



## 美国硅谷第二可用区开放

10余款云产品上线  
全美双可用区尽享ECS88折, 10.10~11.9 仅此一月

[查看详情](#)[🏠](#) » [技术](#) » [学习](#) » [查看内容](#)

## RHCSA 系列（二）：如何进行文件和目录管理

2015-9-6 09:30 评论: 2 收藏: 8

原文: <http://www.tecmint.com/file-and-directory-management-in-linux/>  
译文: LCTT <https://linux.cn/article-6155-1.html>

作者: Gabriel Cánepa  
译者: xiqingongzi

在本篇中，我们将回顾一些系统管理员日常任务需要的技能。

*RHCSA：运行文件以及进行文件夹管理 - 第二部分*

### 创建、删除、复制和移动文件及目录

文件和目录管理是每一个系统管理员都应该掌握的必备技能。它包括了从头开始的创建、删除文本文件（每个程序的核心配置）以及目录（你用来组织文件和其它目录），以及识别已有文件的类型。

**touch** 命令 <<https://linux.cn/article-2740-1.html>> 不仅仅能用来创建空文件，还能用来更新已有文件的访问时间和修改时间。

*touch 命令示例*

你可以使用 **file [filename]** 来判断一个文件的类型 (在你用文本编辑器编辑之前，判断类型将会更方便编辑)。

*file 命令示例*

使用 **rm [filename]** 可以删除文件。

*rm 命令示例*

对于目录，你可以使用 **mkdir [directory]** 在已经存在的路径中创建目录，或者使用 **mkdir -p [/full/path/to/directory]** 带全路径创建文件夹。

*mkdir 命令示例*

当你想要去删除目录时，在你使用 **rmdir [directory]** 前，你需要先确保目录是空的，或者使用更加强力的命令(小心使用它!) **rm -rf [directory]**。后者会强制删除 **[directory]** 以及它的内容，所以使用这个命令存在一定的风险。

### 输入输出重定向以及管道

命令行环境提供了两个非常有用的功能：允许重定向命令的输入和输出为另一个文件，以及发送命令的输出到另一个命令，这分别称为重定向和管道。

为了理解这两个重要概念，我们首先需要理解三个最重要的字符输入输出流类型，以 \*nix 的话来说，它们实际上是特殊的文件。

- 标准输入 (即 stdin)，默认连接到键盘。换句话说，键盘是输入命令到命令行的标准输入设备。
- 标准输出 (即 stdout)，默认连接到屏幕。找个设备“接受”命令的输出，并展示到屏幕上。

- 标准错误 (即 `stderr`)，默认是命令的状态消息出现的地方，它也是屏幕。

在下面的例子中，`ls /var` 的结果被发送到 `stdout`(屏幕展示)，`ls /tecmint` 的结果也一样。但在后一种情况下，它显示在标准错误输出上。

输入和输出命令实例

为了更容易识别这些特殊文件，每个文件都被分配有一个文件描述符，这是用于访问它们的抽象标识。主要要理解的是，这些文件就像其它的一样，可以被重定向。这就意味着你可以从一个文件或脚本中捕获输出，并将它传送到另一个文件、命令或脚本中。这样你就可以在磁盘上存储命令的输出结果，用于稍后的分析。

要重定向 `stdin` (fd 0)、`stdout` (fd 1) 或 `stderr` (fd 2)，可以使用如下操作符。

转向操作	效果
<code>&gt;</code>	重定向标准输出到一个文件。如果目标文件存在，内容就会被重写。
<code>&gt;&gt;</code>	添加标准输出到文件尾部。
<code>2&gt;</code>	重定向标准错误输出到一个文件。如果目标文件存在，内容就会被重写。
<code>2&gt;&gt;</code>	添加标准错误输出到文件尾部。
<code>&amp;&gt;</code>	重定向标准错误和标准输出到一个文件。如果目标文件存在，内容就会被重写。
<code>&lt;</code>	使用特定的文件做标准输入。
<code>&lt;&gt;</code>	使用特定的文件做标准输入和标准输出。

与重定向相比，管道是通过在命令后和另外一个命令前之间添加一个竖杠 `(|)`。

记得:

- 重定向是用来定向命令的输出到一个文件，或把一个文件发送作为到一个命令的输入。
- 管道是用来将命令的输出转发到另一个命令作为其输入。

重定向和管道的使用实例

例1：将一个命令的输出到文件

有些时候，你需要遍历一个文件列表。要做到这样，你可以先将该列表保存到文件中，然后再按行读取该文件。虽然你可以直接遍历 `ls` 的输出，不过这个例子是用来说明重定向。

```
1. | # ls -l /var/mail > mail.txt
```

