

Linux文件与目录管理

我们知道Linux的目录结构为树状结构，最顶级的目录为根目录 /。

其他目录通过挂载可以将它们添加到树中，通过解除挂载可以移除它们。

在开始本教程前我们需要先知道什么是绝对路径与相对路径。

- **绝对路径：**

路径的写法，由根目录 / 写起，例如： /usr/share/doc 这个目录。

- **相对路径：**

路径的写法，不是由 / 写起，例如由 /usr/share/doc 要到 /usr/share/man 底下时，可以写成： cd ../man 这就是相对路径的写法啦！

处理目录的常用命令

接下来我们就来看几个常见的处理目录的命令吧：

- ls: 列出目录
- cd: 切换目录
- pwd: 显示目前的目录
- mkdir: 创建一个新的目录
- rmdir: 删除一个空的目录
- cp: 复制文件或目录
- rm: 移除文件或目录

你可以使用 *man [命令]* 来查看各个命令的使用文档，如： man cp。

ls (列出目录)

在Linux系统当中， ls 命令可能是最常被运行的。

语法：

```
[root@www ~]# ls [-aAdfFhilnrRSt] 目录名称
[root@www ~]# ls [--color={never,auto,always}] 目录名称
[root@www ~]# ls [--full-time] 目录名称
```

选项与参数：

- -a：全部的文件，连同隐藏档(开头为 . 的文件)一起列出来(常用)
- -d：仅列出目录本身，而不是列出目录内的文件数据(常用)
- -l：长数据串列出，包含文件的属性与权限等等数据；(常用)

将家目录下的所有文件列出来(含属性与隐藏档)

```
[root@www ~]# ls -al ~
```

cd (切换目录)

cd是Change Directory的缩写，这是用来变换工作目录的命令。

语法：

```
cd [相对路径或绝对路径]
```

#使用 mkdir 命令创建 runoob 目录

```
[root@www ~]# mkdir runoob
```

#使用绝对路径切换到 runoob 目录

```
[root@www ~]# cd /root/runoob/
```

#使用相对路径切换到 runoob 目录

```
[root@www ~]# cd ./runoob/
```

表示回到自己的家目录，亦即是 /root 这个目录

```
[root@www runoob]# cd ~
```

表示去到目前的上一级目录，亦即是 /root 的上一级目录的意思；

```
[root@www ~]# cd ..
```

接下来大家多操作几次应该就可以很好的理解 cd 命令的。

pwd (显示目前所在的目录)

pwd 是 **Print Working Directory** 的缩写，也就是显示目前所在目录的命令。

```
[root@www ~]# pwd [-P]
```

选项与参数：

- **-P**：显示出确实的路径，而非使用连结 (link) 路径。

实例：单纯显示出目前的工作目录：

```
[root@www ~]# pwd
```

```
/root    <== 显示出目录啦~
```

实例显示出实际的工作目录，而非连结档本身的目录名而已。

```
[root@www ~]# cd /var/mail    <==注意，/var/mail是一个连结档
```

```
[root@www mail]# pwd
```

```
/var/mail    <==列出目前的工作目录
```

```
[root@www mail]# pwd -P
```

```
/var/spool/mail    <==怎么回事？有没有加 -P 差很多~
```

```
[root@www mail]# ls -ld /var/mail
```

```
lrwxrwxrwx 1 root root 10 Sep  4 17:54 /var/mail -> spool/mail
```

看到这里应该知道为啥了吧？因为 /var/mail 是连结档，连结到 /var/spool/mail

所以，加上 pwd -P 的选项后，会不以连结档的数据显示，而是显示正确的完整路径啊！

mkdir (创建新目录)

如果想要创建新的目录的话，那么就使用mkdir (make directory)吧。

语法：

```
mkdir [-mp] 目录名称
```

选项与参数：

- -m：配置文件的权限喔！直接配置，不需要看默认权限 (umask) 的脸色~
- -p：帮助你直接将所需要的目录(包含上一级目录)递归创建起来！

实例：请到/tmp底下尝试创建数个新目录看看：

```
[root@www ~]# cd /tmp
[root@www tmp]# mkdir test    <==创建一名为 test 的新目录
[root@www tmp]# mkdir test1/test2/test3/test4
mkdir: cannot create directory `test1/test2/test3/test4':
No such file or directory    <== 没办法直接创建此目录啊！
[root@www tmp]# mkdir -p test1/test2/test3/test4
```

