜像，您将此行添加至 `sources.list` 文件：

```
deb http://mirrors.kernel.org/debian lenny main contrib
```

注意明显加上去的‘`dists`’，这个发行版名字后的参数用于将路径扩展成多个目录。

添加新源后，在 `sources.list` 中通过在以前存在的“`deb`”行前加上一个井号(`#`)来禁用它们。

添加本地镜像 **APT** 源

您可能希望通过修改 `/etc/apt/sources.list` 来使用一个本地磁盘上的镜像(也许是通过 NFS 挂载的)，来代替 HTTP 或 FTP 软件源镜像。

例如，假设您的镜像位于 `/var/ftp/devian/` 下，并且有如下的主目录：

```
/var/ftp/debian/dists/lenny/main/binary-arm/...
/var/ftp/debian/dists/lenny/contrib/binary-arm/...
```

要让 `apt` 使用它，需要把这一行添加至 `sources.list` 文件：

```
deb file:/var/ftp/debian lenny main contrib
```

注意明显加上去的‘`dists`’，这个发行版名字后的参数用于将路径扩展成多个目录。

添加新源后，在 `sources.list` 中通过在以前存在的“`deb`”行前加上一个井号(`#`)来禁用它们。

从 **CD-ROM** 或 **DVD** 添加 **APT** 源

如果您只想用 CD 安装，在 `/etc/apt/sources.list` 中的“`deb`”行前放上井号(`#`)注释掉它们。

确保在 `/etc/fstab` 中有一行允许您挂载 CD-ROM 于 `/cdrom` 挂载点(**apt-cdrom** 必须指定挂载点为 `/cdrom`)。例如, 假设 `/dev/hdc` 就是您的 CD-ROM, `/etc/fstab` 中应该带有一行:

```
/dev/hdc /cdrom auto defaults,noauto,ro 0 0
```

注意在第四段 `defaults,noauto,ro` 之间必须无空格。

要验证设置是否有效, 插入一片 CD, 尝试运行

```
# mount /cdrom      # 挂载 CD 至挂载点
# ls -alF /cdrom    # 显示 CD 的根目录
# umount /cdrom     # 卸载 CD
```

下一步, 运行:

```
# apt-cdrom add
```

每片您所拥有的 Debian 二进制 CD-ROM 都要这么做, 以便将每片 CD 的数据添加至 APT 的数据库。

升级软件包

推荐的从以前 Debian GNU/Linux 发行版升级的方法是使用包管理器 **aptitude**。此程序会对安装包采取比直接运行 **apt-get** 更保守的选择。

不要忘记挂载所有必需的分区 (尤其是根分区和 `/usr` 分区) 为可读写状态, 用以下命令:

```
# mount -o remount,rw /mountpoint
```

下一步您应该详细检查 APT 源记录 (`/etc/apt/source.list`) 要么指向 “lenny” 又或指向 “stable”。不应该有任何指向 `etch` 的源记录。

注意

CD-ROM 的源通常会指向 “unstable”。尽管这令人困惑, 但您不应该改变它。

记录会话

强烈推荐您使用 `/usr/bin/script` 程序来记录升级会话中的交互信息。这样如果有问题了, 您就有一份问题报告。而且需要的话, 您也可以在错误报告中提供额外信息。要开始记录, 输入:

```
# script -t 2>~/upgrade-lenny.time -a ~/upgrade-lenny.script
```

或是类似命令。不要将输出文件放在临时目录下, 如 `/tmp` 或 `/var/tmp` (这些目录下的文件可能会在升级或重启时被删除)

输出文件也可让您复查屏幕上滚动过去的信息。只要切换至虚拟终端 2 (使用 **Alt+F2**)，在登入后，用 `less -R ~/root/upgrade-lenny.script` 查看文件。

在完成升级后，您可以在提示符下输入 `exit` 停止 **script**。

如果已经对 **script** 使用了 `-t` 选项，您就可以用 **scriptplay** 程序来回放整个过程：

```
# scriptreplay ~/upgrade-lenny.time ~/upgrade-lenny.script
```

更新包列表

首先，需要获取新发行版的可用包列表。执行：

```
# aptitude update
```

首次执行此命令，会更新新源并打印出一些有关源可用性的警告信息。这些警告没关系，而且在您下一次执行时就没有了。

确保您有足够的空间升级

在升级系统前必须确保在进行如“升级系统的其它部分”一节中所述的完全系统升级时，您有足够的磁盘空间。首先，安装过程所需的任何从网上下载的包都被保存在 `/var/cache/apt/archives` (下载期间还会放在 `partial/` 子目录下)，因此您必须确保在文件系统分区 `/var/` 下有足够的空间存放临时下载包，这些包将会安装在系统中。下载之后，您可能在其它文件系统分区中需要更多的空间来同时安装升级包(可能会包含更大的二进制程序或更多数据)和升级中带入的新包。如果系统中没有足够的空间，您可能以一个未完成的升级结束，而这可能难以恢复。

aptitude 和 **apt** 都会显示给您安装所需磁盘空间的详细信息。在执行升级操作前，您可以运行以下命令来看到这个估计值：

```
# aptitude -y -s -f --with-recommends dist-upgrade
[ ... ]
XXX 个已升级，XXX 个刚装上，XXX 个待删除以及 XXX 个未升级。
需要获取 xx.xMB/yyyMB。解开后，将占用 AAAMB。
会下载/安装/删除包。
```

注意

由于下一节将要谈到的问题，在即将开始更新前执行这条命令可能会给出一个错误。如果那样的话，您需要等至已经如“最小系统升级”一节所述的那样做过最小系统升级并且升级过内核，这样您才可以运行这条命令来预估所需的磁盘空间。

如果没有足够空间来升级，确保您预先释放了空间。可以这样：

删除以前安装时下载的包 (位于 `/var/cache/apt/archives`)。用 **apt-get clean** 或 **aptitude clean** 来清理包缓存, 这会删除所有以前下载过的包文件。

删除被遗忘的包。如果您安装过 `popularity-contest`, 可以用 **popcon-largest-unused** 来列出系统中占用大量空间而又不用的那些包。您也可以用 **debporphan** 或是 **debfostr** 来找出过时的包(参看“过时的包”一节)。另一种可选方案是, 您以“图形模式”运行 **aptitude** 在“过时的且由本机创建的包”下找到过时的包。

删除占用太多空间且目前不需要的包(您总是可以在升级后重装它们的)。您可以用 **dpigs** (可在 `debian-goodies` 包中找到)或是用 **wajig** (运行 `wajig size`) 列出占用大量空间的包。

You can list packages that take up most of the disk space with `aptitude`. Start **aptitude** into “visual mode”, select Views → New Flat Package List (this menu entry is available only after etch version), press **I** and enter `~i`, press **S** and enter `~installsize`, then it will give you nice list to work with. Doing this after upgrading `aptitude` should give you access to this new feature.

如果不需要的话可以从系统中移除翻译和本地化文件。您可以安装并配置 `localepurge` 软件包, 这样只有一小部分选择的本地化设置被保留在系统中。这将减少 `/usr/share/locale` 所使用的磁盘空间。

将 `/var/log/` 下的系统日志临时放到其它系统, 或是永久删除。

使用临时目录 `/var/cache/apt/archives`: 您可以使用来自另一个系统的临时缓存目录(USB 存储设备, 临时硬盘, 使用中的文件系统.....)

