Linux培训系列

————第五讲

欢迎学习“ 编译源代码和管理软件包 ”，在本教程中，我们将向您演示如何从源代码编译程序、如何管理共享库以及如何使用 Red Hat 和 Debian 软件包管理系统。

内容基础，语言简短简洁

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
红联Linux论坛： www.linuxdiyf.com/bbs
下载:Linux电子书籍：
http://www.linux286.com/linux/linuxdzsj.htm

目录

共享库

共享库
静态可执行程序与动态可执行程序比较
动态链接相关性
动态装入器
ld.so.conf
ld.so.cache
ldconfig 技巧
LD_LIBRARY_PATH

从源代码编译应用程序

从源代码编译应用程序－介绍
下载
解包
列出压缩文档
解包 bzip2 压缩的压缩文档
bzip2 管道
bzip2 管道（续）
检查源代码
配置
使用配置
--prefix 选项
使用 --prefix
FHS 怎么样？
该配置了
该配置了（续）
config.cache
配置脚本和制作文件
制作文件介绍
调用 make
安装
make install
安装之后
好，完成了！
可能出现的问题
遗漏一些库
其它问题
其它问题（续）

软件包管理概念

软件包管理概念
软件包管理的缺点

rpm，Red Hat 软件包管理器

rpm，Red Hat 软件包管理器
安装 rpm
重新安装一个 rpm
强制安装一个 rpm
用 --nodeps 安装或除去
更新软件包
用 rpm -q 查询

用 rpm -ql 列出文件
用 rpm -qp 查询软件包
查询所有已安装的软件包
查找文件的所有者
显示相关性
验证软件包的完整性
验证已安装的软件包
配置 rpm

Debian 软件包管理

Debian 软件包管理－介绍 apt-get
模拟安装
软件包资源列表：apt-setup
从 apt-get 到 dselect
启动 dselect
使用 dselect 的 Select 方式
软件包状态
安装和配置（dpkg-reconfigure）
获取已安装软件包的状态
文件与其 .deb 之间的链接
查找要安装的软件包

Linux海量文章

海量Linux技术文章

共享库

共享库

发布时间 :2007-01-27 19:50:34

Linux 系统上有两类根本不同的 Linux 可执行程序。第一类是静态链接的可执行程序。静态可执行程序包含执行所需的所有函数 — 换句话说，它们是“ 完整的 ”。因为这一原因，静态可执行程序不依赖任何外部库就可以运行。

第二类是动态链接的可执行程序。我们将在下页讨论这一内容。

静态可执行程序与动态可执行程序比较

发布时间 :2007-01-27 19:51:11

我们可以用 ldd 命令来确定某一特定可执行程序是否为静态链接的：

```
# ldd /sbin/sln
        not a dynamic executable
```

“ not a dynamic executable ”是 ldd 说明 sln 是静态链接的一种方式。现在，让我们比较 sln 与其非静态同类 ln 的大小：

```
# ls -l /bin/ln /sbin/sln
-rwxr-xr-x    1 root     root        23000 Jan 14 00:36 /bin/ln
-rwxr-xr-x    1 root     root       381072 Jan 14 00:31 /sbin/sln
```

如您所见，sln 的大小超过 ln 十倍。ln 比 sln 小这么多是因为它是动态可执行程序。动态可执行程序是不完整的程序，它依靠外部共享库来提供运行所需的许多函数。

动态链接相关性

发布时间 :2007-01-27 19:51:45

要查看 ln 依赖的所有共享库的列表，可以使用 ldd 命令：

```
# ldd /bin/ln
        libc.so.6 => /lib/libc.so.6 (0x40021000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

如您所见，ln 依赖外部共享库 libc.so.6 和 ld-linux.so.2。通常，动态链接的程序比其静态链接的等价程序小得多。不过，静态链接的程序可以在某些低级维护任务中发挥作用。例如，sln 是修改位于 /lib 中的不同库符号链接的极佳工具。但通常您会发现几乎所有 Linux 系统上的可执行程序都是某种动态链接的变体。

动态装入器

发布时间 :2007-01-27 19:52:19

那么，如果动态可执行程序不包含运行所需的所有函数，Linux 的哪部分负责将这些程序和所有必需的共享库一起装入，以使它们能正确执行呢？答案是动态装入器（dynamic loader），它实际上是您在 ln 的 ldd 清单中看到的作为共享库相关性列出的 ld-linux.so.2 库。动态装入器负责装入动态链接的可执行程序运行所需的共享库。现在，让我们迅速查看一下动态装入器如何在系统上找到适当的共享库。

ld.so.conf

发布时间 :2007-01-27 19:52:54

动态装入器找到共享库要依靠两个文件 — /etc/ld.so.conf 和 /etc/ld.so.cache。如果您对 /etc/ld.so.conf 文件进行 cat 操作，您可能会看到一个与下面类似的清单：

