

# Linux系统管理

南通师范高等专科学校 朱亚林



# 本章导言



- 在Linux系统中，万物皆文件。不仅普通的文件，目录、字符设备、块设备、套接字等在 Unix/Linux 中都是以文件被对待。因此，要深入探究Linux的奥秘，必须要掌握好Linux文件的操作方法。
- 本章内容将就
  - 目录和文件的创建、复制、移动、删除等操作进行了详细讲解；
  - 认识两种“链接”类型；
  - 深入探讨文件阅读以及查找的方法。



## 第4章

# 开启Linux文件操作的大门

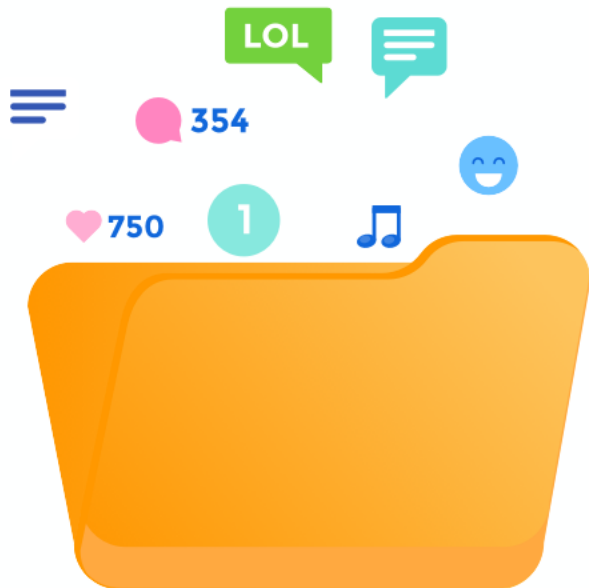
---

## 4.1 目录的操作

---



结合Windows  
应用，总结目录  
的操作有哪些？



## 4.1.1 新建目录

- ✓ 新建目录使用到的命令是**mkdir** (make directory) , 它的使用格式如下:

```
mkdir [-p] 目录名
```

选项-p的作用是：确保其后跨级创建的所有层级的目录名称存在，不存在的就建



## 4.1.1 新建目录

案例：

- 在用户主目录下创建一个叫做Mydir的目录
- 在用户主目录下创建一个叫做Mydir2的目录，并且在这个目录中创建了一个叫做Mydir3的目录



## 4.1.2 删除目录

- ✓ 删除目录可以有两种方法，这里介绍的是rmdir（remove directory）。它的使用格式如下：

```
rmdir [-p] 目录名
```

-p一般用于删除多级目录时，是指当子目录被删除后使它也成为空目录的话，则一并删除。





## 4.1.2 删除目录

案例：

- 将上例中创建的Mydir进行删除。
- 将上例中创建的Mydir3、Mydir2进行删除



## 4.1.3 复制目录

- ✓ 在Linux中目录的复制和文件复制的命令是同一个命令cp（copy），它的使用格式如下

```
cp [选项] 源文件/源目录 目标文件/目标目录
```



## 4.1.3 复制目录

参数	作用
-a	相当于 -d、-p、-r 选项的集合；
-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；
-f	覆盖已经存在的目标文件而不给出提示；
-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；
-u	若目标文件比源文件有差异，则使用该选项可以更新目标文件，此选项可用于对文件的升级和备用。



## 4.1.3 复制目录

案例：

- ✓ 运行以下代码，建立环境：`bash <(wget -qO- http://t.edutest.fun/j)`
- ✓ 进入用户主目录中LinuxStudy，将其中的Chapter4复制一份，名称为Chapter4.bak



## 4.1.4 移动目录

- ✓ 移动操作，顾名思义，就是将一个目录从当前位置移动到指定的新位置。一般使用mv (move file) 命令。
- ✓ mv命令的格式如下：

```
mv [选项] 源文件/源目录 目标文件/目标目录
```



## 4.1.4 移动目录

- ✓ 移动操作，顾名思义，就是将一个目录从当前位置移动到指定的新位置。一般使用mv (move file) 命令。
- ✓ mv命令的格式如下：

```
mv [选项] 源文件/源目录 目标文件/目标目录
```



## 4.1.4 移动目录

- ✓ 移动操作，顾名思义，就是将一个目录从当前位置移动到指定的新位置。一般使用mv (move file) 命令。
- ✓ mv命令的格式如下：

```
mv [选项] 源文件/源目录 目标文件/目标目录
```



## 4.1.4 移动目录

✓ mv命令的选项如下表所示

选项	作用
-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	显示文件或目录的移动过程；
-u	若目标文件已经存在，但两者相比，源文件更新，则会对目标文件进行升级；