注意

不要使用 NFS, 这是因为网络连接可能在升级期间断开。

例如, 假设您将 USB 盘挂在 `/media/usbkey` 下:

删除以前安装时下载的包:

```
# apt-get clean
```

将 `/var/cache/apt/archives` 目录复制到 USB 磁盘:

```
# cp -ax /var/cache/apt/archives /media/usbkey/
```

将临时缓存目录挂载至当前目录:

```
# mount --bind /media/usbkey/archives /var/cache/apt/archives
```

升级后, 还原 `/var/cache/apt/archives` 目录:

```
# umount /media/usbkey/archives
```

删除留下的 `/media/usbkey/archives`。

您可以在挂载于系统中的任何文件系统上创建临时缓存目录。

注意为了安全地移除软件包, 建议您换回 etch 的 `sources.list`, 就像“检查您的 `sources list`”一节中描述的那样。

先升级 **apt** 和/或 **aptitude**

Several bug reports have shown that the versions of the `aptitude` and `apt` packages in `etch` are often unable to handle the upgrade to `lenny`. In `lenny`, `apt` is better at dealing with complex chains of packages requiring immediate configuration and `aptitude` is smarter at searching for solutions to satisfy the dependencies. These two features are heavily involved during the `dist-upgrade` to `lenny`, so it is necessary to upgrade these two packages before upgrading anything else.

The following command will upgrade both `aptitude` and `apt`:

```
# aptitude install aptitude apt dpkg
```

This step will also automatically upgrade `libc6` and `locales`. At this point, some running services will be restarted, including **xdm**, **gdm** and **kdm**. As a consequence, local X11 sessions might be disconnected.

Upgrading with `apt`

Please note that using **apt-get** is not recommended for the upgrade from `etch` to `lenny`. If you do not have `aptitude` installed you are recommended to install it first.

If you want to perform the upgrade with `apt` or if the upgrade with `aptitude` failed and you want to try the upgrade with `apt`' dependency chain resolution algorithm, you should run:

```
# apt-get install apt
```

Note that you will have to adapt other **aptitude** commands to use **apt-get** instead.

使用 `aptitudes` 记录的 `apt` 自动安装的软件包列表

`aptitude` 维护了一个自动安装的包列表(就像另一个包的依赖一样)。在 `lenny` 中, `apt` 现在也有这个功能了。

首次运行 `lenny` 版本的 `aptitude` 时, 它会先自动读入安装包的列表, 然后再转换成 `lenny apt` 可用的版本。如果您安装了 `aptitude`, 您至少应该先发出一条 **aptitude** 指令以便转换。一种方法是搜索一个不存在的包:

```
# aptitude search "?false"
```

最小系统升级

由于在 `etch` 和 `lenny` 间有某些必需包存在冲突, 因此直接运行 `aptitude dist-upgrade` 通常会删除大量您想要保留的包。因此我们推荐一种两步升级过程, 首先做一次最小升级来解决这些冲突, 然后做一次完整的 `dist-upgrade` 升级。

首先, 运行:

```
# aptitude safe-upgrade
```

这会升级那些不需要删除或安装其它任何包的软件。

下一步会相当依赖您安装过的包。发行注记会给出所用方法的通用建议, 但如有疑问, 推荐您在继续操作前检查每种方法建议的需要删除的包。

某些通用包要被删除，包括 `base-config`，`hotplug`，`xlibs`，`netkit-inetd`，`python2.3`，`xfree86-common` 和 `xserver-common`。要了解更多在 `lenny` 中废弃包的信息，参看“过时的包”一节。

升级系统的其它部分

现在，您要准备继续升级重要部分。执行：

```
# aptitude dist-upgrade
```

这将对系统进行一次全面的升级，比如：安装所有包的最新可用版本，解决不同发行版的包之间所有潜在的依赖性变动。如有必要，它会安装一些新包(通常是新库，或是重命名的包)，并删除任何有冲突的过时包。

当从一套 CD-ROM (或 DVD)升级时，您会在升级期间的某几个地方被要求插入指定的 CD 盘片。您可能不得不多次插入相同的盘片，这是因为互相依赖的包已经被分散到整个 CD 套件中去了。

如果不改变被指定在当前状态(显示为“held back”)的另一个包，当前安装包的新版本就无法升级。这个问题可以通过使用 **aptitude** 选择这些包来安装或是用 `aptitude -f install package` 来解决。

升级期间可能遇到的问题

如果使用 **aptitude**, **apt-get** 或 **dpkg** 操作时出现错误

```
E: Dynamic MMap ran out of room
```

默认的缓存区空间不足。您可以通过删除或注释掉 `/etc/apt/sources.list` 中不需要的行，或通过增加缓存大小来解决这个问题。缓存大小能通过 `/etc/apt/apt.conf` 中设置 `APT::Cache-Limit` 来增加。下面的命令会为它设置一个足够升级值：

```
# echo 'APT::Cache-Limit "12500000";' >> /etc/apt/apt.conf
```

这里假设您还未在那个文件中对该变量进行过设置。

有时有必要在 APT 中开启 `APT::Force-LoopBreak` 选项使之能够临时移除一个重要的包，这是为了满足冲突或是预依赖循环。**aptitude** 会警告您这一点并退出升级。您可以通过在 **aptitude** 命令行上指定 `-o APT::Force-LoopBreak=1` 选项来解决这个问题。

有可能系统的依赖关系太乱了以至于需要手工干预。通常这意味着使用 **aptitude** 或是

```
# dpkg --remove package_name
```

来消除某些引起问题的包，或是

```
# aptitude -f install  
# dpkg --configure --pending
```

```
# dpkg --install /path/to/package_name.deb
```

如果您从“纯粹的” **etch** 系统升级就不会出现文件冲突，但如果您装有非官方的 **backport** 包就可能出现冲突。文件冲突会导致类似以下这样的错误：

```
Unpacking <package-foo> (from <package-foo-file>) ...  
dpkg: error processing <package-foo> (--install):  
  trying to overwrite `<some-file-name>',  
  which is also in package <package-bar>  
dpkg-deb: subprocess paste killed by signal (Broken pipe)  
Errors were encountered while processing:  
<package-foo>
```

您可以尝试用错误信息中的最后一行提示，强制删除包来解决文件冲突：

```
# dpkg -r --force-depends package_name
```

在修复这些问题后，您应该可以通过重复前述的 **aptitude** 命令来继续升级。

升级期间，您会被问到有关配置或是重新配置几个包的问题。如果您被问到是否任何存在于 `/etc/init.d` 或 `/etc/terminfo` 目录，或是 `/etc/manpath.config` 目录下的文件应该被包维护者的版本所替换时，通常有必要回答‘yes’来确保系统一致性。由于它们会被以 `.dpkg-old` 为后缀的文件名保存，所以您总是可以恢复成老版本的配置文件。

如果您不确定该做什么，那就记下包或文件的名称，并稍后排出它们的顺序。您可以通过在 `script` 命令记录的输出文件中搜索来查看升级期间显示在屏幕上的信息。

升级内核与相关包

这一节解释了如何升级您的内核，以及明确有关此次升级的潜在问题。您可以安装由 Debian 提供的 `linux-image-*` 包，或者从源码编译一个自定义的内核。

