Linux 入门指南

Linux 命令

文件状态： [ ] 正在修改 [ ] 正在发布	当前版本：	V1.0
	作者：	Adolph
	完成日期：	2019.2.25
	审核：	
	完成日期：	

版本历史

版本号	作者	修改日期	修改说明	审核	备注
V1.0	Adolph	2019.2.25	初始版本		

目录

Linux 命令基本格式：

本节开始，我们不会再见到图形界面了，因为对服务器来讲，图形界面会占用更多的系统资源，而且会安装更多的服务、开放更多的端口，这对服务器的稳定性和安全性都有负面影响。其实，服务器是一个连显示器都没有的家伙，要图形界面干十么？

说到这里，有很多人会很崩溃。笔者就经常听到抱怨 Linux 是落后于时代的老古董，就像笔者的白头发一样！但是，大家要理解，对服务器来讲，稳定性、可靠性、安全性才是最主要的。而简单易用不是服务器需要考虑的事情，所以学习 Linux，这些枯燥的命令是必须学习和记忆的内容。

命令提示符

登录系统后，第一眼看到的内容是：

[root@localhost ~]#

这就是 Linux 系统的命令提示符。那么，这个提示符的含义是什么呢？

[]：这是提示符的分隔符号，没有特殊含义。

root：显示的是当前的登录用户，笔者现在使用的是 root 用户登录。

@：分隔符号，没有特殊含义。

localhost：当前系统的简写主机名（完整主机名是 localhost.localdomain）。

~：代表用户当前所在的目录，此例中用户当前所在的目录是家目录。

#：命令提示符，Linux 用这个符号标识登录的用户权限等级。如果是超级用户，提示符就是 #；如果是普通用户，提示符就是 $。

家目录是什么？ Linux 系统是纯字符界面，用户登录后，要有一个初始登录的位置，这个初始登录位置就称为用户的家：

超级用户的家目录：/root/。

普通用户的家目录：/home/用户名/。

用户在自己的家目录中拥有完整权限，所以我们也建议操作实验可以放在家目录中进行。我们切换一下用户所在目录，看看有什么效果。

[root@localhost ~]# cd /usr/local/

[root@localhost local]#

仔细看，如果切换用户所在目录，那么命令提示符中的会变成用户当前所在目录的最后一个目录（不显示完整的所在目录 /usr/ local/，只显示最后一个目录 local)。

命令的基本格式

接下来看看 Linux 命令的基本格式:

[root@localhost ~]# 命令[选项][参数]

命令格式中的 [] 代表可选项，也就是有些命令可以不写选项或参数，也能执行。那么，我们就用 Linux 中最常见的 ls 命令来解释一下命令的格式。如果按照命令的分类，那么 ls 命令应该属于目录操作命令。

[root@localhost ~]# ls

anaconda-ks.cfg install.log install.log.syslog

1) 选项的作用

ls 命令之后不加选项和参数也能执行，不过只能执行最基本的功能，即显示当前目录下的文件名。那么加入一个选项，会出现什么结果？

[root@localhost ~]# Is -l

总用量 44

-rw-------.1 root root 1207 1 月 14 18:18 anaconda-ks.cfg

-rw-r--r--.1 root root 24772 1 月 14 18:17 install.log

-rw-r--r--.1 root root 7690 1 月 14 18:17 install.log.syslog

如果加一个"-l"选项，则可以看到显示的内容明显增多了。"-l"是长格式(long list)的意思，也就是显示文件的详细信息。至于"-l"选项的具体含义，我们稍后再详细讲解。可以看到选项的作用是调整命令功能。如果没有选项，那么命令只能执行最基本的功能；而一旦有选项，则可以显示更加丰富的数据。

Linux 的选项又分为短格式选项（-l）和长格式选项（--all）。短格式选项是英文的简写，用一个减号调用，例如：

[root@localhost ~]# ls -l

而长格式选项是英文完整单词，一般用两个减号调用，例如：

[root@localhost ~]# ls --all

一般情况下，短格式选项是长格式选项的缩写，也就是一个短格式选项会有对应的长格式选项。当然也有例外，比如 ls 命令的短格式选项 -l 就没有对应的长格式选项。所以具体的命令选项可以通过后面我们要学习的帮助命令来进行查询。

2）参数的作用

参数是命令的操作对象，一般文件、目录、用户和进程等可以作为参数被命令操作。例如：

[root@localhost ~]# ls -l anaconda-ks.cfg

-rw-------.1 root root 1207 1 月 14 18:18 anaconda-ks.cfg

但是为什么一开始 ls 命令可以省略参数？那是因为有默认参数。命令一般都需要加入参数，用于指定命令操作的对象是谁。如果可以省略参数，则一般都有默认参数。例如：

[root@localhost ~]# ls

anaconda-ks.cfg install.log install.log.syslog

这个 ls 命令后面没有指定参数，默认参数是当前所在位置，所以会显示当前目录下的文件名。

linux ls 命令：显示当前目录下的文件:

ls 是最常见的目录操作命令，主要作用是显示目录下的内容。这个命令的基本信息如下：

命令名称：ls。

英文原意：list。

所在路径：/bin/ls。

执行权限：所有用户。

功能描述：显示目录下的内容。

对命令的基本信息进行说明：英文原意有助于理解和记忆命令；执行权限是命令只能被超级用户执行，还是可以被所有用户执行；功能描述指的是这个命令的基本作用。

本节主要讲解基本命令，基本信息有助于大家记忆，本章所有命令都会加入命令的基本信息。在后续章节中，大家要学会通过帮助命令、搜索命令来自己查询这些信息，所以不再浪费篇幅来写了。

命令格式

[root@localhost ~]#ls [选项][文件名或目录名]

选项：

-a：显示所有文件；

--color=when：支持颜色输出，when 的值默认是 always（总显示颜色），也可以是 never（从不显示颜色）和 auto（自动）；

-d：显示目录信息，而不是目录下的文件；

-h：人性化显示，按照我们习惯的单位显示文件大小；

-i：显示文件的 i 节点号；

-l：长格式显示；

学习命令，主要学习的是命令选项，但是每个命令的选项非常多，比如 ls 命令就支持五六十个选项，我们不可能讲解每个选项，也没必要讲解每个选项，本章只能讲解最为常用的选项，即可满足我们曰常操作使用。