```
$ cat /etc/ld.so.conf
/usr/X11R6/lib
/usr/lib/gcc-lib/i686-pc-linux-gnu/2.95.3
/usr/lib/mozilla
/usr/lib/qt-x11-2.3.1/lib
/usr/local/lib
```

ld.so.conf 文件包含一个所有目录（/lib 和 /usr/lib 除外，它们会自动包含在其中）的清单，动态装入器将在其中查找共享库。

ld.so.cache

发布时间 :2007-01-27 19:53:29

但是在动态装入器能“ 看到 ”这一信息之前，必须将它转换到 ld.so.cache 文件中。可以通过运行 ldconfig 命令做到这一点：

ldconfig

当 ldconfig 操作结束时，您会有一个最新的 /etc/ld.so.cache 文件，它反映您对 /etc/ld.so.conf 所做的更改。从这一刻起，动态装入器在寻找共享库时会查看您在 /etc/ld.so.conf 中指定的所有新目录。

ldconfig 技巧

发布时间 :2007-01-27 19:54:03

要查看 ldconfig 可以“ 看到 ”的所有共享库，请输入：

ldconfig -p | less

还有另一个方便的技巧可以用来配置共享库路径。有时候您希望告诉动态装入器在尝试任何 /etc/ld.so.conf 路径以前先尝试使用特定目录中的共享库。在您运行的较旧的应用程序不能与当前安装的库版本一起工作的情况下，这会比较方便。

LD_LIBRARY_PATH

发布时间 :2007-01-27 19:54:39

要指示动态装入器首先检查某个目录，请将 LD_LIBRARY_PATH 变量设置成您希望搜索的目录。多个路径之间用逗号分隔；例如：

export LD_LIBRARY_PATH="/usr/lib/old:/opt/lib"

导出 LD_LIBRARY_PATH 后，如有可能，所有从当前 shell 启动的可执行程序都将使用 /usr/lib/old 或 /opt/lib 中的库，如果仍不能满足一些共享库相关性要求，则转回到 /etc/ld.so.conf 中指定的库。

我们已经完成了对 Linux 共享库的介绍。要了解有关共享库的更多内容，请输入 man ldconfig 和 man ld.so。

从源代码编译应用程序

从源代码编译应用程序－介绍

发布时间 :2007-01-27 19:55:28

假设您发现了一个特定的应用程序并想将它安装在系统上。您可能需要运行这个程序的最新版本，但还没有这个最新版本的打包格式（如 rpm）。或许您只能得到这个特定应用程序的源代码格式，或者您需要启用缺省情况下在 rpm 中未启用的该程序的某些特性。

无论什么原因，不管您必须还是仅仅因为您想从源代码编译该程序，本节将向您演示如何做。

下载

发布时间 :2007-01-27 19:55:58

首先，要找到并下载您要编译的源代码。它们可能在以 .tar.gz、tar.Z、tar.bz2 或 .tgz 扩展名结尾的单个压缩文档中。继续进行，用您喜爱的浏览器或 ftp 程序下载这个压缩文档。如果这个程序碰巧有一个网页，那么最好访问该网页以熟悉所有可能有用的安装文档。

您正在安装的程序可能依赖于许多当前在系统上已安装或未安装的其它程序。如果您确切知道该程序依赖的其它程序或库当前没有安装在系统上，则您需要先安装这些软件包（从象 rpm 那样的二进制软件包安装或同样从它们的源代码进行编译）。这样，您将为成功安装原始源代码文件做好准备。

解包

发布时间 :2007-01-27 19:56:27

解包源压缩文档相对较简单。如果压缩文档名称以 .tar.gz、.tar.Z 或 .tgz 结尾，您应该可以通过输入以下内容来解包：

```
$ tar xzvf archivename.tar.gz
```

（x 用于解压缩，z 用于 gzip 解压，v 用于显示详细信息 — 打印解压缩的文件名，而 f 意味着文件名将接着在命令行上出现）。

几乎所有的“ 源代码 tar 包（tarball） ”都将创建一个包含程序所有源代码的主目录。这样，当您解包这个压缩文档时，您的当前工作目录不会被大量的文件搞得乱七八糟 — 相反，所有的文件被整齐地组织在单个目录中，不会妨碍工作。

列出压缩文档

发布时间 :2007-01-27 19:57:13

时不时您会遇到这样的压缩文档，在对它解压缩时，会在您的当前工作目录中创建数量巨大的文件。尽管大多数 tar 包不是这样创建的，但这样的情况确有发生。如果您希望验证您的 tar 包被正确地组装在一起并会创建一个主目录来容纳源代码，您可以输入以下内容来查看其内容：

$ tar tzvf archivename.tar.gz | more

（t 用于显示压缩文档的文本清单。不进行解压缩）。

如果压缩文档清单的左边没有列出公共目录，则您要创建一个新的目录，将 tar 包移至该目录下，进入该目录，这时再解压缩这个 tar 包。不这样做的话，您会面对一团乱麻！

解包 bzip2 压缩的压缩文档

发布时间 :2007-01-27 19:57:47

您的压缩文档有可能是 .tar.bz2 格式。具有这种扩展名的压缩文档是用 bzip2 进行压缩的。bzip2 的压缩效果通常比 gzip 好得多。它唯一的不足之处是压缩和解压缩的速度较慢，并且在运行时， bzip2 比 gzip 消耗更多的内存。对于现代计算机，这不成问题，因此，可以预料 bzip2 会随着时间的推移变得越来越流行。

因为 bzip2 日益受到欢迎，所以许多 Linux 分发版（distribution）都带有经过补丁程序修正的 tar 版本，这样传递一个 y 或 i 选项将通知 tar：压缩文档是 bzip2 格式，需要用 bzip2 程序自动解压缩。要查看您是否带有用补丁修正过的 tar 版本，可以试着输入：