加了这个 -p 的选项，可以自行帮你创建多层目录！

实例：创建权限为 **rwX--X--X** 的目录。

```
[root@www tmp]# mkdir -m 711 test2
[root@www tmp]# ls -l
drwxr-xr-x  3 root  root 4096 Jul 18 12:50 test
drwxr-xr-x  3 root  root 4096 Jul 18 12:53 test1
drwx--x--x  2 root  root 4096 Jul 18 12:54 test2
```

上面的权限部分，如果没有加上 -m 来强制配置属性，系统会使用默认属性。

如果我们使用 -m，如上例我们给予 -m 711 来给予新的目录 drwx--x--x 的权限。

rmdir (删除空的目录)

语法：

```
rmdir [-p] 目录名称
```

选项与参数：

- **-p**：连同上一级『空的』目录也一起删除

删除 runoob 目录

```
[root@www tmp]# rmdir runoob/
```

将 mkdir 实例中创建的目录(/tmp 底下)删除掉！

```
[root@www tmp]# ls -l    <==看看有多少目录存在？
drwxr-xr-x  3 root  root 4096 Jul 18 12:50 test
drwxr-xr-x  3 root  root 4096 Jul 18 12:53 test1
drwx--x--x  2 root  root 4096 Jul 18 12:54 test2
[root@www tmp]# rmdir test    <==可直接删除掉，没问题
[root@www tmp]# rmdir test1    <==因为尚有内容，所以无法删除！
rmdir: `test1': Directory not empty
[root@www tmp]# rmdir -p test1/test2/test3/test4
[root@www tmp]# ls -l    <==您看看，底下的输出中test与test1不见了！
drwx--x--x  2 root  root 4096 Jul 18 12:54 test2
```

利用 -p 这个选项，立刻就可以将 test1/test2/test3/test4 一次删除。

不过要注意的是，这个 rmdir 仅能删除空的目录，你可以使用 rm 命令来删除非空目录。

cp (复制文件或目录)

cp 即拷贝文件和目录。

语法：

```
[root@www ~]# cp [-adfilprsu] 来源档(source) 目标档(destination)
[root@www ~]# cp [options] source1 source2 source3 .... directory
```

选项与参数：

- **-a**：相当於 -pdr 的意思，至於 pdr 请参考下列说明；(常用)
- **-d**：若来源档为连结档的属性(link file)，则复制连结档属性而非文件本身；
- **-f**：为强制(force)的意思，若目标文件已经存在且无法开启，则移除后再尝试一次；
- **-i**：若目标档(destination)已经存在时，在覆盖时会先询问动作的进行(常用)
- **-l**：进行硬式连结(hard link)的连结档创建，而非复制文件本身；
- **-p**：连同文件的属性一起复制过去，而非使用默认属性(备份常用)；
- **-r**：递归持续复制，用於目录的复制行为；(常用)
- **-s**：复制成为符号连结档(symbolic link)，亦即『捷径』文件；
- **-u**：若 destination 比 source 旧才升级 destination ！

用 root 身份，将 root 目录下的 .bashrc 复制到 /tmp 下，并命名为 bashrc

```
[root@www ~]# cp ~/.bashrc /tmp/bashrc
[root@www ~]# cp -i ~/.bashrc /tmp/bashrc
cp: overwrite `/tmp/bashrc'? n <==n不覆盖，y为覆盖
```

rm (移除文件或目录)

语法：

```
rm [-fir] 文件或目录
```

选项与参数：

- **-f**：就是 force 的意思，忽略不存在的文件，不会出现警告信息；
- **-i**：互动模式，在删除前会询问使用者是否动作
- **-r**：递归删除啊！最常用在目录的删除了！这是非常危险的选项！！

将刚刚在 cp 的实例中创建的 bashrc 删除掉！

```
[root@www tmp]# rm -i bashrc
rm: remove regular file `bashrc'? y
```

如果加上 -i 的选项就会主动询问喔，避免你删除到错误的档名！

mv (移动文件与目录，或修改名称)

语法：

```
[root@www ~]# mv [-fiu] source destination
[root@www ~]# mv [options] source1 source2 source3 .... directory
```

选项与参数：

- **-f**：force 强制的意思，如果目标文件已经存在，不会询问而直接覆盖；
- **-i**：若目标文件(destination)已经存在时，就会询问是否覆盖！
- **-u**：若目标文件已经存在，且 source 比较新，才会升级(update)

复制一文件，创建一目录，将文件移动到目录中

```
[root@www ~]# cd /tmp
[root@www tmp]# cp ~/.bashrc bashrc
[root@www tmp]# mkdir mvtest
[root@www tmp]# mv bashrc mvtest
```