常见用法

【例 1】"-a"选项

-a 选项中的 a 是 all 的意思，也就是显示隐藏文件。例如：

```
[root@localhost ~]# ls
anaconda-ks.cfg install.log install.log.syslog
[root@localhost ~]# Is -a
.anaconda-ks.cfg .bash_logout .bashrc
install.log .mysql_history .viminfo ...bash_history .bash_profile .cshrc install.log.syslog .tcshrc
```

可以看到，加入"-a"选项后，显示出来的文件明显变多了。而多出来的这些文件都有一个共同的特性，就是以"."开头。在 Linux 中以"."开头的文件是隐藏文件，只有通过"-a"选项才能查看。

说到隐藏文件的查看方式，曾经有读者问我："为什么在 Linux 中查看隐藏文件这么简单？这样的话隐藏文件还有什么意义?"其实，他理解错了隐藏文件的含义。

隐藏文件不是为了把文件藏起来不让其他用户找到，而是为了告诉用户这些文件都是重要的系统文件，如非必要，不要乱动！所以，不论是 Linux 还是 Windows 都可以非常简单地查看隐藏文件，只是在 Windows 中绝大多数的病毒和木马都会把自己变成隐藏文件，给用户带来了错觉，以为隐藏文件是为了不让用户发现。

【例 2】"-l"选项

```
[root@localhost ~]# ls -l
总用量 44
-rw-------.1 root root 1207 1 月 14 18:18 anaconda-ks.cfg
-rw-r-r--.1 root root 24772 1 月 14 18:17 install.log
-rw-r-r--.1 root root 7690 1 月 14 18:17 install.log.syslog
#权限 引用计数 所有者 所属组 大小 文件修改时间 文件名
```

我们已经知道"-l"选项用于显示文件的详细信息，那么"-l"选项显示的这 7 列分别是什么含义？

第一列：权限，具体权限的含义将在后续章节中讲解。

第二列：引用计数，文件的引用计数代表该文件的硬链接个数，而目录的引用计数代表该目录有多少个一级子目录。

第三列：所有者，也就是这个文件属于哪个用户。默认所有者是文件的建立用户

第四列：所属组，默认所属组是文件建立用户的有效组，一般情况下就是建立用户的所在组。

第五列：大小，默认单位是字节。

第六列：文件修改时间，文件状态修改时间或文件数据修改时间都会更改这个时间，注意这个时间不是文件的创建时间。

第七列：文件名。

【例 3】"-d"选项

如果我们想查看某个目录的详细信息，例如：

```
[root@localhost ~]# ls -l /root/
总用量 44
-rw-------.1 root root 1207 1 月 14 18:18 anaconda-ks.cfg
-rw-r-r--.1 root root 24772 1 月 14 18:17 install.log
-rw-r-r--.1 root root 7690 1 月 14 18:17 install.log.syslog
```

这个命令会显示目录下的内容，而不会显示这个目录本身的详细信息。如果想显示目录本身的信息，就必须加入"-d"选项。

```
[root@localhost ~]# ls -ld /root/
dr-xr-x---.2 root root 4096 1 月 20 12:30 /root/
```

【例 4】"-h"选项

"ls-l"显示的文件大小是字节，但是我们更加习惯的是千字节用 KB 显示，兆字节用 MB 显示，而"-h"选项就是按照人们习惯的单位显示文件大小的，例如：

```
[root@localhost ~]# ls  -lh
总用量 44K
-rw-------.1 root root1.2K 1 月 14 18:18 anaconda-ks.cfg
-rw-r-r--.1 root root25K 1 月 14 18:17 install.log
-rw-r-r--.1 root root7.6K 1 月 14 18:17 install.log.syslog
```

【例 5】"-i"选项

每个文件都有一个被称作 inode（i 节点）的隐藏属性，可以看成系统搜索这个文件的 ID，而"-i"选项就是用来

查看文件的 inode 号的，例如：

[root@localhost ~]# ls -i

262418 anaconda-ks.cfg 262147 install.log 262148 install.log.syslog

Linux cd 命令：切换目录:

cd 是切换所在目录的命令，这个命令的基本信息如下。

命令名称：cd。

英文原意：change directory。

所在路径：Shell 内置命令。

执行权限：所有用户。

功能描述：切换所在目录。

Linux 的命令按照来源方式分为两种：Shell 内置命令和外部命令。所谓 Shell 内置命令，就是 Shell 自带的命令，这些命令是没有执行文件的；而外部命令就是由程序员单独开发的，是命令，所以会有命令的执行文件。Linux 中的绝大多数命令是外部命令，而 cd 命令是一个典型的 Shell 内置命令，所以 cd 命令没有执行文件所在路径。

命令格式

[root@localhost ~]#cd [目录名]

cd 命令是非常简单的命令，仅有的两个选项 -P 和 -L 的作用非常有限，很少使用：

-P（大写）是指如果切换的目录是软链接目录，则进入其原始的物理目录，而不是进入软链接目录；

-L（大写）是指如果切换的目录是软链接目录，则直接进入软链接目录。

常见用法

【例 1】基本用法

cd 命令切换目录只需在命令后加目录名称即可。例如：

[root@localhost ~]# cd /usr/local/src/

[root@localhost src]#

#进入/usr/local/src/ 目录

通过命令提示符，我们可以确定当前所在目录已经切换。

【例 2】简化用法

cd 命令可以识别一些特殊符号，用于决速切换所在目录，这些符号如表 1 所示。

表 1 cd 命令的特殊符号

特殊符号	作　用
~	代表用户的家目录
-	代表上次所在目录
.	代表当前目录
..	代表上级目录

这些简化用法以加快命令切换，我们来试试。

[root@localhost src]# cd ~

[root@localhost ~]#

"cd~"命令可以快速回到用户的家目录，cd 命令直接按回车键也是快速切换到家目录。

[root@localhost~]#cd /etc/

[root@localhost etc]#cd

[root@localhost ~]#

直接使用 cd 命令，也回到了家目录。

再试试"cd-"命令。

[root@localhost ~]# cd/usr/local/src/

#进入/usr/local/src/目录

[root@localhost src]# cd -/root

[root@localhost ~]#

#"cd-"命令回到进入 src 目录之前的家目录

[root@localhost ~]# cd-

```
/usr/local/src
[root@localhost src]#
#再执行一遍"cd-"命令，又回到了 /usr/local/src/ 目录。
```