## 4.1.4 移动目录

案例：

- 将用户主目录下的test.txt移动到LinuxStudy/Chapter4中
- 尝试一下，在同一目录下移动对象是什么效果？



## 4.1.4 移动目录

- ✓ 其实重命名也是移动目录的一种应用



## 4.2 文件的操作

---

## 4.2.1 文件的建立

### 1. 使用输出重定向符（“>”）

- ✓ 标准输出重定向符允许创建一个 0KB 的空文件。它通常用于重定向一个命令的输出到一个新文件中。在没有命令的情况下使用重定向符号时，它会创建一个文件。但是它不允许你在创建文件时向其中输入任何文本。

```
zz@debian:~$ > test.txt
```



## 4.2.1 文件的建立

### 2. 使用touch命令创建文件

- ✓ touch 命令常用于将每个文件的访问和修改时间更新为当前时间。如果指定的文件名不存在，将会创建一个新的文件。touch 不允许我们在创建文件的同时向其中输入一些文本。它默认创建一个 0KB 的空文件。

```
zz@debian:~$ touch test-1.txt
```



## 4.2.1 文件的建立

### 3. 使用 echo 命令创建文件

echo命令是Linux中最基本和最常用的命令之一。传递给echo的参数被打印到标准输出中。

echo命令的格式如下：

```
echo [选项] [参数]
```



## 4.2.1 文件的建立

### 3. 使用 echo 命令创建文件

echo中常见的选项如下表所示

选项	作用
-n	取消尾随换行符
-e	解释以下反斜杠转义字符： \ 显示反斜杠字符 \a 警报 (BEL) \b 显示退格字符 \c 禁止任何进一步的输出 \e 显示转义字符 \f 显示窗体提要字符 \n 显示新行 \r 显示回车 \t 显示水平标签 \v 显示垂直标签
-E	禁用转义字符的解释 (默认值)



## 4.2.1 文件的建立

3. 使用 echo 命令创建文件

案例：

- 使用echo将信息输出到屏幕
- 使用echo将信息输出到文件





## 4.2.2 文件的删除

Linux中删除文件可以使用rm (remove) 命令。rm不仅适用于文件，也适用于目录。它会将目录或文件从系统中永久删除。rm命令的格式如下：

```
rm [选项] [参数]
```



## 4.2.2 文件的删除

rm命令常见选项如下表所示：

选项	作用
-i	删除前逐一询问确认。
-f	即使原档案属性设为只读，也直接删除，无需逐一确认。
-r	删除指定目录中所有的内容，包括子目录及其中的所有内容



## 4.2.2 文件的删除

案例：使用rm删除目录及文件

- 通过ls命令查询一下待删除目录的情况
- 使用rm命令删除目标目录及文件

**重点提醒**，不要使用“`sudo rm -rf`”这样的命令去操作任何你不熟悉的目录



## 4.2.3 文件的复制

- 文件的复制使用到的命令之前已经做过介绍，和目录一样，都是cp。
- 使用cp复制文件时，被复制的文件可以是一个，也可以是N个。如果是一个，那么第二个参数即是复制后的文件名；如果被复制的文件是多个，那么最后一个参数应该是前N个文件将要复制到的目录名。
- 在复制文件的时候，可以给cp加上相应的选项，-i表示复制文件时，如果目标位置已经存在同名文件了，会给出是否覆盖的提示，确认后再进行；-f与-i正好相反，在执行时不会给出提示，如果已存在同名文件，则覆盖。



## 4.2.3 文件的复制

案例：

- 使用cp命令复制单个文件
- 使用cp命令复制多个文件



## 4.2.4 文件的移动

- 文件的移动使用的命令与目录的移动是一致的，也是mv。
- 其操作方式与cp类似，也是可以移动一个或多个文件。
- 如果mv之后有多个参数（文件名），最后一个参数必须是一个目录名，表示将之前的文件移动到该目录中；如果mv之后的两个参数都是文件名，那么表示文件的重命名。



## 4.2.4 文件的移动

案例：

- 移动多个文件到新的目录
- 对文件进行重命名



## 4.3 文件信息的查看

---

对于文件的信息，除了可以使用ls来查看大小、所有者、修改时间等，还可以去查看文件属于什么类型，统计文件中的字数等。



## 4.3.1 file命令

- file命令可以确定文件的类型。在Windows中，一般通过文件的后缀名来判定文件是属于哪种类型，如：docx、mp3、avi等；在Linux中则不同，它并不强调后缀名。
- file命令的格式如下：

```
file [选项] 文件名
```



## 4.3.1 file命令

- file命令的常用选项如下表所示

选项	作用
-b	列出文件辨识结果时，不显示文件名称。
-c	详细显示指令执行过程，便于排错或分析程序执行的情形
-f	列出文件中文件名的文件类型
-F	使用指定分隔符号替换输出文件名后的默认的": "分隔符。
-i	输出mime类型的字符串
-L	查看对应软链接对应文件的文件类型
-z	尝试去解读压缩文件的内容



