

Linux攻略使用Stow工具管理软件包的技巧 PDF转换可能丢失  
图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/252/2021\\_2022\\_Linux\\_E6\\_94\\_BB\\_E7\\_95\\_c103\\_252924.htm](https://www.100test.com/kao_ti2020/252/2021_2022_Linux_E6_94_BB_E7_95_c103_252924.htm) 本文是关于 Stow 的，它是用于 Linux 的软件安装管理实用程序，它许多地方都优于“久经考验”的 Red Hat 和 Debian 软件包管理系统。通过使用 Stow，可以将应用程序打包成标准的 tar 文件，并按照逻辑安排应用程序二进制文件，以易于访问。Stow 是用 Perl 编写的，易于安装和使用，它虽然简单，但对于在 Linux 机器上组织和管理各种软件安装却很有效。Stow 可以方便地将不同软件包安排在组织良好的目录树结构中。这不仅有助于将属于特殊软件包的文件与其它文件分开，而且还给予用户在任何期望的位置存储或安装软件包的自由，同时使软件看上去被安装在软件包或 OS 要求的位置上。注：如果具有 Linux 的中级知识并安装过 Linux 应用程序，则有助于理解本文。在本文中，我们使用 Stow V1.3.3 和 Perl V5.005。Linux 的软件安装管理在我们开始了解 Stow 之前，先让我们研究一下软件安装管理，它也称作软件包管理。软件安装管理在总体上描述了在系统上安装、卸载、更新和组织软件应用程序（或称为软件包）的活动。在这些活动中，组织应用程序是尤其重要的活动。如果应用程序组织得井井有条，那么在 Linux 机器上安装、升级和卸载应用程序会变得更加容易且更方便。在 Linux 系统上，大多数应用程序必须安装在某一特定目录（通常是 /usr/local/）中才能正确运行和起作用；这种需求来自 Linux 或应用程序本身。如果只在 Linux 系统上安装了很少的应用程序并且它们各自的文件都可以很容易地彼此区分，那么这

不会造成任何严重问题。然而，在实际方案中，可能有大量应用程序安装在同一目录中，该目录不久就会变得很杂乱。设想一下，在安装一个应用程序时，它意外地覆盖了属于另一个应用程序的文件，那么您不得不替换该文件。或者设想一下，在卸载和删除应用程序之前，设法确定哪些文件属于该应用程序。或者，您可能希望找到并删除应用程序不需要的文件。所有这些情形都会花费许多精力和时间，并且要求牢记文件名。为了克服这样的问题，所以将软件安装管理实用程序用于 Linux 系统。它们不仅有助于组织和安排多个软件安装，而且还可以根据需要，任意次数地更改这样的安排，而且不费吹灰之力。Stow 就是这样一种用于 Linux 的实用程序。介绍 Stow 虽然一些 Linux 分发版（如 Red Hat 和 Debian）提供了它们自己的软件包管理实用程序（分别是 rpm 和 apt-get），这些实用程序与 Stow 一样有效，但它们仅用于特定的打包格式（分别是 .rpm 和 .deb）。当管理只打包成 .tar 文件的应用程序时，Stow 是最佳选择。另外，如果人们除了使用 Red Hat 和 Debian 之外，还要使用其它 Linux 分发版，或者希望为现有的软件包管理实用程序找到便宜或免费的备用方案，又或者只想自己控制每件事情，那么 Stow 是一个好的选择。Stow 是遵循 GNU GPL（通用公共许可证，General Public License）的免费软件，它灵活并且可以用充足的用户支持和可用资源进行定制。Stow 通过将应用程序的安装文件存储并组织到其自己的单独目录（称为软件包目录），从而使它们彼此分开，来管理不同应用程序。这些软件包目录驻留在一个 stow 目录中，该目录包含所有已装载的应用程序的软件包目录，是这些目录的父目录。Stow 允许用户

自由地将应用程序及其相关文件安装和存储在任何期望位置的软件包目录中（通过将该位置指定为装载目录），而不是本地安装的应用程序的公共位置（`/usr/local/` 目录）或由应用程序本身指定的缺省位置。该方法有助于使 `/usr/local/` 目录保持整洁，而应用程序被安排在井然有序的目录树结构中，易于查找。注：缺省 `stow` 目录是 `Stow` 自己的目录，位于 `/usr/local/bin` 中；在安装 `Stow` 时可以更改它（请参阅安装 `Stow`）。`Stow` 不仅可以将应用程序组织在 `stow` 目录树中，而且还可以在缺省位置（应用程序最初被要求安装的位置）创建相应的符号链接（`symlink`）。在 `Stow` 术语中，该缺省位置称为目标目录。缺省情况下，目标目录是 `stow` 目录的父目录。尽管对于所有用 `stow` 处理的应用程序而言，用 `/usr/local/stow` 作为 `stow` 目录并且用 `/usr/local` 作为目标目录是习惯作法，但用户可以自由地为 `Stow` 和目标目录指定其它位置（请参阅 `stow` 命令）。`stow` 目录中的应用程序目录树应该与目标目录中所期望的树一样。应用程序相应的符号链接（位于应用程序目标目录中）指向 `stow` 目录中应用程序的实际位置（其软件包目录）。符号链接有点类似于 Windows 快捷方式或 MacOS 别名，它可以表示驻留在另一个位置（或许是不同的驱动器或目录）的应用程序的映象。然后，使用这些符号链接来调用或运行该应用程序。这让人觉得应用程序本身被安装在目标目录中并且是从该目录中运行的。当其它应用程序需要调用正在讨论的应用程序或要与之进行交互，并期望它驻留在其缺省位置时，该方法很有用。注：`Stow` 创建相对符号链接；即符号链接指明了至软件包目录的相对路径（相对于符号链接自己的位置）。`Stow` 设法为每个应用程序

创建最小数目的符号链接 如果可能，仅为软件包目录中的整个子目录树创建一个符号链接。创建这种单一符号链接的过程称为树折叠。Stow 还可用于执行同一应用程序的多次安装。不同的安装可以驻留在不同的位置上，并且 Stow 可用于创建和更新一组应用程序符号链接，使之指向任何期望的位置。Stow 不仅可以组织软件安装，而且还可以根据需要任意次数地更改任何以前的软件安装布局。Stow 将决不会删除 stow 目录中的任何文件或目录，即使在卸载期间也是如此。它只更新或除去目标树中指向应用程序安装的符号链接，所以要再次安装已卸载的应用程序，只要重新装载它就行了。安装和运行 Stow 需求 安装 Stow 确实非常简单。安装和运行 Stow 的唯一需求是 Stow 本身和 Perl V5.005 或更高版本。可以从 GNU FTP 站点（请参阅参考资料以获取链接）下载 Stow 的最新版本。本文使用 V1.3.3，在编写本文时，该版本是最新的。Perl 是所有主要 Linux 分发版标准安装的一部分。要查看您所拥有的版本，使用命令 `perl -v`。可以从 [perl.com](http://perl.com)（请参阅参考资料以获取链接）获得有关 Perl 及其最新版本的更多信息。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)