```
$ tar tyvf archive.tar.bz2 | more
```

或

```
$ tar tivf archive.tar.bz2 | more
```

即使两个命令都不起作用（而且 tar 提示参数无效），仍有办法 — 请继续阅读。

bzip2 管道

发布时间 :2007-01-27 19:58:24

哦，您的 tar 版本不能识别这些方便的 bzip2 快捷方式 — 该怎么办呢？所幸有一个简单的方法可以解压缩 bzip2 tar 包的内容，并且这个方法几乎可以在所有的 UNIX 系统上工作，即使正被讨论的系统碰巧有一个非 GNU 版本的 tar。要查看 bzip2 文件的内容，我们可以创建一个管道：

$ cat archive.tar.bz2 | bzip2 -d | tar tvf - | most

接下来的这个管道实际上将解压缩 archive.tar.bz2 的内容：

$ cat archive.tar.bz2 | bzip2 -d | tar xvf -

bzip2 管道（续）

发布时间 :2007-01-27 19:58:57

在前两个示例中，我们创建了一个标准的 UNIX 管道从压缩文档查看并解压缩文件。因为用了 f - 选项来调用 tar，因此它从标准输入（stdin）读取 tar 数据，而不是试图从磁盘上的文件读取数据。

如果尝试用管道方法解压缩您的压缩文档的内容，而系统提示找不到 bzip2，则系统可能没有安装 bzip2。可以从 http://sourceware.cygnus.com/bzip2 下载 bzip2 的源代码。安装 bzip2 源代码之后（通过遵照本教程），您才能首先解包并安装您希望安装的应用程序

检查源代码

发布时间 :2007-01-27 19:59:28

解包源代码之后，您可以进入解包的目录并检查其中的内容。最好是能找到所有与安装有关的文档。通常，这一信息可以在位于主源代码目录的 README 或 INSTALL 文件中找到。另外，可以查找 README.platform 和 INSTALL.platform 文件，这里的 platform 是您的特定操作系统名称。

配置

发布时间 :2007-01-27 19:59:55

现在，许多源代码在主源代码目录中包含配置脚本。这个脚本（通常由开发人员使用 GNU autoconf 程序生成）特别设计用来设置源代码以使它们能在您的系统上正确编译。这个配置脚本在运行时会探测您的系统以确定系统性能，然后创建制作文件（Makefile），其中包含在您的系统上构建和安装源代码的指令。

这个配置脚本几乎总是被命名为“ configure ”。如果您在主源代码目录中找到配置脚本，那么它可以供您使用。如果您没有发现配置脚本，那么您的源代码可能带有一个设计用来跨越不同系统工作的标准制作文件 — 这意味着您可以略过以下配置步骤，在我们开始讨论“ make ”处继续学习本教程。

使用配置

发布时间 :2007-01-27 20:00:28

在运行配置脚本之前，最好是先熟悉它。输入 ./configure --help，您可以查看您的程序能够使用的所有不同配置选项。您所看到的选项，特别是在 --help 打印输出的顶部列出的那些项，都是几乎可以在每个配置脚本中找到的标准选项。在结尾部分列出的选项通常与您正尝试编译的特定软件包相关。查看它们并注意那些您希望启用或禁用的选项。

--prefix 选项

发布时间 :2007-01-27 20:00:57

大多数基于 GNU autoconf 的配置脚本都有 --prefix 选项来允许您控制程序的安装位置。缺省情况下，大多数源代码安装时都用 /usr/local 前缀。这意味着二进制文件最终在 /usr/local/bin 中，手册页则在 /usr/local/man 中，等等。这通常就是您所期望的； /usr/local 通常用于存储您自己编译的程序。

使用 --prefix

发布时间 :2007-01-27 20:01:27

如果您想把源代码安装在别的地方，假设安装在 ./usr 中，则您要向配置脚本传递 --prefix=/usr 选项。同样地，您也可以用 --prefix=/opt 选项告诉配置脚本将源代码安装到 /opt 目录。

FHS 怎么样？

发布时间 :2007-01-27 20:02:01

有时候，一个特定程序可能缺省地将它的一些文件安装到磁盘上的非标准位置。特别地，一个源代码压缩文档可能有许多没有遵守 Linux 文件系统层次结构标准（FHS）的安装路径。幸运的是，配置脚本不仅允许更改安装前缀，而且允许我们更改各个系统部件（如手册页）的安装位置。

因为大多数源代码压缩文档还不符合 FHS，所以这种能力非常有用。为了使您的源代码包符合 FHS，您几乎总要向配置命令行添加 --mandir=/usr/share/man 和 --infodir=/usr/share/info。

该配置了

发布时间 :2007-01-27 20:02:36

一旦已经检查了各种配置选项并确定了要使用的选项，就可以运行配置脚本了。请注意您可能不需要在运行配置脚本时包含所有命令行选项 — 在大多数情形下，缺省值可以发挥作用（但可能不完全是您想要的）。

该配置了（续）

发布时间 :2007-01-27 20:03:07

要运行配置脚本，请输入：

$ ./configure <options>

这看起来象：

$ ./configure

或

$ ./configure --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-threads

您需要的选项取决于您正在配置的特定软件包。当您运行配置脚本时，它需要一两分钟来探测您系统上可用的特定功能部件或工具，并随着它工作的进行打印出各种配置检查的结果。

config.cache

发布时间 :2007-01-27 20:03:41

配置过程完成后，配置脚本将它所有的配置数据存储在一个名为 config.cache 的文件中。这个文件和配置脚本本身都在同一目录中。如果您在更新系统配置后需要再次运行 ./configure，请确保您先执行 rm config.cache 命令；否则配置脚本将只使用旧的设置而不重新检查系统。

配置脚本和制作文件

发布时间 :2007-01-27 20:04:13

配置脚本完成之后，就是将源代码编译为可运行程序的时候了。可以用名为 make 的程序来执行这一步骤。如果您的软件包含配置脚本，那么当您运行它时，配置脚本会创建一些为您的系统特别定制的制作文件。这些文件会告诉 make 程序如何构建源代码以及如何安装所产生的二进制文件、手册页和支持文件。

制作文件介绍

发布时间 :2007-01-27 20:04:43

Makefile 通常被称为 makefile 或 Makefile。每个含有源文件的目录通常都有一个制作文件，另外主源代码目录中还有一个制作文件。autoconf 生成的制作文件含有确定如何构建某些目标（如您希望安装的程序）的指示信息（正式名称是规则）。make 指出所有规则运行的顺序。