将某个文件移动到某个目录去，就是这样做！

将刚刚的目录名称更名为 mvtest2

```
[root@www tmp]# mv mvtest mvtest2
```

Linux 文件内容查看

Linux系统中使用以下命令来查看文件的内容：

- cat 由第一行开始显示文件内容
- tac 从最后一行开始显示，可以看出 tac 是 cat 的倒著写！
- nl 显示的时候，顺道输出行号！
- more 一页一页的显示文件内容
- less 与 more 类似，但是比 more 更好的是，他可以往前翻页！
- head 只看头几行
- tail 只看尾巴几行

你可以使用 *man [命令]* 来查看各个命令的使用文档，如：man cp。

cat

由第一行开始显示文件内容

语法：

```
cat [-AbEnTv]
```

选项与参数：

- -A：相当於 -vET 的整合选项，可列出一些特殊字符而不是空白而已；
- -b：列出行号，仅针对非空白行做行号显示，空白行不标行号！
- -E：将结尾的断行字节 \$ 显示出来；
- -n：列印出行号，连同空白行也会有行号，与 -b 的选项不同；
- -T：将 [tab] 按键以 ^I 显示出来；
- -v：列出一些看不出来的特殊字符

检看 /etc/issue 这个文件的内容：

```
[root@www ~]# cat /etc/issue
CentOS release 6.4 (Final)
Kernel \r on an \m
```

tac

tac与cat命令刚好相反，文件内容从最后一行开始显示，可以看出 tac 是 cat 的倒着写！如：

```
[root@www ~]# tac /etc/issue

Kernel \r on an \m
CentOS release 6.4 (Final)
```

nl

显示行号

语法：

nl [-bnw] 文件

选项与参数：

- -b：指定行号指定的方式，主要有两种：

-b a：表示不论是否为空行，也同样列出行号(类似 cat -n)；

-b t：如果有空行，空的那一行不要列出行号(默认值)；

- -n：列出行号表示的方法，主要有三种：

-n ln：行号在荧幕的最左方显示；

-n rn：行号在自己栏位的最右方显示，且不加 0；

-n rz：行号在自己栏位的最右方显示，且加 0；

- -w：行号栏位的占用的位数。

实例一：用 nl 列出 /etc/issue 的内容

```
[root@www ~]# nl /etc/issue
1  CentOS release 6.4 (Final)
2  Kernel \r on an \m
```

more

一页一页翻动

```
[root@www ~]# more /etc/man.config
#
# Generated automatically from man.conf.in by the
# configure script.
#
# man.conf from man-1.6d
.... (中间省略) ....
--More--(28%)  <== 重点在这一行喔！你的光标也会在这里等待你的命令
```

在 more 这个程序的运行过程中，你有几个按键可以按的：

- 空白键 (space)：代表向下翻一页；
- Enter：代表向下翻『一行』；
- /字符串：代表在这个显示的内容当中，向下搜寻『字符串』这个关键字；
- :f：立刻显示出档名以及目前显示的行数；
- q：代表立刻离开 more，不再显示该文件内容。
- b 或 [ctrl]-b：代表来回翻页，不过这动作只对文件有用，对管线无用。

less

一页一页翻动，以下实例输出/etc/man.config文件的内容：

```
[root@www ~]# less /etc/man.config
#
# Generated automatically from man.conf.in by the
# configure script.
#
# man.conf from man-1.6d
```

.... (中间省略)

: <== 这里可以等待你输入命令!

less运行时可以输入的命令有:

- 空白键 : 向下翻动一页;
- [pagedown]: 向下翻动一页;
- [pageup] : 向上翻动一页;
- /字串 : 向下搜寻『字串』的功能;
- ?字串 : 向上搜寻『字串』的功能;
- n : 重复前一个搜寻 (与 / 或 ? 有关!)
- N : 反向的重复前一个搜寻 (与 / 或 ? 有关!)
- q : 离开 less 这个程序;

head

取出文件前面几行

语法:

```
head [-n number] 文件
```

选项与参数:

- -n : 后面接数字, 代表显示几行的意思

```
[root@www ~]# head /etc/man.config
```

默认的情况中, 显示前面 10 行! 若要显示前 20 行, 就得要这样:

```
[root@www ~]# head -n 20 /etc/man.config
```

tail

取出文件后面几行

语法:

```
tail [-n number] 文件
```

选项与参数:

- -n : 后面接数字, 代表显示几行的意思
- -f : 表示持续侦测后面所接的档名, 要等到按下[ctrl]-c才会结束tail的侦测

```
[root@www ~]# tail /etc/man.config
```

默认的情况中, 显示最后的十行! 若要显示最后的 20 行, 就得要这样:

```
[root@www ~]# tail -n 20 /etc/man.config
```