再来试试"."和".."。

```
[root@localhost ~]# cd /usr/local/src/
#进入测试目录
[root@localhost src]# cd..
#进入上级目录
[root@localhost local]# pwd
/usr/local
#pwd 是查看当前所在目录的命令，可以看到我们进入了上级目 /usr/local/
[root@localhost local]# cd.
#进入当前目录
[root@localhost local]# pwd
/usr/local
#这个命令不会有目录的改变，只是告诉大家"."代表当前目录。
```

绝对路径和相对路径

cd 命令本身不难，但有两个非常重要的概念，就是绝对路径和相对路径。初学者由于对字符界面不熟悉，所以有大量的错误都是因为对这两个路径没有搞明白，比如进错了目录、打开不了文件、打开的文件和系统文件不一致等。所以我们先来区分一下这两个路径。

首先，我们先要弄明白什么是绝对、什么又是相对。其实我们一直说现实生活中没有绝对的事情，没有绝对的大，也没有绝对的小；没有绝对的快，也没有绝对的慢。这只是由于参照物的不同或认知的局限，导致会暂时认为某些东西可能是绝对的、不能改变的。比如目前我们认为光速是最快的速度，我们不能突破光速的限制。但也有可能随着技术的进步，我们会突破这一限制。

但在 Linux 的路径中是有绝对路径的，那是因为 Linux 有最高目录，也就是根目录。如果路径是从根目录开始，一级一级指定的，那使用的就是绝对路径。例如：

[root@localhost ~]# cd /usr/local/src/

[root@localhost src]# cd /etc/rc.d/init.d/

这些切换目录的方法使用的就是绝对路径。所谓相对路径，就是只从当前所在目录开始，切换目录。例如：

[root@localhost /]# cd etc/

#当前所在路径是/目录，而/目录下有 etc 目录，所以可以切换

[root@localhost etc]# cd etc/

-bash:cd:etc/:没有那个文件或目录

#而同样的命令，由于当前所在目录改变了，所以就算是同一个命令也会报错，除非在/etc/目录中还有一个 etc 目录

所以，虽然绝对路径输入更加烦琐，但是更准确，报错的可能性也更小。对初学者而言，笔者还是建议大家使用绝对路径。本教程为了使命令更容易理解，也会尽量使用绝对路径。

再举个例子，假设我当前在 root 用户的家目录中。

[root@localhost ~]#

那么，该如何使用相对路径进入 /usr/local/src/ 目录中呢？

[root@localhost ~]# cd ../usr/local/src/

从我当前所在路径算起，加入"../"代表进入上一级目录，而上一级目录是根目录，而根目录中有 usr 目录，就会一级一级地进入 src 目录了。

Linux mkdir 命令：创建目录（文件夹）：

mkdir 是创建目录的命令，其基本信息如下：

命令名称：mkdir。

英文原意：make directories。

所在路径：/bin/mkdir。

执行权限：所有用户。

功能描述：创建空目录。

命令格式

[root@localhost -]# mkdir [选项]目录名

选项：

-p: 递归建立所需目录

mkdir 也是一个非常简单的命令，其主要作用就是新建一个空目录。

常见用法

【例 1】建立目录。

[root@localhost ~]#mkdir cangls

[root@localhost ~]#ls

anaconda-ks.cfg cangls install.log install.log.syslog

我们建立一个名为 cangls 的目录，通过 ls 命令可以查看到这个目录已经建立。注意，我们在建立目录的时候使用的是相对路径，所以这个目录被建立到当前目录下。

【例 2】递归建立目录。

如果想建立一串空目录，可以吗？

[root@localhost ~]# mkdir lm/movie/jp/cangls

mkdir:无法创建目录"lm/movie/jp/cangls":没有那个文件或目录

笔者想建立一个保存电影的目录，结果这条命令报错，没有正确执行。这是因为这 4 个目录都是不存在的，mkdir 默认只能在已经存在的目录中建立新目录。

如果需要建立一系列的新目录，则需要加入"-p"选项，递归建立才可以。例如:

[root@localhost ~]# mkdir -p lm/movie/jp/cangls

[root@localhost ~]# ls

anaconda-ks.cfg cangls install.log install.log.syslog lm

[root@localhost ~]# ls lm/

movie

#这里只查看一级子目录，其实后续的 jp 目录、cangls 目录都已经建立

所谓的递归建立，就是一级一级地建立目录。

Linux rmdir 命令：删除空目录:

既然有建立目录的命令，就一定会有删除目录的命令 rmdir，其基本信息如下：

命令名称：rmdir。

英文原意：remove empty directories。

所在路径：/bin/rmdir。

执行权限：所有用户。

功能描述：删除空目录。

命令格式

[root@localhost -]# rmdir [选项]目录名

选项：

-p： 递归删除目录

常见用法

[root@localhost ~]#rmdir cangls

就这么简单，命令后面加目录名称即可。既然可以递归建立目录，当然也可以递归删除目录。例如：

[root@localhost ~]# rmdir -p lm/movie/jp/cangls/

但 rmdir 命令的作用十分有限，因为只能删除空目录，所以一旦目录中有内容，就会报错。例如：

[root@localhost # mkdir test

#建立测试目录

[root@localhost ~]# touch test/boduo

[root@localhost ~]# touch test/longze

#在测试目录中建立两个文件

[root@localhost ~]# rmdir test/

rmdir:删除"test/"失败：目录非空

这个命令比较"笨"，所以我们不太常用。后续我们不论删除的是文件还是目录，都会使用 rm 命令（后续章节会讲）。

Linux touch 命令：修改文件的时间戳:

