

文件处理命令

剖析一个文件列表

ls 命令用来查看用户有执行权限的任意目录中的文件列表，该命令有许多有趣的选项。例如：

```
bordercolorlight="black" bordercolordark="#FFFFFF"
align="center" border="1" cellpadding="2" cellspacing="0"
width="400"
```

```
$ ls -liah *
22684 -rw-r--r--    1 blucher  users          952
Dec 28 18:43 .profile
19942 -rw-r--r--    1 scalish  users           30
Jan  3 20:00 test2.out
925 -rwxr-xr-x    1 scalish  users        378 Sep
2  2002 test.sh
```

上面的列表显示 8 列：

- 第 1 列指示文件的 inode，因为我们使用了 -i 选项。剩下的列通过 -l 选项来进行正常显示。
- 第 2 列显示文件类型和文件访问权限。
- 第 3 列显示链接数，包括目录。
- 第 4 和第 5 列显示文件的所有者和组所有者。这里，所有者 "blucher" 属于组 "users"。
- 第 6 列显示文件大小（单位为所显示的单位，而不是默认的字节数，因为我们使用了 -h 选项）。
- 第 7 列显示日期（它看起来像是三列），包括月、日和年，以及当天的时间。
- 第 8 列显示文件名。在选项列表中使用 -a 将使列表中包含隐藏文件（如 .profile）的列表。

处理文件

可以移动 (mv)、复制 (cp) 或删除 (rm) 文件和目录。明智地使用 -i 选项，以获得确认通常是个不错的主意。

```
$ cp -i ls.out ls2.out
```

```
cp:overwrite `ls2.out'?
```

mv 命令允许使用 -b 选项，这将在移动文件之前作一个备份拷贝。rm 和 cp 接受功能强大但却危险的 -r 选项，它将在一个目录和它的文件上递归执行。

```
$ rm -ir Test
```

```
rm:descend into directory `Test'? y
```

可以用 mkdir 来创建目录，用 rmdir 来删除目录。不过，因为不能用 rmdir 来删除包含文件的目录，所以使用 rm 加 -r 选项通常要更方便些。

出于安全的原因，所有的文件都有所有权和保护。文件访问权限（或文件模式）包含之前提到的相同的 10 个字符：

- 第一个字符指示文件类型。最常见的是 - 代表文件，d 代表目录，而 l 代表链接。
- 接下来的 9 个字符是三种用户类别的访问权限：文件所有者（字符 2-4），用户组（5-7）和其它（8-10），其中 r 表示读权限，w 表示写权限，x 指示一个文件上的执行权限。破折号 - 如果出现在这九个位置的任意一个，则指示这个操作对该类别的用户禁止。

可以使用 chmod 命令通过字符符号或二进制掩码来设置访问权限。要使用二进制掩码，必须将三个权限组的字符表示转换成二进制格式，然后转换成八进制格式：

用户类别： 所有者 用户组 其它

字符表示： rwx r-x r--

二进制表示： 111 101 100

八进制表示： 7 5 4

将写权限赋予用户组，您可以使用：

```
chmod g+w test.sh or chmod 774 test.sh
```

用 umask 命令，在 /etc/init.dev 文件中进行系统范围内的、或在 .profile 文件中进行的本地的文件权限默认设置。这个命令指示用 777

减去这个数字来获取默认的权限：

```
$ umask 022
```

这将为用户创建的所有新文件生成一个默认的文件权限 755。

可以用 `chown` 来修改文件的所有权：

```
$ chown blucher ls.out
```

这里，`blucher` 是新的文件所有者。类似地，组成员资格将按以下方式修改：

```
$ chgrp devgrp ls.out
```

这里，`devgrp` 是新的用户组。

`ls` 不提供哪些文件是文本的，哪些文件是二进制的的信息。要了解这个信息，您可以使用 `file *` 命令。

重命名文件

赋予一个文件多个名字的两流行方法是利用链接和 `alias` 命令。

`Alias` 可以用来为一个更长的命令重新起一个更方便的名字：

```
$ alias ll='ls -l'
```

```
$ ll
```

注意单引号的使用，这使得 `BASH` 将该项目传递给别名，而不是自己估计它。别名还可用作较长的路径名的缩写：

```
$ alias jdev9i=/jdev9i/jdev/bin/jdev
```

关于 `alias` 和它的反命令 `unalias` 的更多信息，请查看 `BASH` 的 `man` 页面的 "SHELL BUILTIN COMMANDS" 子部分。在最后一个例子中，定义了一个环境变量来实现相同的结果。

```
$ export JDEV_HOME=/jdev9i/jdev/bin/jdev
```

```
$ echo $JDEV_HOME
```

```
/jdev9i/jdev/bin/jdev
```

```
$ $JDEV_HOME
```

链接允许几个文件名引用单个源文件，格式如下：

```
ln [-s] fileyouwanttolinkto newname
```

单独的 `ln` 命令创建到文件的一个硬链接，而使用 `-s` 选项来创建一个符号链接。简而言之，一个硬链接几乎不能和原始的文件区分开（除了这两个文件的 `inode` 将相同之外）。符号链接较容易区分，因为它们出现在一个长的文件列表中，并用 `->` 来指示源文件，`l` 指示文件类型。