调用 make

发布时间 :2007-01-27 20:05:16

调用 make 比较简单；只需在当前目录中输入“ make ”即可。make 程序随即将在当前目录中查找并解释名为 makefile 或 Makefile 的文件。如果您只输入“ make ”本身，那么它将构建缺省目标。开发人员通常设置制作文件以使缺省目标编译所有源代码：

$ make

有些 makefile 没有缺省目标，因此您要指定一个以使编译得以开始：

$ make all

输入这些命令的其中之一后，您的计算机将用几分钟的时间将您的程序编译成目标代码。假设它不出差错地完成编译，您就可以准备将已编译程序安装到系统上。

安装

发布时间 :2007-01-27 20:05:46

编译好程序之后，还有一个更重要的步骤：安装。尽管程序已编译，但它仍未做好使用准备。您需要将它的所有组件从源代码目录复制到您的文件系统上适当的、“ 活动的 ”位置。例如，您需要将所有的二进制文件复制到 /usr/local/bin，将所有的手册页安装到 /usr/local/man，等等。

在安装软件以前，您要成为 root 用户。实现这一点通常有两种方式，在另一台终端以 root 用户登录，或者输入“ su ”，那时就会提示您输入 root 用户的密码。输入密码以后，您就将一直拥有 root 用户的特权，直到您输入“ exit ”或按 control-D 键组合从当前 shell 会话退出为止。如果您已经是 root 用户，那就可以进行下一步行动了！

make install

发布时间 :2007-01-27 20:06:16

安装源代码比较简单。只需在主源代码目录中输入：

make install

输入“ make install ”告诉 make 要满足“ install ”目标；这个目标通常用来将所有新近创建的源文件复制到磁盘上的正确位置，以便可以使用您的程序。如果您没有指定 --prefix 选项，则很有可能有相当多的文件和目录将被复制到 /usr/local 目录。根据程序大小的不同，安装目标需要的时间可能从几秒到几分钟不等。

除了简单地复制文件以外，make install 还会确保安装的文件有正确的所有权和许可权。make install 成功完成后，程序就安装完毕并且可以使用（或几乎可以使用）。

安装之后

发布时间 :2007-01-27 20:06:50

现在您的程序已经安装完毕，那么下一步是什么？当然是运行它啦！如果您不熟悉如何使用刚刚安装的程序，则可以通过输入以下命令来阅读这个程序的手册页：

$ man programname

程序可能需要另外的配置步骤。例如，如果您安装了 Web 服务器，则您需要将它配置成在系统引导时自动启动。在应用程序运行以前，您可能还需要在 /etc 中定制一个配置文件。

好，完成了！

发布时间 :2007-01-27 20:07:21

既然您已经从源代码完整地安装了一个特定的软件包，现在可以运行它了！要启动这个程序，请输入：

$ programname

祝贺您！

可能出现的问题

发布时间 :2007-01-27 20:07:49

configure 或 make（甚至可能是 make install）很有可能会因某种错误代码而异常终止。以下几页将帮助您改正一些常见问题。

遗漏一些库

发布时间 :2007-01-27 20:08:21

您可能时常会有 configure 完全失败的经历，这往往是因为某个库没有安装。为了能使构建过程继续进行，您需要暂停当前的程序配置，搜寻源代码或二进制软件包以找到程序所需的库。当安装正确的库后，configure 或 make 应该能顺利运行并成功地完成。

其它问题

发布时间 :2007-01-27 20:08:51

有时候，您会碰到某种不知如何修复的错误。随着您的 UNIX／Linux 经验的增长，您将能够诊断越来越多在 configure 和 make 过程中遇到的、看似神秘的错误情况。

有时，错误的出现是因为安装的库太旧（甚至也可能是因为太新了！）。还有些时候，碰到的问题实际上是开发人员的过错，他们可能没有预计到他们的程序会在您这样的系统上运行 — 或者他们只是打字有误

其它问题（续）

发布时间 :2007-01-27 20:09:21

对于这样的问题，您要清楚能在哪里获得帮助。如果这是您首次尝试从源代码编译程序，那么最好还是选择另一个更简单的程序来编译。当您编译了较简单的程序后，您就会有修正最初遇到的问题所必需的经验。随着您继续学习更多有关 UNIX 如何工作的知识，您将更接近这样的境界：“ 微调 ”制作文件和源代码就可以让那些看上去甚至有些古怪的代码顺利地编译。

软件包管理概念

软件包管理概念

发布时间 :2007-01-27 20:10:09

除了从源代码构建应用程序以外，还有另一种在 Linux 系统上安装软件的方法。所有的 Linux 分发版都使用某种形式的软件包管理来安装、更新和卸载软件包。与直接从源代码安装相比，软件包管理有明显优势：

易于安装和卸载
易于更新已安装的软件包
保护配置文件
轻松跟踪已安装文件

软件包管理的缺点

发布时间 :2007-01-27 20:10:43

在讲解如何使用最流行的软件包管理工具以前，我得承认有些 Linux 用户不喜欢软件包管理。他们可能会提出软件包管理的以下缺点：

为特定系统构建的二进制文件性能更好
解决软件包相关性比较麻烦
软件包数据库的破坏会导致系统不可维护
创建软件包比较困难

这些说法确有其事，但 Linux 用户的一般看法是软件包管理的优势大于劣势。另外，上面列出的每个不利因素都有一个相应的反证：可以为不同的系统，构建多个优化的软件包；可以增强软件包管理器来自动解决相关性；可以基于其它文件重建数据库；而以后更新或除去这个软件包时的方便性可以弥补最初创建它时所做的努力。

rpm，Red Hat 软件包管理器

rpm，Red Hat 软件包管理器

发布时间 :2007-01-27 20:12:26

rpm，Red Hat 软件包管理器（(R)ed Hat (P)ackage (M)anager）

rpm 入门

对 Linux 分发版而言，1995 年 Red Hat 引入 rpm 是一个巨大的进步。它不仅使 Red Hat Linux 上的软件包管理成为可能，而且因为它的 GPL 许可证，rpm 已经成为开放源代码打包的事实标准。

尽管有 GUI 和基于 Web 的工具提供更友好的界面，rpm 程序在缺省情况下是一个命令行界面。在这一节中，我们将介绍最常用的命令行操作，并且将 Xsnow 程序作为示例使用。如果您愿意按照下面的步骤执行，可以在下面下载 rpm，它应该可以用于大多数基于 rpm 的分发版。

xsnow-1.41-1.i386.rpm

注：如果这一节中“ rpm ”一词的不同使用让您有些困惑的话，请记住“ rpm ”通常指程序，而“ 一个 rpm ”或“ 那个 rpm ”则通常指一个 rpm 软件包。

安装 rpm

发布时间 :2007-01-27 20:13:01

首先让我们使用 rpm -i 来安装 Xsnow rpm：