touch 的意思是触摸，如果文件不存在，则会建立空文件；如果文件已经存在，则会修改文件的时间戳（访问时间、数据修改时间、状态修改时间都会改变）。

千万不要把 touch 命令当成新建文件的命令，牢牢记住这是触摸的意思。这个命令的基本信息如下：

命令名称：touch。

英文原意：change file timestamps。

所在路径：/bin/touch。

执行权限：所有用户。

功能描述：修改文件的时间戳。

命令格式

[root@localhost ~]# touch [选项]文件名或目录名

选项：

-a：只修改文件的访问时间（Access Time）

-c：如果文件不存在，则不建立新文件

-d：把文件的时间改为指定的时间

-m：只修改文件的数据修改时间（Modify Time）

Linux 中的每个文件都有三个时间，分别是访问时间（Access Time）、数据修改时间（Modify Time）和状态修改时间（Change Time）。这三个时间可以通过 stat 命令来进行查看。

不过，touch 命令只能手工指定只修改访问时间，或是只修改数据修改时间，而不能指定只修改状态修改时间。因为不论是修改访问时间，还是修改文件的数据时间，对文件来讲，状态都会发生改变，即状态修改时间会随之改变。我们稍后讲 stat 命令时再具体举例。

注意，在 Linux 中，文件没有创建时间。

常见用法

[root@localhost ~]#touch bols

#建立名为 bols 的空文件

如果文件不存在，则会建立文件。

[root@localhost ~]#touch bols

[root@localhost ~]#touch bols

#而如果文件已经存在，则也不会报错，只是会修改文件的访问时间

常见用法

[root@localhost ~]#touch bols

#建立名为 bols 的空文件

如果文件不存在，则会建立文件。

[root@localhost ~]#touch bols

[root@localhost ~]#touch bols

#而如果文件已经存在，则也不会报错，只是会修改文件的访问时间

Linux rm 命令：删除文件或目录:

rm 是强大的删除命令，不仅可以删除文件，也可以删除目录。这个命令的基本信息如下。

- 命令名称：rm
- 英文原意：remove files or directories。
- 所在路径：/bin/rm。
- 执行权限：所有用户。
- 功能描述：删除文件或目录。

命令格式

[root@localhost ~]# rm[选项] 文件或目录

选项：

- -f：强制删除（force）
- -i：交互删除，在删除之前会询问用户
- -r：递归删除，可以删除目录（recursive）

常见用法

【例 1】基本用法。

rm 命令如果任何选项都不加，则默认执行的是"rm -i 文件名"，也就是在删除一个文件之前会先询问是否删除。

例如：

```
[root@localhost ~]# touch cangls
[root@localhost ~]# rm cangls
rm:是否删除普通空文件"cangls"?y
#删除前会询问是否删除
```

【例 2】 删除目录。

如果需要删除目录，则需要使用"-r"选项。例如:

```
[root@localhost ~]# mkdir -p /test/lm/movie/jp/
#递归建立测试目录
[root@localhost ~]# rm /test/
rm:无法删除"/test/": 是一个目录
#如果不加"-r"选项，则会报错
[root@localhost ~]# rm -r /test/
rm:是否进入目录"/test"?y
rm:是否进入目录"/test/lm/movie"?y
rm:是否删除目录"/test/lm/movie/jp"?y
rm:是否删除目录"/test/lm/movie"?y
rm:是否删除目录"/test/lm"?y
rm:是否删除目录"/test"?y
#会分别询问是否进入子目录、是否删除子目录
```

大家会发现，如果每级目录和每个文件都需要确认，那么在实际使用中简直是灾难！

【例 3】强制删除。

如果要删除的目录中有 1 万个子目录或子文件，那么普通的 rm 删除最少需要确认 1 万次。所以，在真正删除文件的时候，我们会选择强制删除。例如：

[root@localhost ~]# mkdir -p /test/lm/movie/jp/

#重新建立测试目录

[root@localhost ~]# rm -rf/test/

#强制删除，一了百了

加入了强制功能之后，删除就会变得很简单，但是需要注意，数据强制删除之后无法恢复，除非依赖第三方的数据恢复工具，如 extundelete 等。但要注意，数据恢复很难恢复完整的数据，一般能恢复 70%~80% 就很难得了。所以，与其把宝压在数据恢复上，不如养成良好的操作习惯。

虽然"-rf"选项是用来删除目录的，但是删除文件也不会报错。所以，为了使用方便，一般不论是删除文件还是删除目录，都会直接使用"-rf"选项。

Linux cp 命令：复制文件和目录:

cp 是用于复制的命令，其基本信息如下：

- 命令名称：cp；
- 英文原意：copy files and directories；
- 所在路径：/bin/cp；
- 执行权限：所有用户；
- 功能描述：复制文件和目录；

命令格式

[root@localhost ~]# cp [选项] 源文件 目标文件

选项：

- -a：相当于 -d、-p、-r 选项的集合，这几个选项我们一一介绍；
- -d：如果源文件为软链接（对硬链接无效），则复制出的目标文件也为软链接；
- -i：询问，如果目标文件已经存在，则会询问是否覆盖；
- -l：把目标文件建立为源文件的硬链接文件，而不是复制源文件；
- -s：把目标文件建立为源文件的软链接文件，而不是复制源文件；

- -p：复制后目标文件保留源文件的属性（包括所有者、所属组、权限和时间）；
- -r：递归复制，用于复制目录；

常见用法

【例 1】基本用法。

cp 命令既可以复制文件，也可以复制目录。我们先来看看如何复制文件，例如：

```
[root@localhost ~]# touch cangls
#建立源文件
[root@localhost ~]# cp cangls /tmp/
#把源文件不改名复制到 /tmp/ 目录下
```