注意本节的很多信息基于假设，即您会使用一个模块化的 Debian 内核，以及 `initramfs-tools` 和 `udev`。如果您选择使用一个不需要 `initrd` 的自定义内核，或是您使用了一种不同的 `initrd` 生成器，对您来说某些信息可能会不一样。

安装内核元数据包

当您从 `etch` 全面升级至 `lenny` 时，强烈推荐您安装新的 `linux-image-2.6-*` 元数据包。此包可能由 `dist-upgrade` 自动安装了。您可以验证此步，执行：

```
# dpkg -l "linux-image*" | grep ^ii
```

如果您没有看到任何输出，那么您需要手工安装一个新的 `linux-image` 包。要查看可用的 `linux-image-2.6` 的元数据包列表，执行：

```
# apt-cache search linux-image-2.6- | grep -v transition
```

如果您不确定要选哪个包，那就运行 `uname -r` 并查找带有类似名称的包。例如，如果您看到‘2.6.18-6-686’，那推荐您安装 `linux-image-2.6-686`。(注意不再有 k7 版本；如果您当前正在用 k7 内核，您应该安装 686 内核代替。)您也可以使用 `apt-cache` 来查看每个包的详细描述，以帮助您选择最好用的那个。例如：

```
# apt-cache show linux-image-2.6-686
```

然后您应该使用 `aptitude install` 来安装。一旦安装了这个新内核，您应该在下次可行时重启机器以获取新内核所提供的特性。

对于爱冒险的人，这里有个简单方法来编译您的自定义 Debian GNU/Linux 内核。安装 `kernel-package` 工具，然后阅读 `/usr/share/doc/kernel-package` 下的文档。

如果有可能，把内核包的升级从主 `dist-upgrade` 中独立出来是有利的，这能减少出现一个临时不可引导的系统的机会。注意，这只能在最小升级(参看“最小系统升级”一节)结束后才能做。

设备枚举的顺序

`lenny` 带来一个比上一发行版更健全的硬件探测机制。但是，这会引入系统中所探测到的设备顺序变化，即影响设备名的分配顺序。例如，如果您有两张驱动不同的网卡，`eth0` 和 `eth1` 设备名可能被互换了。请注意，新机制意味着假设您在运行着的 `lenny` 系统中互换以太网适配器，新的适配器也会获得一个新的接口名。

对于网络设备，您可以通过使用 `udev` 机制避免此重排，更明确点，通过在 `/etc/udev/rules.d/70-persistent-net.rules` 中指定。另一种方法，您可以在引导时用 `ifrename` 工具将物理设备绑定至指定的名字。参看 `ifrename(8)` 和 `ifstab(5)` 获取更多信息。两个可选方法 (`udev` 和 `ifrename`) 不应该同时使用。

对于存储设备，通过使用 `initramfs-tools` 并配置它按当前的顺序载入存储设备，这样您就可以避免这个顺序重排。要这么做的话，通过查看 `lsmod` 命令的输出来明确您系统中载入的存储模块的顺序。`lsmod` 以它们载入的反序显示，即列表中的第一个模块是最后一个被载入的。注意这只对内核以固定顺序枚举出的设备有效(像 PCI 设备)。

但是，在首次引导后删除并重新载入模块会影响这个顺序。还有，您的内核可能有些静态链接的驱动，而且这些模块名称不会出现在 `lsmod` 的输出中。您可能会解读这些驱动名并从 `/var/log/kern.log` 或是 `dmesg` 的输出中载入引导顺序。

将这些模块名按它们引导时载入的顺序添加至 `/etc/initramfs-tools/modules`。某些模块名可能在 `etch` 和 `lenny` 有所变动。例如，`sym53c8xx_2` 已经变成了 `sym53c8xx`。

然后您需要运行 `update-initramfs -u -k all` 重新生成 `initramfs` 镜像。

一旦您正运行一个 `lenny` 内核和 `udev`，您可以重新配置系统以通过一个不依赖于驱动载入顺序的别名来访问磁盘。这些别名存在于 `/dev/disk/` 目录结构下。

引导期间的问题

如果用 `initramfs-tools` 创建的 `initrd` 引导系统，在某些情况下由 `udev` 创建的设备文件可能对于启动脚本的执行来讲太慢了。

这通常表现为引导失败，由于根文件系统无法挂载，您会进入调试环境。但当您回头检查时，所需的所有设备都存在于 `/dev` 下。只要根文件系统存在于 USB 磁盘或是 RAID 上，尤其是使用了 `LILO` 时，就会出现这种问题。

解决此问题的方法就是用引导参数 `rootdelay=9`。这个超时值(秒)可能需要调整一下。

在重启之前要做的事情

当 `aptitude dist-upgrade` 完成时，“正式的”升级就算结束了。但在下一次重启之前，还有一些其它的事情需要关注一下。

重新运行 `lilo`

如果您正在用 `lilo` 作为您的启动管理器(这是某些 `etch` 安装的默认引导器)，强烈建议您在升级后重新运行 `lilo`：

```
# /sbin/lilo
```

注意，即使您不升级系统内核也需要这一步，这是由于包的升级 `lilo` 的 `second stage` 会改变。

还有，检查 `/etc/kernel-img.conf` 中的内容并确保其中有 `do_bootloader = Yes` 这一行。这样，在内核升级后，启动管理器总会重新运行一下。

如果您在运行 **lilo** 时遇到任何问题，请检查 / 下至 **vmlinuz** 和 **initrd** 的符号链接，以及 **/etc/lilo.conf** 中内容不一致的地方。

如果您在重启前或是在您未手工重启系统就偶然重启前，忘记重新运行 **lilo** 的话，您的系统可能会无法引导。当系统引导时您将只会看到 **LI** 而不是 **lilo** 提示符。有关如何从此处恢复的信息请参看“准备恢复”一节。

系统引导时在显示 **Waiting for root file system** 后挂起

从 **/dev/hda** 变成 **/dev/sda** 的补救方法

某些用户已经报告，升级可能引起内核无法在重启后找到系统根分区。

在那种情况下，系统引导会挂起并出现以下信息：

```
Waiting for root file system ...
```

且几秒钟后会只出现一个 **busybox** 提示符。

当内核升级使用新一代的 **IDE** 驱动程序时，就会出现这个问题。老的 **IDE** 磁盘的命名规则是 **hda, hdb, hdc, hdd**。新的规则将分别命名为 **sda, sdb, sdc, sdd**。当升级时未生成一个新的 **/boot/grub/menu.lst** 文件来产生新的命名规则时，就会出现这个问题。引导时，**Grub** 会传递给内核一个无法找到的系统根分区。

如果您在升级后已经遇到此问题，转至“如何在升级后解决此问题”一节。要在升级前避免此问题，就提前看一下。

如何在升级前避免此问题

通过给根文件系统使用一个不会在引导时改变的标识符，您可以完全避免这个问题。要做到这点有两种可用的方案——给文件系统加标签，或是用文件系统的通用唯一标识符 (**UUID**)。Debian 从 'etch' 发行版起就已经支持这些方法了。