将一个命令的输出重定向到文件

例2：重定向 `stdout` 和 `stderr` 到 `/dev/null`

如果不想让标准输出和标准错误展示在屏幕上，我们可以把这两个文件描述符重定向到 `/dev/null`。请注意对于同样的命令，重定向是如何改变了输出。

```
1. | # ls /var /tecmint
2. | # ls /var/ /tecmint &> /dev/null
```

重定向 `stdout` 和 `stderr` 到 `/dev/null`

例3：使用一个文件作为命令的输入

`cat` 命令 `<http://www.tecmint.com/13-basic-cat-command-examples-in-linux/>` 的经典用法如下

```
1. | # cat [file(s)]
```

您还可以使用正确的重定向操作符发送一个文件作为输入。

```
1. | # cat < mail.txt
```

#### cat 命令实例

#### 例4：发送一个命令的输出作为另一个命令的输入

如果你有一个较大的目录或进程列表，并且想快速定位，你或许需要将列表通过管道传送给grep。

接下来我们会在下面的命令中使用管道，第一个管道是查找所需的关键词，第二个管道是除去产生的 **grep command**。这个例子列举了所有与apache用户有关的进程：

```
1. | # ps -ef | grep apache | grep -v grep
```

#### 发送一个命令的输出作为另一个命令的输入

### 归档，压缩，解包，解压文件

如果你需要传输、备份、或者通过邮件发送一组文件，你可以使用一个存档（或打包）工具，如 **tar** <http://www.tecmint.com/18-tar-command-examples-in-linux/>，通常与gzip，bzip2，或xz等压缩工具配合使用。

您选择的压缩工具每一个都有自己不同的压缩速度和压缩率。这三种压缩工具，gzip是最古老和可以较小压缩的工具，bzip2提供经过改进的压缩，以及xz是最新的而且压缩最大。通常情况下，使用这些压缩工具压缩的文件的扩展名依次是.gz、.bz2或.xz。

命令	缩写	描述
-create	c	创建一个tar归档
-concatenate	A	添加tar归档到另外一个归档中
-append	r	添加非tar归档到另外一个归档中
-update	u	添加比归档中的文件更新的文件
-diff or -compare	d	将归档中的文件和硬盘的文件进行对比
-list	t	列举一个tar压缩包的内容
-extract or -get	x	从归档中提取文件

操作参数	缩写	描述
—directory dir	C	在执行操作前更改目录
—same-permissions and —same-owner	p	分别保留权限和所有者信息
-verbose	v	列举所有读取或提取的文件，如果和 --list 参数一起使用，也会显示文件的大小、所有权和时间戳
—exclude file	—	从存档中排除文件。在这种情况下，文件可以是一个实际的文件或匹配模式。
—gzip or —gunzip	z	使用gzip压缩归档
-bzip2	j	使用bzip2压缩归档

-xz	J	使用xz压缩归档
-----	---	----------

### 例5：创建一个tar文件，然后使用三种压缩工具压缩

在决定使用这个还是那个工具之前，您可能想比较每个工具的压缩效率。请注意压缩小文件或几个文件，结果可能不会有太大的差异，但可能会让你看出它们的差异。

```
1. # tar cf ApacheLogs-$(date +%Y%m%d).tar /var/log/httpd/* # Create an ordinary tarball
2. # tar czf ApacheLogs-$(date +%Y%m%d).tar.gz /var/log/httpd/* # Create a tarball and compress
   with gzip
3. # tar cjf ApacheLogs-$(date +%Y%m%d).tar.bz2 /var/log/httpd/* # Create a tarball and compress
   with bzip2
4. # tar cJf ApacheLogs-$(date +%Y%m%d).tar.xz /var/log/httpd/* # Create a tarball and compress
   with xz
```

*tar 命令实例*

### 例6：归档时同时保存原始权限和所有权

如果你正在从用户的主目录创建备份，你需要要存储的个人文件与原始权限和所有权，而不是通过改变它们的用户帐户或守护进程来执行备份。下面的命令可以在归档时保留文件属性。

```
1. # tar cJf ApacheLogs-$(date +%Y%m%d).tar.xz /var/log/httpd/* --same-permissions --same-owner
```

## 创建软连接和硬链接

在Linux中，有2种类型的链接文件：硬链接和软（也称为符号）链接。因为硬链接文件只是现存文件的另一个名字，使用相同的 inode 号，它指向实际的数据；而符号链接只是指向的文件名。

此外，硬链接不占用磁盘上的空间，而符号链接则占用少量的空间来存储的链接本身的文本。硬链接的缺点就是要求它们必须在同一个文件系统内，因为 inode 在一个文件系统内是唯一的。而符号链接没有这个限制，它们通过文件名而不是 inode 指向其它文件或目录，所以可以跨文件系统。