```
# rpm -i xsnow-1.41-1.i386.rpm
```

如果这个命令没有产生输出，那么它就起作用了！您应该能够运行 Xsnow 在您的 X 桌面上享受一场暴风雪。我们本身是希望在安装一个 rpm 时有某些可视化反馈的，所以我们可以加上 -h（用 # 号表示进展）和 -v（详细信息）选项：

```
# rpm -ivh xsnow-1.41-1.i386.rpm
xsnow                   ##########################################################
```

重新安装一个 rpm

发布时间 :2007-01-27 20:13:36

如果您直接按照步骤执行，您可能会在前一个示例中看到来自 rpm 的下列消息：

```
# rpm -ivh xsnow-1.41-1.i386.rpm
package xsnow-1.41-1 is already installed
```

在有些情况下，您可能希望重新安装一个 rpm，例如，您可能不小心删除了二进制文件 /usr/X11R6/bin/xsnow 。在那种情况下，您应该首先用 rpm -e 除去那个 rpm，然后重新安装它。请注意：下面的示例中来自 rpm 的资料信息不妨碍从系统除去软件包。

```
# rpm -e xsnow
removal of /usr/X11R6/bin/xsnow failed: No such file or directory
```

```
# rpm -ivh xsnow-1.41-1.i386.rpm
xsnow                   ##########################################################
```

强制安装一个 rpm

发布时间 :2007-01-27 20:14:10

有时除去一个 rpm 是不现实的，尤其是系统上有别的程序依赖于它的时候。例如，您可能已经安装了一个 “ x-amusements ” rpm，它将 Xsnow 列为相关程序，因此用 rpm -e 除去 Xsnow 是不允许的：

```
# rpm -e xsnow
error: removing these packages would break dependencies:
        /usr/X11R6/bin/xsnow is needed by x-amusements-1.0-1
```

在这种情况下，您可以用 --force 选项重新安装 Xsnow：

```
# rpm -ivh --force xsnow-1.41-1.i386.rpm
xsnow                   ########################################################
```

用 --nodeps 安装或除去

发布时间 :2007-01-27 20:14:46

上页中使用 --force 的一个替代方法是用 --nodeps 选项除去 rpm。这一选项取消了 rpm 的内部相关性检查，所以在大多数情况下不推荐使用。然而，它偶尔会有用处：

```
# rpm -e --nodeps xsnow
```

```
# rpm -ivh xsnow-1.41-1.i386.rpm
xsnow                   ##########################################################
```

在安装 rpm 时，您也可以使用 --nodeps。再次重申上面所说的，不推荐使用 --nodeps，但它有时是必需的：

```
# rpm -ivh --nodeps xsnow-1.41-1.i386.rpm
xsnow                   ##########################################################
```

更新软件包

发布时间 :2007-01-27 20:15:19

最终可能将有 Xsnow 版本 1.42 的 rpm，它可以在 Xsnow 作者的网站上获得。这时，您将要更新现有的 Xsnow 安装。如果您要使用 rpm -ivh --force，似乎会奏效，但 rpm 的内部数据库会列出两个版本都已安装。相反，您应该用 rpm -U 来更新您的安装：

```
# rpm -Uvh xsnow-1.42-1.i386.rpm
xsnow                   ###########################################################
```

这里有一个小窍门：我们几乎不使用 rpm -i，因为如果系统上不存在 rpm，则 rpm -U 会安装一个。如果您在命令行指定多个软件包，其中有些软件包当前已安装而有些则没有，那么这一选项会特别有用。

```
# rpm -Uvh xsnow-1.42-1.i386.rpm xfishtank-2.1tp-1.i386.rpm
xsnow                   ###########################################################
xfishtank               ###########################################################
```

用 rpm -q 查询

发布时间 :2007-01-27 20:16:02

您可能已经在前几个示例中注意到：安装 rpm 需要完整的文件名，但除去 rpm 只需要文件的名称即可。这是因为 rpm 维护一个当前已安装软件包的内部数据库，所以您可以通过名称来引用安装的软件包。例如，让我们查询 rpm 安装的是什么版本的 Xsnow：

```
# rpm -q xsnow
xsnow-1.41-1
```

事实上，rpm 知道的有关已安装软件包的信息远不止是名称和版本。我们可以用 rpm -qi 查询有关 Xsnow rpm 的更多信息：

```
# rpm -qi xsnow
Name        : xsnow                        Relocations: (not relocateable)
Version     : 1.41                           Vendor: Dan E. Anderson  http://www.dan.drydog.com/
Release     : 1                          Build Date: Thu 10 May 2001 01:12:26 AM EDT
Install date: Sat 02 Feb 2002 01:00:43 PM EST      Build Host: danx.drydog.com
Group       : Amusements/Graphics          Source RPM: xsnow-1.41-1.src.rpm
Size        : 91877                        License: Copyright 1984, 1988, 1990, 1993-1995, 2000 by Rick Jansen.
Allows packaging & necessary changes for Unix/Linux distributions.
Packager    : Dan E. Anderson  http://dan.drydog.com/
URL         :  http://www.euronet.nl/~rja/Xsnow/
Summary     : An X Window System based dose of Christmas cheer.
Descr ī ption :
The Xsnow toy provides a continual gentle snowfall, trees, and Santa
Claus flying his sleigh around the screen on the root window.
Xsnow is only for the X Window System, though; consoles just get coal.
```

用 rpm -ql 列出文件

发布时间 :2007-01-27 20:16:35

由 rpm 维护的数据库包含很多信息。我们已经看到它记录了已安装软件包的版本，以及它们的相关信息。还可以用 rpm -ql 列出给定的已安装软件包所拥有的文件：

```
# rpm -ql xsnow
/etc/X11/applnk/Games/xsnow.desktop
/usr/X11R6/bin/xsnow
/usr/X11R6/man/man1/xsnow.1x.gz
```

结合 -c 选项或 -d 选项，您可以将输出分别限制为配置文件或文档文件。这一类型的查询对有较长文件列表的较大 rpm 更有用，但我们仍然可以用 Xsnow rpm 来演示：

```
# rpm -qlc xsnow
/etc/X11/applnk/Games/xsnow.desktop
```

```
# rpm -qld xsnow
/usr/X11R6/man/man1/xsnow.1x.gz
```

用 rpm -qp 查询软件包

发布时间 :2007-01-27 20:17:11

如果您在安装软件包之前已经有了用 rpm -qi 获得的信息，您也许就能够更好地决定是否安装这个软件包。实际上，使用 rpm -qp 允许您查询一个 rpm 文件而不是查询数据库。到目前为止我们见到的所有查询命令都可以用于 rpm 文件和已安装的软件包。下面再次列出所有的示例，这次使用的是 -p 选项：