两种方法各有优缺点。文件系统标签可读性好，但如果您的系统中有另一个同名的文件系统时，就会带来问题。**UUID** 可读性较差，但极不可能出现两个互相冲突的 **UUID**。

以下范例中，我们假设根文件系统位于 **/dev/hda6**，同时再假设您的系统装有运行正常的 **udev** 和 **ext2** 或是 **ext3** 文件系统。

要使用文件系统标签：

通过执行命令 **e2label /dev/hda6 rootfilesystem** 来标记文件系统（名称必须小于 16 个字符）。

编辑 `/boot/grub/menu.lst` 并更改以下这行：

```
# kopt=root=/dev/hda6 ro
```

为

```
# kopt=root=LABEL=rootfilesystem ro
```

注意

不要删除行首的 **#**，那个要的。

通过运行 **update-grub** 更新 `menu.lst` 中的 `kernel` 行。

编辑 `/etc/fstab` 并改变挂载 `/` 分区的那行，如：

```
/dev/hda6      /      ext3 defaults,errors=remount-ro 0 1
```

为

```
LABEL=rootfilesystem      /      ext3 defaults,errors=remount-ro 0 1
```

在这里改变的是第一栏，您不必修改此行的其它栏目。

要使用 UUID：

Find out the universally unique identifier of your filesystem by issuing: **ls -l /dev/disk/by-uuid | grep hda6**. You can also use **vol_id --uuid /dev/hda6** (in etch) or **blkid /dev/hda6** (if already upgraded to lenny).

您应该得到类似于以下内容的一行:

```
lrwxrwxrwx 1 root root 24 2008-09-25 08:16 d0dfcc8a-417a-41e3-  
ad2e-9736317f2d8a -> ../../hda6
```

UUID 是指向 `/dev/hda6` 的符号链接的名字, 如 `d0dfcc8a-417a-41e3-ad2e-9736317f2d8a`。

注意

您文件系统的 UUID 可能是不同的字符串。

编辑 `/boot/grub/menu.lst` 并更改以下这行:

```
# kopt=root=/dev/hda6 ro
```

为

```
# kopt=root=UUID=d0dfcc8a-417a-41e3-ad2e-9736317f2d8 ro
```

注意

不要删除行首的 `#`, 那个要的。

通过运行 **update-grub** 更新 `menu.lst` 中的 `kernel` 行。

编辑 `/etc/fstab` 并改变挂载 `/` 分区的那行, 如:

```
/dev/hda6 / ext3 defaults,errors=remount-ro 0 1
```

为

```
UUID=d0dfcc8a-417a-41e3-ad2e-9736317f2d8 / ext3  
defaults,errors=remount-ro 0 1
```

在这里改变的是第一栏, 您不必修改此行的其它栏目。

如何在升级后解决此问题

方案一

当 **Grub** 显示菜单界面让您选择您想启动的菜单项时, 可以做这步操作。如果没出现那样的菜单, 尝试在内核引导显示它之前, 按下 **Esc** 键。如果您还是无法进入此菜单, 尝试“方案二”一节或是“方案三”一节。

在 **Grub** 菜单中，高亮您想启动的菜单项。按下 **e** 键来编辑有关此条目的选项。您将会看到一些东西，类似于：

```
root (hd0,0)
kernel /vmlinuz-2.6.26-1-686 root=/dev/hda6 ro
initrd /initrd.img-2.6.26-1-686
```

高亮此行

```
kernel /vmlinuz-2.6.26-1-686 root=/dev/hda6 ro
```

按下 **e** 键并用 **sdX** 代替 **hdX** (**X** 可以是字符 **a, b, c** 或 **d**，这依赖于您的系统)。在我的例子中这一行变为：

```
kernel /vmlinuz-2.6.26-1-686 root=/dev/sda6 ro
```

然后按下 **Enter** 来保存修改。如果还有其它带 **hdX** 字样的行，则一并修改。不要改变类似于 **root (hd0,0)** 这样的条目。一旦做好所有的修改，按下 **b** 键。然后您的系统现在应该能正常启动。

现在您的系统已经启动了，您需要永久修复这个问题。转至“如何在升级前避免此问题”一节并使用两个方法中的一个。

方案二

从 **debian** 安装介质(CD/DVD)引导，选择 **rescue** 来启动急救模式。选择您的语言、地区、键盘映射，无论是否成功都让它配置您的网络。过一会儿，您应该被要求选择一个您想让它作为根文件系统使用的分区。推荐的选择看起来类似：

```
/dev/ide/host0/bus0/target0/lun0/part1
/dev/ide/host0/bus0/target0/lun0/part2
/dev/ide/host0/bus0/target0/lun0/part5
/dev/ide/host0/bus0/target0/lun0/part6
```

如果您知道哪个分区是您的根文件系统，就选择正确的那个。如果您不知道，只好试一下第一个。如果它报告说是无效的根文件系统分区，就试下一个，依次类推。一个接一个地尝试不会损伤您的分区，而且如果您仅在磁盘上装有一个系统，您应该很容易就找出正确的根文件系统分区。如果您在磁盘上装有多系统，最好能准确地知道哪个是正确的分区。

一旦您已选中一个分区，有几个选项可以选择。选择在所选分区中启动 **shell** 环境的选项。如果启动失败，那就试另一个分区。

现在您应该拥有挂载于 **/target** 的根文件系统 **shell** 环境的 **root** 用户访问权。您需要访问硬盘中 **/boot**、**/sbin** 和 **/usr** 目录中的内容，它们现在应该位于 **/target/boot**、**/target/sbin** 和 **/target/usr** 目录下。如果这些目录需要从其它分区挂载，那就先挂载它们。(如果您不知道挂载哪个分区，请参考 **/etc/fstab**)。

转至“如何在升级前避免此问题”一节并使用两个方法中的一个来永久修复这个问题。然后输入 **exit** 退出急救 **shell** 环境，并选择 **reboot** 正常重启系统(别忘了拿走可引导介质)。

方案三

从您最喜欢的 LiveCD 发行版启动，比如 Debian Live, Knoppix, Ubuntu Live。

挂载 `/boot` 目录所在的分区。如果您不知道是哪个，利用 `dmesg` 的输出来找是否您的磁盘是 `hda, hdb, hdc, hdd` 或是 `sda, sdb, sdc, sdd`。一旦您知道是哪个磁盘，例如是 `sdb`，执行命令 `fdisk -l /dev/sdb` 来查看分区表并找出正确的分区。

假设，您已经在 `/mnt` 下挂载了正确的分区，而且此分区含有 `/boot` 目录及其内容，那就编辑 `/mnt/boot/grub/menu.lst` 文件。

找到类似于以下的这一段：

```
## ## End Default Options ##

title          Debian GNU/Linux, kernel 2.6.26-1-686
root           (hd0,0)
kernel        /vmlinuz-2.6.26-1-686 root=/dev/hda6 ro
initrd        /initrd.img-2.6.26-1-686

title          Debian GNU/Linux, kernel 2.6.26-1-686 (single-
user mode)
root           (hd0,0)
kernel        /vmlinuz-2.6.26-1-686 root=/dev/hda6 ro single
initrd        /initrd.img-2.6.26-1-686

### END DEBIAN AUTOMAGIC KERNELS LIST
```