如果需要改名复制，则命令如下：

```
[root@localhost ~]# cp cangls /tmp/bols
#改名复制
```

如果复制的目标位置已经存在同名的文件，则会提示是否覆盖，因为 cp 命令默认执行的是“cp -i”的别名，例如：

```
[root@localhost ~]# cp cangls /tmp/
cp:是否覆盖"/tmp/cangls"?y
#目标位置有同名文件，所以会提示是否覆盖
```

接下来我们看看如何复制目录，其实复制目录只需使用“-r”选项即可，例如：

```
[root@localhost ~]# mkdir movie
#建立测试目录
[root@localhost ~]# cp -r /root/movie/ /tmp/
#目录原名复制
```

【例 2】复制软链接属性

如果源文件不是一个普通文件，而是一个软链接文件，那么是否可以复制软链接的属性呢？我们试试：

```
[root@localhost ~]# ln -s /root/cangls /tmp/cangls_slink
#建立一个测试软链接文件/tmp/cangls_slink
[root@localhost ~]# ll /tmp/cangls_slink
lrwxrwxrwx 1 root root 12 6 月 14 05:53 /tmp/cangls_slink -> /root/cangls
#源文件本身就是一个软链接文件
[root@localhost ~]# cp /tmp/cangls_slink /tmp/cangls_t1
#复制软链接文件，但是不加"-d"选项
[root@localhost ~]# cp -d /tmp/cangls_slink /tmp/cangls_t2
#复制软链接文件，加入"-d"选项
[root@localhost ~]# ll /tmp/cangls_t1 /tmp/cangls_t2
-rw-r--r-- 1 root root 0 6 月 14 05:56 /tmp/cangls_t1
#会发现不加"-d"选项，实际复制的是软链接的源文件，而不是软链接文件
lrwxrwxrwx 1 root root 12 6 月 14 05:56/tmp/ cangls_t2-> /root/cangls
#而如果加入了"-d"选项，则会复制软链接文件
```

这个例子说明：如果在复制软链接文件时不使用"-d"选项，则 cp 命令复制的是源文件，而不是软链接文件；只有加入了"-d"选项，才会复制软链接文件。请大家注意，"-d"选项对硬链接是无效的。

【例 3】保留源文件属性复制

我们发现，在执行复制命令后，目标文件的时间会变成复制命令的执行时间，而不是源文件的时间。例如：

```
[root@localhost ~]# cp /var/lib/mlocate/mlocate.db /tmp/
[root@localhost ~]# ll /var/lib/mlocate/mlocate.db
-rw-r-----1 root slocate2328027 6 月 14 02:08/var/lib/mlocate/mlocate.db
#注意源文件的时间和所属组
[root@localhost ~]#ll /tmp/mlocate.db
-rw-r----- 1 root root2328027 6 月 14 06:05/tmp/mlocate.db
#由于复制命令由 root 用户执行，所以目标文件的所属组为了 root，而且时间也变成了复制命令的执行时间
```

而当我们执行备份、曰志备份的时候，这些文件的时间可能是一个重要的参数，这就需执行"-p"选项了。这个选项会保留源文件的属性，包括所有者、所属组和时间。例如：

```
[root@localhost ~]# cp -p /var/lib/mlocate/mlocate.db /tmp/mlocate.db_2
#使用"-p"选项
[root@localhost ~]# ll /var/lib/mlocate/mlocate.db /tmp/mlocate.db_2
-rw-r----- root slocate 2328027 6 月 14 02:08 /tmp/mlocate.db_2
-rw-r----- root slocate 2328027 6 月 14 02:08 /var/lib/mlocate/mlocate.db
#源文件和目标文件的所有属性都一致，包括时间
```

我们之前讲过，"-a"选项相当于"-d、-p、-r"选项，这几个选项我们已经分别讲过了。所以，当我们使用"-a"选项时，目标文件和源文件的所有属性都一致，包街原文件的所有者，所属组、时间和软链接性。使用"-a" 选项来取代"-d、-p、-r"选项更加方便。

【例 4】 "-l"和"-s"选项

我们如果使用"-l"选项，则目标文件会被建立为源文件的硬链接；而如果使用了 "-s"选项，则目标文件会被建立为源文件的软链接。

这两个选项和"-d"选项是不同的，"d"选项要求源文件必须是软链接，目标文件才会复制为软链接；而"-l"和"-s"选项的源文件只需是普通文件，目标文件就可以直接复制为硬链接和软链接。例如：

```
[root@localhost ~]# touch bols
#建立测试文件
[root@localhost ~]# ll -i bols
262154-rw-r--r-- 1 root root 0 6 月 14 06:26 bols
#源文件只是一个普通文件，而不是软链接文件
[root@localhost ~]# cp -l /root/bols /tmp/bols_h
[root@localhost ~]# cp -s /root/bols /tmp/bols_s
#使用"-l" 和"-s"选项复制
[root@localhost ~]# ll -i /tmp/bols_h /tmp/bols_s
262154-rw-r--r-- 2root root 0 6 月 14 06:26/tmp/bols_h
#目标文件 /tmp/bols_h 为源文件的硬链接文件
932113 lrwxrwxrwx 1 root root 10 6 月 14 06:27/tmp/bols_s -> /root/bols
#目标文件 /tmp/bols_s 为源文件的软链接文件
```

Linux mv 命令：移动文件或改名：

mv 是用来剪切的命令，其基本信息如下。

- 命令名称：mv。
- 英文原意：move(rename)files。
- 所在路径：/bin/mv。
- 执行权限：所有用户。
- 功能描述：移动文件或改名。

命令格式

[root@localhost ~]# mv 【选项】 源文件 目标文件

选项：

- -f：强制覆盖，如果目标文件已经存在，则不询问，直接强制覆盖；
- -i：交互移动，如果目标文件已经存在，则询问用户是否覆盖（默认选项）；
- -n：如果目标文件已经存在，则不会覆盖移动，而且不询问用户；
- -v：显示详细信息；

常见用法

【例 1】移动文件或目录。

```
[root@localhost ~]# mv cangls /tmp/
#移动之后，源文件会被删除，类似剪切
[root@localhost ~]# mkdir movie
[root@localhost ~]# mv movie/ /tmp/
#也可以移动目录。和 rm、cp 不同的是，mv 移动目录不需要加入"-r"选项
```