```
# rpm -qp xsnow-1.41-1.i386.rpm
xsnow-1.41-1

# rpm -qpi xsnow-1.41-1.i386.rpm
[与上页中的 rpm -qi 有相同的输出]

# rpm -qpl xsnow-1.41-1.i386.rpm
/etc/X11/applnk/Games/xsnow.desktop
/usr/X11R6/bin/xsnow
/usr/X11R6/man/man1/xsnow.1x.gz

# rpm -qplc xsnow-1.41-1.i386.rpm
/etc/X11/applnk/Games/xsnow.desktop

# rpm -qpld xsnow-1.41-1.i386.rpm
/usr/X11R6/man/man1/xsnow.1x.gz
```

查询所有已安装的软件包

发布时间 :2007-01-27 20:17:47

您可以加上 -a 选项来查询安装在系统上的所有软件包。如果您用管道命令将输出通过 sort 传入一个分页程序，则可一览系统上安装的软件包。例如：

```
# rpm -qa | sort | less
[省略输出]
```

以下是在我们的一个系统上安装的 rpm 数目：

```
# rpm -qa | wc -l
   287
```

而以下是所有这些 rpm 中的文件总数：

```
# rpm -qal | wc -l
  45706
```

这里有一个简单窍门：使用 rpm -qa 可以简化多个系统的管理。如果您在一台机器上将排序的输出重定向至一个文件，然后在另一台机器上做同样操作，您可以用 diff 程序来观察二者的区别。

查找文件的所有者

发布时间 :2007-01-27 20:18:31

有时找出什么 rpm 拥有给定的文件是非常有用的。理论上，您可以用与下面类似的 shell 构造了解哪个 rpm 拥有 /usr/X11R6/bin/xsnow（假装您不记得了）：

```
# rpm -qa | while read p; do rpm -ql $p | grep -q '^/usr/X11R6/bin/xsnow$' && echo $p; done
xsnow-1.41-1
```

因为输入这些要很长的时间，而运行的时间更长（在我们的一台奔腾机上要 1 分 50 秒），rpm 开发人员很聪明地把这种功能包含进了 rpm。您可以用 rpm -qf 查询一个特定文件的所有者：

```
# rpm -qf /usr/X11R6/bin/xsnow
xsnow-1.41-1
```

即使在奔腾机上，这也只要 0.3 秒的时间运行。而且，与复杂的 shell 构造相比，连打字高手也会喜欢 rpm -qf 的简单。

显示相关性

发布时间 :2007-01-27 20:19:10

除非您使用如 --nodeps 那样的选项，rpm 通常不允许您安装或除去那些会破坏相关性的软件包。例如，您不能在没有 X 库的系统上安装 Xsnow。安装 Xsnow 之后，您不能在没有除去 Xsnow（而且可能还有半数您已安装的软件包）的情况下除去 X 库。

这就是 rpm 的强项，这一点有时甚至令人头疼。它意味着只要您安装了 rpm，它就应该正常工作。您无需做额外的工作，因为 rpm 已经验证了系统上存在相关性。

有时候，在您解决相关性问题时，用 -R 选项查询一个软件包有助于了解这个软件包在系统上被期望具有的所有情况。例如，Xsnow 软件包依赖 C 库、math 库、X 库和 rpm 的特定版本：

```
# rpm -qpR xsnow-1.41-1.i386.rpm
rpmlib(PayloadFilesHavePrefix) <= 4.0-1
ld-linux.so.2
libX11.so.6
libXext.so.6
libXpm.so.4
libc.so.6
libm.so.6
libc.so.6(GLIBC_2.0)
libc.so.6(GLIBC_2.1.3)
rpmlib(CompressedFileNames) <= 3.0.4-1
```

您也可以省略 -p 来向已安装的数据库查询同样的信息：

```
# rpm -qR xsnow
```

验证软件包的完整性

发布时间 :2007-01-27 20:19:51

当您从 Web 或 ftp 站点下载 rpm 时，出于安全性考虑，您可能希望在安装该软件包之前，先验证它的完整性。所有的 rpm 都已使用 MD5 校验和 “ 签名 ” 。另外，有些作者使用 PGP 或 GPG 签名进一步加强他们的软件包的安全性。要查看软件包的签名，您可以使用 --checksig 选项：

```
# rpm --checksig xsnow-1.41-1.i386.rpm
xsnow-1.41-1.i386.rpm: md5 GPG NOT OK
```

稍等片刻！根据那个输出，GPG 签名为 NOT OK。让我们加一些详细说明看看出了什么问题：

```
# rpm --checksig -v xsnow-1.41-1.i386.rpm
xsnow-1.41-1.i386.rpm:
MD5 sum OK: 8ebe63b1dbe86ccd9eaf736a7aa56fd8
gpg: Signature made Thu 10 May 2001 01:16:27 AM EDT using DSA key ID B1F6E46C
gpg: Can't check signature: public key not found
```

那么，问题出在我们不能获取作者的公钥。在我们从软件包作者的网站获得公钥后，（如来自 rpm -qi 的输出所示），签名检出如下：

```
# gpg --import dan.asc
gpg: key B1F6E46C: public key imported
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: Total number processed: 1
gpg:               imported: 1
```

```
# rpm --checksig xsnow-1.41-1.i386.rpm
xsnow-1.41-1.i386.rpm: md5 gpg OK
```

验证已安装的软件包

发布时间 :2007-01-27 20:20:31

与检查 rpm 的完整性相似，您也可以用 rpm -V 检查已安装文件的完整性。这一步骤确保文件从 rpm 安装以后没有再被修改。

```
# rpm -V xsnow
```

通常这一命令没有输出，这表明文件正常。让我们把事情弄得更有趣味，然后再试一次：

```
# rm /usr/X11R6/man/man1/xsnow.1x.gz
```

```
# cp /bin/sh /usr/X11R6/bin/xsnow
```

```
# rpm -V xsnow
S.5....T   /usr/X11R6/bin/xsnow
missing    /usr/X11R6/man/man1/xsnow.1x.gz
```

这个输出向我们显示 Xsnow 二进制文件的 MD5 校验和、文件大小和 mtime 测试失效。而且手册页全部丢失！让我们修复这个损坏的安装：