并将 `hda, hdb, hdc, hdd` 替换为 `sda, sdb, sdc, sdd`。不要修改类似于这样的行：

```
root           (hd0,0)
```

重启系统，弹出 LiveCD，您的系统应该能正确的启动了。

当系统引导时，使用“如何在升级前避免此问题”一节中所建议的两种处理方法中的一个来永久修复这个问题。

为下个发行版做准备

升级完成后，您可以为下个发行版做点事。

如果新的内核镜像元数据包被当作旧镜像的依赖包而安装，那新镜像将被标记为自动安装，这应该被更正一下：

```
# aptitude unmarkauto $(dpkg-query -W 'linux-image-2.6-*' | cut -f1)
```

像“过时的包”一节中描述的那样移除过时的和未使用的包。您应该核查这些包所用的配置文件，并考虑完全删除这些软件包以移除它们的配置文件。

过时的包

在带来几千个新软件包的同时，**lenny** 也清除了曾经存在于 **etch** 中超过两千个的旧软件包。**lenny** 不提供这些过时包的升级。当然，在您需要的地方没什么可以阻止您继续使用这些过时的包，**Debian** 计划通常会在 **lenny** 发行版后的一年间不断的提供安全更新，且通常不会同时提供其它支持。推荐用可用的替代品代替它们。

为什么包会从发行版中移除？这有很多原因，如：它们不再被上游作者维护了；没有 **Debian** 开发者对维护这个包感兴趣；这些包提供的功能被不同的软件（或新版本）替代了；或者由于它们自身的问题，因而被认为不适用于 **lenny**。最后一种情况下，这些包有可能仍然位于“**unstable**”版之下。

检测在已更新的系统中哪些包是“过时”的现在容易得很，因为包管理的前台程序会标记它们的。如果您使用的是 **aptitude**，您将在“过时的和本机创建的包”条目中看到这些包。**dselect** 提供类似的条目，但它显示的列表有所不同。

还有，如果您曾经用 **aptitude** 手工在 **etch** 中安装过包，**aptitude** 会保存那些您手工安装的记录。如果有某个包被移除并导致出现无依赖性，即不再需要那些包，**aptitude** 会将那些包标记为过时的包。还有 **aptitude**，不同于 **deborphan**，不将您手工安装而非那些通过依赖性自动安装的软件包标记为过时的包。

您可以用其它的工具来找出过时的包，例如 **deborphan**，**debfoaster** 或 **cruft**。推荐使用 **deborphan**，尽管它(默认情况下)只报告过时的库，即“**libs**”或是“**oldlibs**”部分中的未被其它包依赖的软件包。不要盲目的移除那些工具要用到的包，尤其是当您正在使用激进的非默认的选项，它们容易引起严重错误。极度推荐在您移除它们之前，手工核查那些建议移除的包(例如：它们的内容、大小以及描述信息)。

Debian 错误跟踪系统 [<http://bugs.debian.org/>]通常会提供额外信息，这些都是有关这个包为什么要被移除的信息。您应该既查看此包自身的归档错误报告，同时也要看一下 [ftp.debian.org pseudo-package](http://ftp.debian.org/pseudo-package) [<http://bugs.debian.org/cgi-bin/pkgreport.cgi?pkg=ftp.debian.org&archive=yes>] 中的归档错误报告。

The list of obsolete packages includes:

apache_{1.3.3} (1.x), 由 apache2_{2.2.3} 代替
bind_{9.3.3} (8), successor is bind9_{9.3.3}
php4_{4.4.0}, 由 php5_{5.2.6} 代替
postgresql-7.4_{7.4.6}, 由 postgresql-8.1_{8.1.3} 代替
exim_{4.73} (3), successor is exim4_{4.73}

哑包

来自 **etch** 的某些包在 **lenny** 中已经被分为好几个包，通常这是为了改善系统的可维护性。在此情况下，为了能平滑升级，**lenny** 通常提供“哑”包(即空包)，它与 **etch** 中的旧包名称相同，但有依赖性，促使它们安装新包。这些“哑”包在升级成功后就会被认为是过时的包，且能够被安全地移除。

绝大多数(并非所有)哑包的描述信息会显示它们的用途。但哑包的描述信息并不统一，因此您可能会发现带上 **--guess** 选项的 **deborphan** 命令对于找出您系统中的哑包是很有用的。注意，某些哑包在升级后是不打算被系统移除的，相反它们被用于长期跟踪某程序的当前可用版本这样的目的。

下个 **Debian** 发行版的计划

抛弃 **ARM ABI**，而改用 **ARM EABI**

Debian lenny 有两个不同，并且不兼容的 ARM 版本：老的 ABI (arm) 和新的 EABI (armel)。Debian lenny 是最后一个支持 ARM ABI 的发行版，以后的发行版将仅支持 ARM EABI 或 armel。因此，全新安装 lenny 时建议使用 armel。

请浏览此页 [<http://wiki.debian.org/ArmEabiPort>]来了解更多的有关 ARM EABI (armel) 的信息。

第 5 章 **lenny** 中需要注意的问题

潜在的问题

有时候，我们无法避免某些变化带来的副作用，或者我们在别的地方暴露了 **Bug**。我们在这里记录我们已知的问题。请您同时阅读勘误，相关软件包的文档，**Bug** 报告以及在“更多读物”一节中提到的其它信息。

与 **udev** 相关的设备的问题

虽然 **udev** 经过了大量的测试，您仍可能在使用某些设备时遇到问题，这些需要修复。最常见的情况是需要修改设备的访问许可和所有权。有些情况下设备可能不会被默认创建(比如 `/dev/video` and `/dev/radio`)。

udev 提供配置机制以处理这些问题。请见 `udev(8)` 和 `/etc/udev` 以获得更多信息。

某些应用程序可能无法在 **2.4** 内核下工作

某些 **lenny** 中的程序可能无法在 2.4 内核下工作，例如有些需要 `epoll()` 支持，而 2.4 内核中没有。这些程序可能根本无法工作或者工作不正常，除非系统重启并使用 2.6 内核。

一个例子是 HTTP 代理 **squid**。

某些网站无法通过 **TCP** 进行连接

从 2.6.17 开始，Linux 大量使用 RFC 1323 中描述的 TCP 窗口缩放。某些服务器会有不正常的行为，并向他们自己宣布错误的窗口大小。更多细节，请参考 **bug** 报告 #381262 [<http://bugs.debian.org/381262>], #395066 [<http://bugs.debian.org/395066>], #401435 [<http://bugs.debian.org/401435>]。

通常有两个替代方法：或者将最大允许 TCP 窗口大小改回到一个较小的值(推荐)或者彻底关闭 TCP 窗口缩放(不推荐)。请参见 **debian-installer errata page** [<http://www.debian.org/devel/debian-installer/errata>] 中的示例命令。

无法自动关机

在一些旧系统中，`shutdown -h` 可能无法关闭机器(只是将其停止)。这是因为这里需要使用 **APM**。将 `acpi=off apm=power_off` 添加到内核命令行，比如添加在 `grub` 或者 `lilo` 的配置文件中，应该可以修正这个问题。请参见 **bug** #390547 [<http://bugs.debian.org/390547>] 以获取更多信息。