如果移动的目标位置已经存在同名的文件，则同样会提示是否覆盖，因为 mv 命令默认执行的也是"mv -i"的别名，例如：

```
[root@localhost ~]# touch cangls
```

#重新建立文件

[root@localhost ~]# mv cangls /tmp/

mv:县否覆盖"tmp/cangls"? y

#由于 /tmp/ 目录下已经存在 cangls 文件，所以会提示是否覆盖，需要手工输入 y 覆盖移动

【例 2】强制移动。

之前说过，如果目标目录下已经存在同名文件，则会提示是否覆盖，需要手工确认。这时如果移动的同名文件较多，则需要一个一个文件进行确认，很不方便。

如果我们确认需要覆盖已经存在的同名文件，则可以使用"-f"选项进行强制移动，这就不再需要用户手工确认了。例如：

[root@localhost ~]# touch cangls

#重新建立文件

[root@localhost ~]# mv -f cangls /tmp/

#就算 /tmp/ 目录下已经存在同名的文件，由于"-f"选项的作用，所以会强制覆盖

【例 3】不覆盖移动。

既然可以强制覆盖移动，那也有可能需要不覆盖的移动。如果需要移动几百个同名文件，但是不想覆盖，这时就需要"-n"选项的帮助了。例如：

[root@localhost ~]# ls /tmp*ls

/tmp/bols /tmp/cangls

#在/tmp/目录下已经存在 bols、cangls 文件了

[root@localhost ~]# mv -vn bols cangls lmls /tmp/

"lmls"->"/tmp/lmls"

#再向 /tmp/ 目录中移动同名文件，如果使用了 "-n"选项，则可以看到只移动了 lmls，而同名的 bols 和 cangls 并没有移动（"-v"选项用于显示移动过程）

【例 4】改名。

如果源文件和目标文件在同一目录中，那就是改名。例如：

```
[root@localhost ~]# mv bols lmls
#把 bols 改名为 lmls
```

目录也可以按照同样的方法改名。

【例 5】显示移动过程。

如果我们想要知道在移动过程中到底有哪些文件进行了移动，则可以使用"-v"选项来查看详细的移动信息。例如：

```
[root@localhost ~]# touch test1.txt test2.txt test3.txt
#建立三个测试文件
[root@localhost ~]# mv -v *.txt/tmp/
"test1.txt" -> "/tmp/test1.txt"
"test2.txt" -> "/tmp/test2.txt"
"test3.txt" -> "/tmp/test3.txt"
#加入"-v"选项，可以看到有哪些文件进行了移动
```

Linux tar 压缩命令：打包与解打包命令:

".tar"格式的打包和解打包都使用 tar 命令，区别只是选项不同。我们先看看 tar 命令的基本信息。

- 命令名称：tar。
- 英文原意：tar。
- 所在路径：/bin/tar。
- 执行权限：所有用户。
- 功能描述：打包与解打包命令。

打包命令格式

[root@localhost ~]#tar [选项] [-f 压缩包名] 源文件或目录

选项：

- -c：打包；
- -f：指定压缩包的文件名。压缩包的扩展名是用来给管理员识别格式的，所以一定要正确指定扩展名；
- -v：显示打包文件过程；

【例 1】基本使用。

我们先打包一个文件练练手。

```
[root@localhost ~]# tar -cvf anaconda-ks.cfg.tar anaconda-ks.cfg
```

#把 anacondehks.cfg 打包为 anacondehks.cfg.tar 文件
选项"-cvf"一般是习惯用法，记住打包时需要指定打包之后的文件名，而且要用".tar"作为扩展名。那打包目录呢？我们也试试：

[root@localhost ~]# ll -d test/
drwxr-xr-x 2 root root 4096 6 月 17 21:09 test/
#test 是我们之前的测试目录
[root@localhost ~]# tar -cvf test.tar test/
test/
test/test3
test/test2
test/test1
#把目录打包为 test.tar 文件
tar 命令也可以打包多个文件或目录，只要用空格分开即可。例如:
[root@localhost ~]# tar -cvf ana.tar anaconda-ks.cfg /tmp/
#把 anaconda-ks.cfg 文件和/tmp 目录打包成 ana.tar 文件包

【例 2】打包压缩目录。
我们已经解释过了，压缩命令不能直接压缩目录，我们就先用 tar 命令把目录打成数据包，然后再用 gzip 命令或 bzip2 命令压缩。例如：

[root@localhost ~]#ll -d test test.tar
drwxr-xr-x 2 root root 4096 6 月 17 21:09 test
-rw-r--r-- 1 root root 10240 6 月 18 01:06 test.tar
#我们之前已经把 test 目录打包成 test.tar 文件
[root@localhost ~]# gzip test.tar
[root@localhost ~]# ll test.tar.gz
-rw-r--r-- 1 root root 176 6 月 18 01:06 test.tar.gz
#gzip 命令会把 test.tar 压缩成 test.tar.gz
[root@localhost ~]# gzip -d test.tar.gz
#解压缩，把 test.tar.gz 解压缩为 test.tar
[root@localhost ~]# bzip2 test.tar
[root@localhost ~]# ll test.tar.bz2
-rw-r--r-- 1 root root 164 6 月 18 01:06 test.tar.bz2
#bzip2 命令会把 test.tar 压缩为 test.tar.bz2 格式
解打包命令格式
".tar"格式的解打包也需要使用 tar 命令，但是选项不太一样。命令格式如下：

[root@localhost ~]#tar [选项] 压缩包
选项：

- -x：解打包；
- -f：指定压缩包的文件名；
- -v：显示打包文件过程；

- -t：测试，就是不解打包，只是查看包中有哪些文件；
- -C 目录：指定解打包位置；

其实解打包和打包相比，只是把打包选项"-cvf"更换为"-xvf"。我们来试试：

[root@localhost ~]# tar -xvf anaconda-ks.cfg. tar
#解打包到当前目录下
如果使用"-xvf"选项，则会把包中的文件解压到当前目录下。如果想要指定解压位置，则需要使用"-C(大写)"选项。例如：