```
# rpm -e xsnow
removal of /usr/X11R6/man/man1/xsnow.1x.gz failed: No such file or directory
```

```
# rpm -ivh xsnow-1.41-1.i386.rpm
xsnow                   ##########################################################
```

配置 rpm

发布时间 :2007-01-27 20:21:01

rpm 很少需要配置。它运行良好。在较旧的 rpm 版本中，您可以在 /etc/rpmrc 中做更改来影响运行时操作。在最近的版本中，这个文件被移到 /usr/lib/rpm/rpmrc，并且不打算让系统管理员编辑它。它通常只列出各个平台的标志和兼容性信息（例如，i386 与所有其它 x86 体系结构兼容）。

如果您希望配置 rpm，可以通过编辑 /etc/rpm/macros 实现。因为这很少有必要，我们将让您在 rpm 捆绑的文档中阅读它。您可以用下列命令找到合适的文档：

```
# rpm -qld rpm | grep macros
```

Debian 软件包管理

Debian 软件包管理－介绍 apt-get

发布时间 :2007-01-27 20:21:52

Debian 软件包管理系统由几个不同的工具组成。命令行工具 apt-get 是安装新软件包的最简单方法。例如，要安装程序 Xsnow，可以 root 用户的身份执行以下操作：

```
# apt-get install xsnow
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  xsnow
0 packages upgraded, 1 newly installed, 0 to remove and 10 not upgraded.
Need to get 17.8kB of archives. After unpacking 42.0kB will be used.
Get:1  http://ftp-mirror.internap.com stable/non-free xsnow 1.40-6 [17.8kB]
Fetched 17.8kB in 0s (18.4kB/s)
Selecting previously deselected package xsnow.
(Reading database ... 5702 files and directories currently installed.)
Unpacking xsnow (from .../archives/xsnow_1.40-6_i386.deb) ...
Setting up xsnow (1.40-6) ...
```

浏览这个输出，您可以看到即将安装 Xsnow，然后从 Web 上获取它，解包，最后设置。

模拟安装

发布时间 :2007-01-27 20:22:30

如果 apt-get 发觉您要安装的软件包依赖其它一些软件包，它会自动获取并安装这些软件包。在上一个示例中，只安装了 Xsnow，这是因为它的所有相关性都已满足。

然而，有时候 apt-get 需要获取的软件包列表会很大，所以一般最好在安装前先查看将要安装的文件。-s 选项正好做这一工作。例如，在我们的一个系统上，如果我们试图安装图形电子邮件程序 balsa：

```
# apt-get -s install balsa
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  esound esound-common gdk-imlib1 gnome-bin gnome-libs-data imlib-base libart2
  libaudiofile0 libesd0 libglib1.2 libgnome32 libgnomesupport0 libgnomeui32
  libgnorba27 libgnorbagtk0 libgtk1.2 libjpeg62 liborbit0 libpng2 libproplist0
  libtiff3g libungif3g zlib1g
The following NEW packages will be installed:
  balsa esound esound-common gdk-imlib1 gnome-bin gnome-libs-data imlib-base
  libart2 libaudiofile0 libesd0 libglib1.2 libgnome32 libgnomesupport0
  libgnomeui32 libgnorba27 libgnorbagtk0 libgtk1.2 libjpeg62 liborbit0 libpng2
  libproplist0 libtiff3g libungif3g zlib1g
0 packages upgraded, 24 newly installed, 0 to remove and 10 not upgraded.
```

然后它会列出软件包将安装和配置（或设置）的顺序。

软件包资源列表： apt-setup

发布时间 :2007-01-27 20:23:03

因为 apt-get 会自动为您获取软件包，所以它必须知道在哪里能找到那些还没有安装的软件包。这一信息在 /etc/apt/sources.list 中。尽管您可以手工编辑这个文件（请参阅 sources.list 手册页），但您会发现使用一个交互工具会更方便：

apt-setup

这个工具遍历可以找到 Debian 软件包的位置，如 CDROM、Web 站点和 ftp 站点。当您完成后，它将更改写到您的 /etc/apt/sources.list 文件，这样 apt-get 可以在您请求这些软件包时找到它们。

从 apt-get 到 dselect

发布时间 :2007-01-27 20:23:36

apt-get 工具有许多命令行选项，您可以在 apt-get 手册页查到。缺省项通常工作得很好，如果您发现您在频繁地使用同一个选项，您可以向 /etc/apt/apt.conf 文件添加一个设置。配置文件的这一语法在 apt.conf 手册页中有描述。

除了我们一直使用的 install 命令以外，apt-get 还有许多其它命令。其中之一是 apt-get dselect-upgrade，它遵守为 Debian 系统上的每个软件包所设置的状态（status）。

启动 dselect

发布时间 :2007-01-27 20:24:16

每个软件包的状态都存储在文件 /var/lib/dpkg/status 中，但用另一个交互工具更新它效果最好。

```
# dselect
Debian GNU/Linux `dselect' package handling frontend.


 * 0. [A]ccess      Choose the access method to use.
   1. [U]pdate      Update list of available packages, if possible.
   2. [S]elect      Request which packages you want on your system.
   3. [I]nstall     Install and upgrade wanted packages.
   4. [C]onfig      Configure any packages that are unconfigured.
   5. [R]emove      Remove unwanted software.
   6. [Q]uit        Quit dselect.


Move around with ^P and ^N, cursor keys, initial letters, or digits;
Press <enter> to confirm selection.   ^L redraws screen.