异步网络初始化可能引起无法预测的情况

在使用 **udev** 为网络接口加载驱动的系统上，因为 **udev** 的异步天性，使得系统启动时网络驱动可能无法在 `/etc/init.d/networking` 执行前加载完成。虽然在 `/etc/network/interfaces` 中加入 `allow-hotplug` (与 `auto` 一起使用)能保证在网络接口可用才将其启用，这仍然无法保证在启动进程开始启动网络服务前完成，而这些网络服务可能因没有网络接口而无法正常运行。

使用 **WPA** 保护无线网络的问题

WPA

在 `etch` 中，`wpa_supplicant` 软件包设置为系统服务，通过 `/etc/default/wpa_supplicant` 和用户提供的 `/etc/wpa_supplicant.conf` 进行配置。

在 lenny 中, /etc/init.d/wpa_supplicant 被抛弃, 现在此 Debian 软件包已经和 /etc/network/interfaces 集成, 类似其它软件包如 wireless-tools。这意味着 wpa_supplicant 不再直接提供系统服务。

更多有关配置 wpa_supplicant 的信息请参考 /usr/share/doc/wpa_supplicant/README.modes.gz, 里面有 /etc/network/interfaces 文件的例子。Debian 中有关使用 wpa_supplicant 软件包的更新信息可以在 Debian Wiki [http://wiki.debian.org/WPA] 中找到。

文件名中使用非 **ASCII** 字符可能引起问题

挂载 vfat, ntfs 或者 iso9660 文件系统时, 如果其中有文件的名称包含非 ASCII 字符时, 会在尝试使用这些文件名时失败, 除非使用 utf8 选项来挂载。迹象是可能会出现这样的失败: “无效或不完整的多字节字符或宽字符”。可行的解决方法是使用 defaults,utf8 作为 vfat, ntfs 和 iso9660 文件系统的挂载参数, 如果它们包含有非 ASCII 字符名称的文件。

请注意当使用 utf8 选项时 Linux 内核不支持处理 vfat 文件系统时忽略文件名的大小写。

声音停止工作

在极端情况下, 升级后可能无声。如果发生了这种情况, 请执行 ALSA 检查列表:

- 以 root 身份执行 **alsaconf**,
- 将你的用户增加到组 **audio**,
- 确认各个选项已开启并没有静音(使用命令 **alsamixer**),
- 确认 **arts** 和 **esound** 没有启动,
- 确认没有 OSS 模块被加载,
- 确认扬声器已打开, 以及
- 检查如下的命令

```
cat /dev/urandom > /dev/audio
```

or the command

```
speaker-test
```

能否以root用户运行。

NFS 挂载现在由 **nfs-common** 来处理

从 util-linux 2.13 开始 NFS 挂载不再由 util-linux 本身处理, 而是由 nfs-common 来处理。由于不是所有的系统都要挂载 NFS 共享分区而且为了避免安装标准端口映射 util-linux 仅仅建议安装 nfs-common。如果您需要挂载 NFS 共享分区, 请确保在您的系统中安装 nfs-common。mount 软件包的预安装脚本会检查是否存在已挂载的 NFS 分区, 并且如果 nfs-common 中的 /usr/sbin/mount.nfs 不存在或者 nfs-common 过时就会中止。此时, 不是升级 nfs-common 就是在升级挂载之前卸下全部 NFS 挂载。

罗马尼亚(ro)键盘布局的改变

由于在 lenny 中升级到 xkb-data 版本 1.3 默认的罗马尼亚(ro)布局变体现在可以产生正确的 şţ 字符(注意下面的逗号)而不是 şț (注意下面的下加符)。还有一些变体已经改名。旧的变体名字仍然可以工作, 不过建议用户更新他们的 /etc/X11/xorg.conf。因为这个改变更多信息以及可能导致的副作用在 wiki (仅罗马尼亚语) [http://wiki.debian.org/L10N/Romanian/Lenny/Notes]。

升级 apache2

Apache2 默认配置在一些方面有所改变, 这需要手工改变您的配置。最重要的改变有:

NameVirtualHost * 被改为 NameVirtualHost *:80。如果您加入了更多基于名称的虚拟主机, 您需要依次将 <VirtualHost *> 修改为 <VirtualHost *:80>。

Apache User 和 Group 以及 PidFile 路径现在在 /etc/apache2/envvars 中进行配置。如果您从这些设置的默认值进行了修改, 您需要修改那个文件。这也意味着您无法通过 **apache2 -k start** 启动 apache2, 您需要使用 **/etc/init.d/apache2** 或者 **apache2ctl**。

Mod_suexec 所需要的 suexec 帮助程序现在通过单独的软件包 apache2-suexec 来发行, 此软件包不会默认安装。

更多模块相关的配置从 /etc/apache2/apache2.conf 移到了 /etc/apache2/mods-available/*.conf。

要获取更多详细信息, 请参见 /usr/share/doc/apache2.2-common/NEWS.Debian.gz 和 /usr/share/doc/apache2.2-common/README.Debian.gz。

NIS 和网络管理器

在 lenny 中, nis 内的 ypbind 支持网络管理器。当网络管理器报告电脑从网络断开时, 它会导致 ypbind 禁用 NIS 客户端功能。因为网络管理器通常会在不使用网络时发出此报告, NIS 客户系统的 NIS 用户应确认禁用这些系统上的网络管理器支持。

要做到这一点可以通过卸载 network-manager 软件包, 或者编辑 /etc/default/nis 并添加 -no-dbus 到 YPBINDARGS。

新安装的 Debian 会默认使用 -no-dbus, 而之前的发行版本不是默认的。

Mozilla 产品的安全状态

Mozilla 程序 firefox、thunderbird 和 sunbird (在 Debian 中分别更名为 iceweasel、icedove、iceowl), 对许多用户来说是重要的工具。不幸的是上游的安全策略是主张用户更新到新版本, 而这与 Debian 的“不在安全更新中发行大的功能改变”的策略相冲突。我们今天无法预测, 但在 lenny 的生命周期中 Debian 安全团队可能会遇到无法继续支持 Mozilla 产品, 并宣布对 Mozilla 产品的安全支持的终止。在部署 Mozilla 时, 如果缺乏安全更新对您来说是一个问题, 您需要考虑到这一点并可以考虑 Debian 中的替代品。

iceape, 去商标化的 seamonkey 网络套件从 lenny 中被移除(除了一些内部库软件包)。

Security status of OCS Inventory and SQL-Ledger

`ocsinventory-server` and `sql-ledger` are included in the lenny release but have special security requirements that users should be aware of before deploying them. These two webservices are designed for deployment only behind an authenticated HTTP zone and should never be made available to untrusted users; and therefore they receive only limited security support from the Debian security team. Users should therefore take particular care when evaluating who to grant access to these services.