创建链接的基本语法看起来是相似的：

```
1. # ln TARGET LINK_NAME #从LINK_NAME到Target的硬链接
2. # ln -s TARGET LINK_NAME #从LINK_NAME到Target的软链接
```

### 例7：创建硬链接和软链接

没有更好的方式来形象的说明一个文件和一个指向它的硬链接或符号链接的关系，而不是创建这些链接。在下面的截图中你会看到文件和指向它的硬链接共享相同的inode，都是使用了相同的466个字节的磁盘。

另一方面，在别的磁盘创建一个硬链接将占用5个字节，这并不是说你将耗尽存储容量，而是这个例子足以说明一个硬链接和软链接之间的区别。

*软链接和硬链接之间的不同*

在Linux系统上符号链接的典型用法是指向一个带版本的文件。假设有几个程序需要访问文件fooX.Y，但麻烦是版本经常变化（像图书馆一样）。每次版本更新时我们都需要更新指向fooX.Y的单一引用，而更安全、更快捷的方式是，我们可以让程序寻找名为foo的符号链接，它实际上指向fooX.Y。

这样的话，当你的X和Y发生变化后，你只需更新符号链接foo到新的目标文件，而不用跟踪每个对目标文件的使用并更新。

## 总结

在这篇文章中，我们回顾了一些基本的文件和目录管理技能，这是每个系统管理员的工具集的一部分。请确保阅读了本系列的其它部分，并将这些

主题与本教程所涵盖的内容相结合。

如果你有任何问题或意见，请随时告诉我们。我们总是很高兴从读者那获取反馈.

via: <http://www.tecmint.com/file-and-directory-management-in-linux/> <<http://www.tecmint.com/file-and-directory-management-in-linux/>>

作者：[Gabriel Cánepa](http://www.tecmint.com/author/gacanepa/) <<http://www.tecmint.com/author/gacanepa/>> 译者：xikingongzi <<https://github.com/xikingongzi>> 校对：[wxy](https://github.com/wxy) <<https://github.com/wxy>>

本文由 [LCTT](https://github.com/LCTT/TranslateProject) <<https://github.com/LCTT/TranslateProject>> 原创翻译，Linux 中国 <<file:///root/github/my-notes/Manual/Linux.cn/RHCSA/RHCSA%E7%B3%BB%E5%88%972%E5%A6%82%E4%BD%95%E8%BF%9B%E8%A1%8C%E6%96%87%E4%BB%B6%E5%92%8C%E7%9B%AE%E5%BD%95%E7%AE%A1%E7%90%86-%E6%8A%80%E6%9C%AF.html>> 荣誉推出

原文：<http://www.tecmint.com/file-and-directory-management-in-linux/> <<http://www.tecmint.com/file-and-directory-management-in-linux/>>

作者：Gabriel Cánepa

译文：[LCTT](http://lctt.github.io/) <<http://lctt.github.io/>> <https://linux.cn/article-6155-1.html> <<https://linux.cn/article-6155-1.html>> 译者：xikingongzi

发表评论

验证码  [换一个](#)



热点评论

- mefirst\_love [Internet Explorer 11.0|Windows 8.1] 2015-9-24 17:23

把英文版全能翻译到这个地步，不错，辛苦了

赞 2
- 妙正灰 [Chrome 45.0|Windows 10] 2015-9-6 13:45

还是这篇比较好

赞 2

体验环境



本文导航

- 创建、删除、复制和移动文件及目录
- 输入输出重定向以及管道
  - 重定向和管道的使用实例
- 归档，压缩，解包，解压文件
- 创建软连接和硬链接
- 总结

#### 相关阅读

 RHCSA

- |  |           |
|--|-----------|
| • RHCSA 系列（一）：回顾基础命令及系统文档                      | 2015-9-2  |
| • RHCSA 系列（七）：使用 ACL（访问控制列表）和挂载 Samba/NFS 共享   | 2015-9-22 |
| • RHCSA 系列（八）：加固 SSH，设定主机名及启用网络服务              | 2015-9-23 |
| • RHCSA 系列（九）：安装、配置及加固一个 Web 和 FTP 服务器         | 2015-9-24 |
| • RHCSA 系列（十）：Yum 包管理、Cron 自动任务计划和监控系统日志       | 2015-9-26 |
| • RHCSA 系列（十一）：使用 firewalld 和 iptables 来控制网络流量 | 2015-9-29 |

Linux.CN © 2003-2015 Linux中国 | Powered by **DX** | 图片存储于七牛云存储

京ICP备05083684号-1 京公网安备110105001595

服务条款 | 除特别申明外，本站原创内容版权遵循 CC-BY-NC-SA 协议规定

 [HTTPS://LINUX.CN](https://linux.cn)    