## 4.3.1 file命令

案例：

使用file命令查看文件类型



## 4.3.2 wc命令

- wc命令是用来计算文件中的行数、字数和字符数。如果不指定文件名称、或是所给予的文件名为 "-", 则wc指令会从标准输入设备读取数据。
- wc命令的格式如下:

```
wc [选项] 文件名
```



## 4.3.2 wc命令

- wc命令常用的选项如下表所示

选项	作用
-c	显示字符数
-l	显示行数
-w	显示字数

**需要注意的是**，如果不指定任何选项，则显示全部三个数字，即：行数、字数、字符数



## 4.3.2 wc命令

案例：

- 使用wc统计文件字数



## 4.4 文件的阅读

---

终端里的每条命令都对应着一个工具，每个工具都专注于完成一项工作。这是Unix在设计之初的哲学理念，同样也被延伸到了Linux系统上。在Linux终端中，阅读文件需要借助于相应的命令完成。当然，根据阅读情境不一样，可以选择不同的命令。

## 4.4.1 cat命令

- 使用cat读取文件时，会将文件中所有的信息一次性读出，且无法翻页。这对于体量较大的文件不是非常友好





## 4.4.2 more命令

- more命令是Linux的一个快速文件浏览工具，对于体量较大的文件有着较为友好的处理界面，它会将内容分屏显示，根据用户的需求进行翻页。
- more命令的格式如下：

```
more [选项] [-num] [+/pattern] [+linenum] 文件名
```



## 4.4.2 more命令

- more命令的常用选用如下表所示：

选项	作用
-d	在页面下方给予用户相应的操作提示
-f	计算行数时，以实际上的行数，而非自动换行过后的行数
-p	通过清除窗口而不是滚屏来对文件进行换页
-s	把连续的多个空行显示为一行
-num	一次显示的行数
+/pattern	在每个文档显示前搜寻该字串（pattern），然后从该字串之后开始显示
+num	从第num 行开始显示



## 4.4.2 more命令

在文件查阅过程中，常用的交互式命令如下：

按下空格（Space）表示前进一屏；

按下回车（Enter）表示前进一行；

按下“b”键表示回退一屏；

按下“=”显示当前行号；

按下“h”键显示帮助信息；

按下“q”键退出more程序。



## 4.4.2 more命令

案例：

- 使用more显示文件



## 4.4.3 less命令

- less命令与more命令非常相似，可以随意浏览文件，支持翻页和搜索，支持向上翻页和向下翻页。
- less命令的格式如下：

```
less [选项] 文件名
```



## 4.4.3 less命令

- less的常用选项如下表所示

选项	作用
-N	显示每行的行号。
-S	行过长时将超出部分舍弃。
-e	当文件显示结束后，自动离开。
-g	只标志最后搜索到的关键同。
-Q	不使用警告音。
-i	忽略搜索时的大小写。
-m	显示类似 more 命令的百分比。
-f	强迫打开特殊文件，比如外围设备代号、目录和二进制文件。
-s	显示连续空行为一行。
-b <缓冲区大小>	设置缓冲区的大小。
-o <文件名>	将 less 输出的内容保存到指定文件中。

## 4.4.3 less命令

- less中常用的交互式命令如下:

/字符串: 向下搜索"字符串"的功能

? 字符串: 向上搜索"字符串"的功能

n: 重复前一个搜索 (与 / 或 ? 有关)

N: 反向重复前一个搜索 (与 / 或 ? 有关)

b 向上翻一页

d 向后翻半页

h 显示帮助界面

Q 退出less 命令

u 向前滚动半页

y 向前滚动一行

空格键 滚动一页

回车键 滚动一行

[pagedown]: 向下翻动一页

[pageup]: 向上翻动一页



## 4.4.3 less命令

案例：

- 使用less显示文件





## 4.4.4 head与tail

- 在读取文件的快速命令中，值得一提的还head和tail。顾名思义，head是用来显示文件开始部分的信息；tail是用来显示文件结束部分的信息。
- 对于head的而言，它可以有一个常用选项-n用于显示行数，默认为 10，即显示 10 行的内容；另一个常用选项-c用于显示的字节数；加上选项-v则可以在显示内容的同时也显示文件名。



## 4.4.4 head与tail

案例：

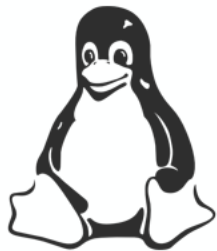
- 使用head显示文件
- 使用tail显示文件



# 本课练习

见教学网页





# Linux系统管理

南通师范高等专科学校 朱亚林