KDE 桌面

相比 `etch` 中发行的版本，KDE 桌面环境没有太大的变化。Lenny 发行 KDE 3.5 的翻译和服务更新的版本是 3.5.9 和 3.5.10 的混合。有些模块被标记为 3.5.9，但已经被更新并包含 3.5.10 版本大部分相同的改变。总的来说，Lenny 发行了 3.5.10，而并没有包含 `kdebase` 的重大改进和一些 `kdepim` 中的 `bug` 修正。

Lenny 将会是最后一个包含 KDE 3 系列环境的稳定发行。

GNOME 桌面改变与支持

lenny 版本中的 GNOME 桌面环境相比 `etch` 中发行的版本有大量的变化，您可以在 GNOME 2.22 发行说明 [<http://library.gnome.org/misc/release-notes/2.22/>] 找到更多信息。

Emacs21* 默认不支持 Unicode

`Emacs21`

Emacs21 and emacs21-nox are not configured to use Unicode by default. For more information and a workaround please see bug #419490 [<http://bugs.debian.org/419490>]. Consider switching to `emacs22-gtk`, `emacs22-gtk`, or `emacs22-nox`.

slurpd/replica 已经被删除

OpenLDAP 从版本 2.4.7 开始，不再支持通过 slurpd 服务执行 LDAP 复制。原先的配置需要修改为使用 LDAP 同步复制引擎(syncrepl)。更详细的文档位于 <http://www.openldap.org/doc/admin24/replication.html>。

桌面没有使用全部屏幕

Intel Mobile GM965 的驱动会错误地把输出类型检测为 VGA，并且设置一个较低的屏幕分辨率以适应输出。这个 `bug` 的典型特征为桌面管理器仅使用屏幕的一部分。添加如下的代码到配置文件 `/etc/X11/xorg.conf` 能纠正驱动的行为。

```
Section "Monitor"
    Identifier "VGA"
    Option "Ignore" "true"
EndSection
```

Please refer to the bug #496169 [<http://bugs.debian.org/496169>] for more informations.

DHCP 失败备选问题

当运行 DHCP 服务器故障转移时，节点名称要一致，否则 DHCP 会崩溃。

Please see bug #513506 [<http://bugs.debian.org/513506>] and <https://lists.isc.org/pipermail/dhcp-users/2007-September/004538.html> for more information.

VServer Disk Limit

To use the disk limit feature of vservers in lenny, you should use the **mount** option tag (instead of tagxid in etch).

You should manually update `/etc/fstab` and/or any script which uses tagxid. Otherwise, the partition will not be mounted and thus the vservers will not

第 6 章 关于 **Debian GNU/Linux** 的 更多信息 更多读物

除了本发行说明和安装指南外，Debian 文档项目 (DDP) 有更多关于 Debian GNU/Linux 的文档，该项目的目标是为 Debian 用户和开发者创建高质量的文档。它包括 Debian 参考手册，Debian 新维护人员指南和 Debian 常见问题等更多的文档。请查阅 DDP 网站 [<http://www.debian.org/doc/ddp>]，以获得现有资源的全部详细资料。

各个软件包的文档被安装到 `/usr/share/doc/package`。它包括版权信息，特定于 Debian 的信息，以及上游的文档。

获得帮助

Debian 的用户可以从很多渠道获得帮助、建议和支持，但您只应该在查阅所有的文档后，却依然找不到问题的答案时，才考虑寻求帮助。本节简单介绍了对 Debian 新用户可能会有帮助的渠道。

邮件列表

邮件列表 `debian-user` (英语) 和其它 `debian-user-language` (其它语言) 中有 Debian 用户最感兴趣的内容。想了解这些邮件列表的详细信息，以及如何订阅，请参阅读 <http://lists.debian.org/>。发问前请先搜索答案，并注意遵守列表有关的礼仪和准则。

互联网在线聊天系统

Debian 在 OFTC IRC 网络上有专门对 Debian 用户提供帮助的 IRC 频道。使用您喜欢的 IRC 客户端连接 `irc.debian.org`，加入 `#debian` 频道即可。

请遵守频道的准则，尊敬其他用户。可以在 Debian 维基 [<http://wiki.debian.org/DebianIRC>]中找到该准则。

请访问网站 [<http://www.oftc.net/>]，以获得更多关于 OFTC 的信息。

报告 **Bug**

我们努力让 Debian GNU/Linux 成为一款高质量的操作系统，但这并不意味着我们提供的软件包完全没有任何 Bug。为了和 Debian 一贯的“开放的开发”原则，以及为我们的用户提供更好的服务相协调，我们在 Bug 追踪系统(BTS)提供全部已经报告的 Bug 的所有信息。您可以通过 <http://bugs.debian.org/> 来访问 BTS。

如果您在本发行版或者软件包中发现了 Bug，请报告它，从而可以在将来的发行中被修复。您需要一个有效的电子邮件地址才能报告 Bug。有这个限制是因为这样我们能够追踪 Bug，以及在开发人员需要更多信息时，能购与提交人联系。

您可以使用程序 `reportbug` 来提交一个 Bug 报告，或者亲自动手发送邮件。可以通过阅读参考文档(如果您安装了 `doc-debian` 包的话，在 `/usr/share/doc/debian` 可以找到)或者在线的 Bug 追踪系统 [<http://bugs.debian.org/>]，以获得更多关于 Bug 追踪系统的信息和用法。

为 **Debian** 做贡献

并不是只有高手才能对 Debian 做出贡献。当您在各种支持用户的邮件列表 [<http://lists.debian.org/>]中帮助其他用户解决问题时，您就是在对整个社区做贡献。参与开发的邮件列表 [<http://lists.debian.org/>]以帮助开发者发现(以及解决)和系统发行相关的问题，对我们也是极大的帮助。为了维持 Debian 系统一贯的高品质，请提交 Bug [<http://bugs.debian.org/>]

bugs.debian.org/], 开协助开发人员跟踪和修复它们。如果您有文才万面的大赋, 您也可以通过创作文档 [<http://www.debian.org/doc/ddp>], 或者翻译 [<http://www.debian.org/international/>]现有文档到您自己的语言, 从而做出更直接的贡献。

如果您能投入更多的时间的话, 您可以负责维护 Debian 发行版中的部分自由软件。如果您能够新增或维护其他用户希望 Debian 所能包含的软件包的话就更好了, 可以在急需人手和被期待的软件包库 (WNPP) [[http://www.debian.org/devel/wnpp/](http://www.debian.org/devel/wnpp)] 中获得相关的信息。如果您在某些特别的领域有兴趣, 那么您可能会很乐意参加向某个特定硬件平台移植的子项目, 以及 Debian Jr. [<http://www.debian.org/devel/debian-jr/>] 和 Debian Med [<http://www.debian.org/devel/debian-med/>] 等项目。

无论如何, 只要您以任何方式参加了自由软件社区的活动, 不管您是用户, 程序员, 作者, 还是译者, 您就已经为自由软件社区做出了贡献。贡献是非常有益和有趣的, 不仅让您能够结交更多的新伙伴, 还能让您内心充满温暖的感觉。

附录 A. 管理您的 **etch** 系统

本附录包含在升级到 **lenny** 之前，弄清楚您怎样能够从 **etch** 安装或升级软件包的相关信息。这应该只在特殊的情况下需要。

升级您的 **etch** 系统

基本上这和您已经做过的从 **etch** 升级没有什么区别。唯一的区别就是，您首先需要确定您的软件包列表仍旧引用像在“检查您的 **sources list**”一节说明的 **etch** 中的软件包。