[root@localhost ~]# tar -xvf test.tar -C /tmp
#把文件包 test.tar 解打包到/tmp/目录下
如果只想查看文件包中有哪些文件，则可以把解打包选项"-x"更换为测试选项"-t"。例如：

[root@localhost ~]# tar -tvf test.tar
drwxr-xr-x root/root 0 2016-06-17 21:09 test/
-rw-r-r- root/root 0 2016-06-17 17:51 test/test3
-rw-r-r- root/root 0 2016-06-17 17:51 test/test2
-rw-r-r- root/root 0 2016-06-17 17:51 test/test1
#会用长格式显示 test.tar 文件包中文件的详细信息
".tar.gz"和".tar.bz2" 格式
你可能会觉得 Linux 实在太不智能了，一个打包压缩，居然还要先打包成".tar"格式，再压缩成".tar.gz"或".tar.bz2"格式。其实 tar 命令是可以同时打包压缩的，前面的讲解之所打包和压缩分开，是为了让大家了解在 Linux 中打包和压缩的不同。

使用 tar 命令直接打包压缩。命令格式如下：

[root@localhost ~]#tar [选项] 压缩包 源文件或目录
选项：

- -z：压缩和解压缩 ".tar.gz"格式
- -j：压缩和街压缩 ".tar.bz2"格式

【例 1】压缩与解压缩 ".tar.gz"格式。
我们先来看看如何压缩".tar.gz"格式：

[root@localhost ~]# tar -zcvf tmp.tar.gz /tmp/
#把/temp/目录直接打包压缩为".tar.gz"格式，通过"-z"来识别格式，"-cvf"和打包选项一致

解压缩也只是在解打包选项"-xvf"前面加了一个"-z"选项。

[root@localhost ~]# tar -zxvf tmp.tar.gz
#解压缩与解打包".tar.gz"格式
前面讲的选项"-C"用于指定解压位置、"-t"用于查看压缩包内容，在这里同样适用。

【例 2】压缩与解压缩".tar.bz2"格式。
和".tar.gz"格式唯一的不同就是"-zcvf"选项换成了 "-jcvf"。
[root@localhost ~]# tar -jcvf tmp.tar.bz2 /tmp/
#打包压缩为".tar.bz2"格式，注意压缩包文件名
[root@localhost ~]# tar -jxvf tmp.tar.bz2
#解压缩与解打包".tar.bz2"格式
把文件直接压缩成".tar.gz"和".tar.bz2"格式，才是 Linux 中最常用的压缩方式，这是大家一定要掌握的压缩和解压缩方法。

Linux shutdown 命令：关机和重启：

在早期的 Linux 系统中，应该尽量使用 shutdown 命令来进行关机和重启。因为在那时的 Linux 中，只有 shutdown 命令在关机或重启之前会正确地中止进程及服务，所以我们一直认为 shutdown 才是最安全的关机与重启命令。

而在现在的系统中，一些其他的命令（如 reboot）也会正确地中止进程及服务，但我们仍建议使用 shutdown 命令来进行关机和重启。

shutdown 命令的基本信息如下。

- 命令名称：shutdown。
- 英文原意：bring the system down。
- 所在路径：/sbin/shutdown。
- 执行权限：超级用户。
- 功能描述：关机和重启

命令格式
[root@localhost ~]# shutdown [选项] 时间 [警告信息]
选项:

- -c：取消已经执行的 shutdown 命令；
- -h：关机；
- -r：重启；

【例 1】重启与定时重启。
先来看看如何使用 shutdown 命令进行重启：
[root@localhost ~]# shutdown -r now
#重启, now 是现在重启的意思
[root@localhost ~]# shutdown -r 05:30
#指定时间重启，但会占用前台终端
[root@localhost ~]# shutdown -r 05:30 &
#把定义重启命令放入后台，&是后台的意思
[root@localhost ~]# shutdown -c
//取消定时重启
[root@localhost ~]# shutdown -r +10
#10 分钟之后重启

【例 2】关机和定时关机。

```
[root@localhost ~]# shutdown -h now
#现在关机
[root@localhost ~]# shutdown -h 05:30
#指定时间关机
```

Linux 关机和重启命令（超详解）：

说到关机和重启，很多人认为，重要的服务器（比如银行的服务器、电信的服务器）如果重启了，则会造成大范围的灾难。笔者在这里解释一下。

首先，就算是银行或电信的服务器，也不是不需要维护，而是依靠备份服务器代替。其次，每个人的经验都是和自己的技术成长环境息息相关的。比如笔者是游戏运维出身，而游戏又是数据为王，所以一切操作的目的就是保证数据的可靠和安全。这时，有计划的重启远比意外岩机造成的损失要小得多，所以定义重启是游戏运维的重要手段。

shutdown 命令

在早期的 Linux 系统中，应该尽量使用 shutdown 命令来进行关机和重启。因为在那时的 Linux 中，只有 shutdown 命令在关机或重启之前会正确地中止进程及服务，所以我们一直认为 shutdown 才是最安全的关机与重启命令。

而在现在的系统中，一些其他的命令（如 reboot）也会正确地中止进程及服务，但我们仍建议使用 shutdown 命令来进行关机和重启。

shutdown 命令的基本信息如下。

- 命令名称：shutdown。
- 英文原意：bring the system down。
- 所在路径：/sbin/shutdown。
- 执行权限：超级用户。
- 功能描述：关机和重启

命令格式

```
[root@localhost ~]# shutdown [选项] 时间 [警告信息]
```

选项:

- -c：取消已经执行的 shutdown 命令；
- -h：关机；
- -r：重启；

【例 1】重启与定时重启。

先来看看如何使用 shutdown 命令进行重启：

```
[root@localhost ~]# shutdown -r now
#重启, now 是现在重启的意思
[root@localhost ~]# shutdown -r 05:30
#指定时间重启，但会占用前台终端
[root@localhost ~]# shutdown -r 05:30 &
#把定义重启命令放入后台，&是后台的意思
[root@localhost ~]# shutdown -c
```