Version 1.6.15 (i386).  Copyright (C) 1994-1996 Ian Jackson.   This is
free software; see the GNU General Public License version 2 or later for
copying conditions.  There is NO warranty.  See dselect --license for details.
```

使用 dselect 的 Select 方式

发布时间 :2007-01-27 20:24:51

选择 Select 选项，您就可以查看和更改软件包的状态。它随即会显示一页帮助信息。读完后就按空格键。现在您可以看到与下面类似的软件包列表：

```
EIOM Pri Section  Package      Inst.ver    Avail.ver   Descr ī ption
     All packages
       Newly available packages
         New Important packages
           New Important packages in section admin
 n* Imp admin    at          <none>      3.1.8-10    Delayed job execution and
 n* Imp admin    cron        <none>      3.0pl1-57.3 management of regular bac
 n* Imp admin    logrotate   <none>      3.2-11      Log rotation utility
           New Important packages in section doc
 n* Imp doc      info        <none>      4.0-4       Standalone GNU Info docum
 n* Imp doc      manpages    <none>      1.29-2      Man pages about using a L
           New Important packages in section editors
 n* Imp editors  ed          <none>      0.2-18.1    The classic unix line edi
 n* Imp editors  nvi         <none>      1.79-16a.1  4.4BSD re-implementation
           New Important packages in section interpreters
 n* Imp interpre perl-5.005  <none>      5.005.03-7. Larry Wall's Practical Ex
           New Important packages in section libs
 n* Imp libs     libident    <none>      0.22-2      simple RFC1413 client lib
 n* Imp libs     libopenldap- <none>     1.2.12-1    OpenLDAP runtime files fo
 n* Imp libs     libopenldap1 <none>     1.2.12-1    OpenLDAP libraries.
 n* Imp libs     libpcre2    <none>      2.08-1      Philip Hazel's Perl Compa
```

软件包状态

发布时间 :2007-01-27 20:25:20

每个软件包的状态都可以在略带神秘的标题 EIOM 下看到。我们关心的列在 M 字符下，其中每个软件包都用下列标记之一来进行标记：

要更改标记，只要按住代表您想要的代码的键（等号、破折号或下划线）即可，但如果您想把标记改为 *（星号），则必须按 +（加号）。

当您完成操作后，用一个大写的 Q 来保存您的更改并退出 Select 屏幕。任何时候，如果您在 dselect 中需要帮助，可以输入 ?（问号）。输入空格则从帮助屏幕返回。

安装和配置（dpkg-reconfigure）

发布时间 :2007-01-27 20:26:04

除非您运行象 apt-get dselect-upgrade 这样的命令，否则 Debian 不会基于状态设置来安装或除去软件包。这个命令实际上立刻为您完成了几个步骤 — 安装、除去和配置。安装和除去步骤不需要停下来询问您任何问题。然而，配置步骤为了按您的希望来设置软件包，可能会问任意多个问题。

要运行这些步骤还有其它方法。例如，您可以从主 dselect 菜单单独选择每一步骤。

有些软件包使用名为 debconf 的系统用于其配置步骤。那些使用 debconf 的软件包可以用各种方式询问其配置问题，询问方式可以是：在文本终端中、通过图形界面或通过 Web 页面等。要配置一个这样的软件包，可以使用 dpkg-reconfigure 命令。您甚至可以用它来确保所有的 debconf 软件包都已完全配置。

```
# dpkg-reconfigure --all
debconf: package "3c5x9utils" is not installed or does not use debconf
debconf: package "3dchess" is not installed or does not use debconf
debconf: package "9menu" is not installed or does not use debconf
debconf: package "9wm" is not installed or does not use debconf
debconf: package "a2ps" is not installed or does not use debconf
debconf: package "a2ps-perl-ja" is not installed or does not use debconf
debconf: package "aalib-bin" is not installed or does not use debconf
```

这会产生一个很长的、未使用 debconf 的软件包列表，但它也会找到使用 debconf 的软件包，并且以易于使用的格式让您回答每个软件包提出的问题。

获取已安装软件包的状态

发布时间 :2007-01-27 20:26:41

目前我们所评论 Debian 软件包管理工具最适于处理有较长软件包列表的多步操作。但它们不包括软件包管理的某些具体操作。对于这一类工作，您可以使用 dpkg。

例如，要获得软件包的完整状态和描述，可以使用 -s 选项：

```
# dpkg -s xsnow
Package: xsnow
Status: install ok installed
Priority: optional
Section: non-free/x11
Installed-Size: 41
Maintainer: Martin Schulze < joey@debian.org>
Version: 1.40-6
Depends: libc6, xlib6g (>= 3.3-5)
Descr ī ption: Brings Christmas to your desktop
 Xsnow is the X-windows application that will let it snow on the
 root window, in between and on windows.  Santa and his reindeer
 will complete your festive-season feeling.
```

文件与其 .deb 之间的链接

发布时间 :2007-01-27 20:27:20

因为 .deb 软件包含有文件，所以您可能会想到应该有办法列出软件包内的文件。哦，没错；只要用 -L 选项：

```
# dpkg -L xsnow
/.
/usr
/usr/doc
/usr/doc/xsnow
/usr/doc/xsnow/copyright
/usr/doc/xsnow/readme.gz
/usr/doc/xsnow/changelog.Debian.gz
/usr/X11R6
/usr/X11R6/bin
/usr/X11R6/bin/xsnow
/usr/X11R6/man
/usr/X11R6/man/man6
/usr/X11R6/man/man6/xsnow.6.gz
```

要用其它方法，并且要找出哪个软件包含有一个特定的文件，可以使用 -S 选项：

```
# dpkg -S /usr/doc/xsnow/copyright
xsnow: /usr/doc/xsnow/copyright
```

软件包的名称刚好在冒号的左边。

查找要安装的软件包

发布时间 :2007-01-27 20:27:53

通常，apt-get 已经知道您可能需要的所有 Debian 软件包。如果它不知道，您或许能在这些 Debian 软件包列表中（或 Web 上的其它地方）找到该软件包。

如果您已经找到并下载了一个 .deb 文件，您可以用 -i 选项安装它：

dpkg -d /tmp/dl/xsnow_1.40-6_i386.deb

如果您找不到希望看到的 .deb 文件，但却发现了 .rpm 文件或一些其它类型的软件包，您或许可以使用 alien 。 alien 程序可以将软件包从各种格式转换成 .deb。

Linux海量文章

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，感谢原创作者！
制作：红联Linux论坛
祝您阅读愉快！