如果您从 Debian 镜像升级，将自动升级到最新的 **etch**。

检查您的 **sources list**

如果在 `/etc/apt/sources.list` 的任何行都引用 `'stable'`，您实际上已经在“使用”**lenny**。如果您已经运行了 `apt-get update`，您仍然可以按照下面的过程恢复而不会出现问题。

如果您已经从 **lenny** 安装了软件包，就没有太大的必要从 **etch** 安装软件包了。在这种情况下，您必须决定是否继续或者终止。可以降级软件包，但是这里不会提及。

用您喜欢的编辑器(以 **root** 权限)打开文件 `/etc/apt/sources.list`，检查所有以 `deb http:` 或 `deb ftp:` 开头，引用“**stable**”的行。如果找到了，将 **stable** 改为 **etch**。

如果有些行以 `deb file:` 开始，您应该自己检查该位置是否包含有 **etch** 或者 **lenny** 的档案。

重要

不要修改任何以 `deb cdrom:` 开始的行。如果这样做了的话，这些行就会失效，您将需要重新运行 `apt-cdrom`。即使在 `'cdrom'` 的行发现“**unstable**”，也不要感到奇怪。虽然令人困惑，但这是正常的。

如果做了任何修改，保存文件后执行

```
# apt-get update
```

刷新软件包列表。

附录 B. 发行注记的贡献者

有许多人对发行注记提供了帮助，包括，但不限于

Adam Di Carlo, Andreas Barth, Frans Pop, Anne Bezemer, Bob Hilliard, Charles Plessy, Christian Perrier, Daniel Baumann, Emmanuel Petrișor, Emmanuel Kasper, Esko Arajärvi, Frans Pop, Giovanni Rapagnani, Gordon Farquharson, Javier Fernández-Sanguino Peña, Jens Seidel, Jonas Meurer, Josip Rodin, Justin B Rye, LaMont Jones, Luk Claes, Martin Michlmayr, Michael Biebl, Moritz Mühlenhoff, Noah Meyerhans, Noritada Kobayashi, Osamu Aoki, Peter Green, Rob Bradford, Samuel Thibault, Simon Bienlein, Simon Paillard, Stefan Fritsch, Steve Langasek, Tobias Scherer, Vincent McIntyre, 和 W. Martin Borraert。

本文档被翻译为多种语言。感谢这些翻译者们！

附录 C. Lenny 献给 Thiemo Seufer

Debian 项目失去了一位积极的社区成员。Thiemo Seufer 在2008年12月26日的车祸中丧生。

Thiemo 用多种方式参与了 Debian 计划。他维护着多个软件包并是 MIPS 的 Debian 移植的主要支持者。他还是我们内核组及 Debian 安装程序组的成员。他的贡献远远超出 Debian 计划。他还致力于 Linux 内核的 MIPS 移植以及 qemu 的 MIPS 仿真等工作，同时还参与多个小项目，这里难以一一述及。

我们将永远怀念 Thiemo 的工作、奉献以及他同其他人分享的广泛的技术知识与能力。Thiemo 的贡献不会被遗忘。Thiemo 工作的高标准我们永难企及。

为了尊敬他对 Debian 的贡献，项目将 Debian GNU/Linux 5.0 “Lenny” 发行献给 Thiemo。

索引

符号

蓝光, 2

视力障碍用户, 2

A

Abiword, 2

Apache, 2

B

BIND, 2

C

CD, 2

Cherokee, 2

Courier, 2

D

Dia, 2

DocBook XML, 1

DVD, 2

E

Ekiga, 2

Emdebian, 5

Evolution, 2

Exim, 2

F

Firefox, 2

G

Gaim, 2

GCC, 2

GIMP, 2

GNOME, 2

GNUcash, 2

GNUmeric, 2

K

KDE, 2

KOffice, 2

L

LILO, 14

Linux 标准基础, 2

LXDE, 2

M

Microsoft Windows, 2

Mozilla, 2, 3

MySQL, 2

N

Netwinder, 21
Network Manager, 3
NIS, 3

O

OCS Inventory, 4
OpenOffice.org, 2
OpenSSH, 2
OpenVZ, 5

P

packages

- apache, 20
- apache2, 20
- apache2-suexec, 3
- apt, 1, 4, 4, 5, 5, 8, 10, 10, 10, 10, 10, 10, 10, 10
- aptitude, 3, 9, 9, 10, 10, 10, 10, 10, 10, 10, 10
- base-config, 11
- bind, 20
- bind9, 20
- dblatex, 1
- debian-goodies, 9
- doc-debian, 1
- docbook-xsl, 1
- eeepc-acpi-scripts, 5
- emacs22, 4
- emacs22-gtk, 4
- emacs22-nox, 4
- exim, 20
- exim4, 20
- firefox, 3
- glibc, 1
- grub, 1
- hotplug, 11
- iceape, 3
- icedove, 2, 3
- iceowl, 3
- iceweasel, 2, 3
- initramfs-tools, 2, 13, 14, 14
- ixp4xx-microcode, 1
- kernel-package, 13
- libc6, 10
- lilo, 14, 1
- linux-image-*, 13
- linux-image-2.6-686, 13
- localepurge, 9
- locales, 10
- lxde, 5
- mount, 2
- netkit-inetd, 11
- network-manager, 3
- nfs-common, 2, 2, 2, 2, 2, 2
- nis, 3

- ocsinventory-server, 4
- php4, 20
- php5, 20
- popularity-contest, 9
- postgresql-7.4, 20
- postgresql-8.1, 20
- python2.3, 11
- release-notes, 1, 1
- rsyslog, 4, 4
- seamonkey, 3
- sql-ledger, 4
- squid, 1
- sunbird, 3
- sysklogd, 4
- thunderbird, 3
- udev, 13, 14, xx, 14, 14, 14, 1, 1, 1, 1
- util-linux, 2, 2, 2
- wireless-tools, 2
- wpa_supplicant, 1, 2, 2
- xfree86-common, 11
- xkb-data, 3
- xlibs, 11
- xmlroff, 1
- xserver-common, 11
- xsltproc, 1
- PHP, 2
- Pidgin, 2
- Postfix, 2
- PostgreSQL, 2

S

- SELinux, 4
- SQL-Ledger, 4

T

- Thunderbird, 2
- Tomcat, 2

U

- Unicode, 4

V

- virtualization, 5
- VServer, 5

W

- WPA, 1

X

- Xfce, 2

术语表

ACPI	高级配置和电源接口
ALSA	高级 Linux 声音架构
APM	高级电源管理
CD	光盘
CD-ROM	只读光盘
DHCP	动态主机配置协议
DNS	域名系统
DVD	数字通用光盘
GIMP	GNU 图像处理程序
GNU	GNU's Not Unix 的递归缩写
GPG	GNU 隐私保护
IDE	电子集成驱动器
LDAP	轻量级目录访问协议
LILO	Linux 加载器
LSB	Linux 标准基础
LVM	逻辑卷管理器
MTA	邮件传输代理
NFS	网络文件系统
NIC	网卡
NIS	网络信息服务
OSS	开放声音系统
RAID	独立冗余磁盘阵列
RPC	远程过程调用
SATA	串行高级技术附件
USB	通用串行总线
UUID	通用惟一标识符
VGA	视频图像阵列
WPA	Wi-Fi 安全网络访问