//取消定时重启
[root@localhost ~]# shutdown -r +10
#10 分钟之后重启
【例 2】关机和定时关机。
[root@localhost ~]# shutdown -h now
#现在关机
[root@localhost ~]# shutdown -h 05:30
#指定时间关机
reboot 命令
在现在的系统中，reboot 命令也是安全的，而且不需要加入过多的选项。
[root@localhost ~]# reboot
#重启
halt 和 poweroff 命令
这两个都是关机命令，直接执行即可。
[root@localhost ~】# halt
#关机
[root@localhost ~】# poweroff
#关机
init 命令
init 是修改 Linux 运行级别的命令，也可以用于关机和重启。
[root@localhost~]# init 0
#关机，也就是调用系统的 0 级别
[root@localhost ~】# init 6
#重启，也就是调用系统的 6 级别

Linux ifconfig 命令：配置网络接口:

ifconfig 是 Linux 中查看和临时修改 IP 地址的命令，其基本信息如下：

- 命令名称：ifconfig。
- 英文原意：configure a network interface。
- 所在路径：/sbin/ifconfig。
- 执行权限：超级用户。
- 功能描述：配置网络接口。

查看 IP 地址信息
ifconfig 命令最主要的作用就是查看 IP 地址的信息，直接输入 ifconfig 命令即可。
[root@localhost ~]# ifconfig
eth0 Link encap:Ethernet HWaddr 00:OC:29:C5:FB:AA #eth0 网卡信息 网络类型为以太网 MAC 地址
inet addr:192.168.44.3 Beast:192.168.44.255 Mask:255.255.255.0
#IP 地址 广播地址 子网掩码
inet6 addr: fe80::20c:29ff:fec5:fbaa/64 Scope:Link #IPv6 的地址（目前不生效）
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 #网络参数 最大传输单元数据包转送次数
RX packets:881 errors:0 dropped:0 overruns:0 frame:0
#接收到的数据包惰况

TX packets:8S3 errors:0 dropped:0 overruns:0 carrier:0
#发送的数据包情况
collisions:0 txqueuelen:1000
#数据包碰撞 数据缓冲区长度
RX bytes:82229 (80.3 KiB) TX bytes:273463 (267.0 KiB)
#接收包的大小 发送包的大小
Interrupt:19 Base address:0x2000
#IRQ 中街 内存地址
lo Link encap:Local Loopback
#本地回环网卡信息
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric：1
RX packets:12 errors：0 dropped:0 overruns:0 frame:0
TX packets:12 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:0
RX bytes：840 (840.0 b) TX bytes:840 (840.0 b)
ifconfig 命令主要用于查看 IP 地址、子网掩码和 MAC 地址这三类信息，其他信息我们有所了解即可。

lo 网卡是 Loopback 的缩写，也就是本地回环网卡，这个网卡的 IP 地址是 127.0.0.1。它只代表我们的网络协议正常，就算不插入网线也可以 ping 通，所以基本没有实际使用价值，大家了解一下即可。

临时配置 IP 地址
ifconfig 命令除可以查看 IP 地址之外，还可以临时配置 IP 地址，但是一旦重启，IP 地址就会失效，所以我们还是应该使用 setup 命令进行 IP 地址配置。使用 ifconfig 命令临时配置 IP 地址的示例如下：
[root@localhost ~]#ifconfig eth0 192.168.44.3
#配置 IP 地址，不指定子网掩码就会使用标准子网掩码
[root@localhost ~]#ifconfig eth0 192.168.44.3 netmask 255.255.255.0
#配置 IP 地址，同时配置子网掩码

Linux ping 命令：向网络主机发送 ICMP 请求:

ping 是常用的网络命令，主要通过 ICMP 协议进行网络探测，测试网络中主机的通信情况。

ping 命令的基本信息如下。

- 命令名称：ping。
- 英文原意：send ICMP ECHO_REQUEST to network hosts。
- 所在路径：/bin/ping。
- 执行权限：所有用户。
- 功能描述：向网络主机发送 ICMP 请求。

命令的基本格式如下：
[root@localhost ~]# ping [选项] IP
选项：

- -b: 后面加入广播地址，用于对整个网段进行探测；
- -c 次数： 用于指定 ping 的次数；
- -s 字节： 指定探测包的大小；

【例 1】探测与指定主机通信。

```
[root@localhost ~]#ping 192.168.103.151
PING 192.168.103.151 (192.168.103.151) 56(84) bytes of data.
64 bytes from 192.168.103.151: icmp_seq=1 ttl=128 time=0.300 ms
64 bytes from 192.168.103.151: icmp_seq=2 ttl=128 time=0.481 ms
…省略部分内容...
#探测与指定主机是否通信
```

Linux 是一个比较实在的操作系统，这个 ping 命令如果不使用"Ctrl+C"快捷键强行中止，就会一直 ping 下去，直到天荒地老……

【例 2】指定 ping 的次数。

既然 ping 这么"实在"，如果不想一直 ping 下去，则可以使用"-c"选项指定 ping 的次数。例如：

```
[root@localhost ~]# ping -c 3 192.168.103.151
#只探测 3 次，就中止 ping 命令
```

【例 3】探测网段中的可用主机。

在 ping 命令中，可以使用"-b"选项，后面加入广播地址，探测整个网段。我们可以使用这个选项知道整个网络中有多少主机是可以和我们通信的，而不用一个一个 IP 进行探测。例如：

```
[root@localhost ~]# ping -b -c 3 192.168.103.255
WARNING: pinging broadcast address
PING 192.168.103.255 (192.168.103.255) 56(84) bytes of data.
64 bytes from 192.168.103.199: icmp_seq=1 ttl=64 time=1.95ms
64 bytes from 192.168.103.168: icmp_seq=1 ttl=64 time=1.97ms(DUP!)
64 bytes from 192.168.103.252: icmp_seq=1 ttl=64 time=2.29ms(DUP!)
…省略部分内容...
#探测 192.168.103.0/24 网段中有多少可以通信的主机
```