

## 12. 文件属性

stat	显示文件与目录的属性。
wc	计算文本文件的字节数、单词数、行数。
du	显示文件与目录所占磁盘空间。
file	辨识（猜测）文件的类型。
touch	改变文件与目录的时间戳。
chown	改变文件与目录的拥有者。
chgrp	改变文件与目录的从属组。
chmod	改变文件与目录的访问模式。
chattr	改变文件与目录的扩展属性。
lsattr	列出文件与目录的扩展属性。

显示Linux文件的内容只是事情的一半。我们还需要能够查阅文件内容之外的信息，像拥有者、大小、访问模式等。虽然 `ls -l` 命令（见第 41 页“8. 文件基本操作”）能够显示文件的部分属性，但是 Linux 还提供了其他工具来显示更详尽的额外信息。

```
stat [options] files coreutils  
/usr/bin stdin stdout -file --opt --help --version
```

`stat` 命令能显示文件（默认行为）或文件系统（加上 `-f` 选项）的重要属性。文件信息的形式如下：

```
$ stat myfile  
File: `myfile'  
Size: 3280 Blocks: 8 IO Block: 4096 regular file  
Device: 302h/770d Inode: 196612 Links: 1  
Access: (0644/-rw-r--r--) Uid: (500/me) Gid: (500/me)  
Access: 2004-12-27 17:48:17.755769360 +0800  
Modify: 2004-12-27 17:48:17.756769208 +0800  
Change: 2004-12-27 17:48:17.756769208 +0800
```

在此例中，你可看到文件的名称、字节数（3280）、所占用的块数（8）、I/O 单位（4096）、文件类型（regular file）、该文件所在的设备类型与编号（302h/

770d)、Inode 编号 (196612)、硬链接的数量 (1)、访问模式 (0644/-rw-r--r--)、拥有者的 UID 与名称 (500/me)、所属组的 GID 与名称 (500/me)、最后一次访问时间、内容修改时间、状态改变时间 (译注 7)。

文件系统的信息类似下面这样：

```
File: "myfile"
  ID: 0          Namelen: 255      Type: ext2/ext3
Blocks: Total: 1032108 Free: 514738
        Available: 462310      Size: 4096
Inodes: Total: 524288 Free: 487317
```

你可看到文件名 (*myfile*)、文件系统的标识符 (0)、该文件系统可接受的文件名长度 (255)、文件系统的类型 (*ext2/ext3*)、块的总数 (1032108)、剩余数 (514738)、有效剩余数 (462310) 与单位大小 (4096)、Inodes 的总数 (524288) 与剩余数 (487317)。

-t 选项可使 `stat` 将所有数据都显示在同一行, 并且省略掉标题, 如此可使得 shell scripts 或其他程序方便分析。

```
$ stat -t /etc/passwd
/etc/passwd 1955 8 81a4 0 0 303 16571 1 0 0 ... (略)
$ stat -f -t /etc/group
/etc/group 0 255 * 515884 497661 471455 4096 ... (略)
```

## 常用选项

- l 跟随符号链接, 并汇报链接对象。
- f 汇报文件所在的文件系统, 而非文件本身。
- t 精简模式 (terse mode): 将所有信息都显示在同一行。

---

```
wc [options] [files] coreutils
/usr/bin stdin stdout -file --opt --help --version
```

`wc` (word count, 字数计算) 程序可算出文本文件所含的字节数、单词数、行数。

---

译注 7: 时间格式随当前的 `LANG` 环境变量而定, 精确度随硬件时钟的精确度和文件系统的类型而定。

```
$ wc myfile
 24 62 428 myfile
```

此文件共有 24 行、62 个以空格隔开的单词以及 428 个字节。

### 常用选项

- l 只显示行数。
- w 只显示单词数。
- c 只显示字节（字符）数。
- L 找出每个文件中最长的行并显示该行的长度（以 byte 为单位）。

### du [options] [files | directories]

coreutils

/usr/bin

stdin stdout -file --opt --help --version

du (disk usage) 命令可计算文件或目录所占的磁盘空间。没有指定任何选项时，它会测量当前工作目录与其所有子目录，分别显示各目录所占的块数，最后才显示工作目录所占的总块数。

```
$ du
8  ./Notes
36 ./Mail
340 ./Files/mine
40 ./Files/bob
416 ./Files
216 ./PC
2404 .
```

此外，wc 也可以测量个别文件所占的磁盘空间：

```
$ du myfile myfile2
4 ./myfile
16 ./myfile2
```

### 常用选项

- b -k -m 以字节（-b）、千字节（-k）或兆字节（-m）为计算单位。
- B N 以 N（默认值 1024）为块大小的计算单位。

- h -H 输出适合用户阅读的单位。举例来说，若有两个目录的大小分别为 1 gigabyte 与 25 kilobytes，则 `du -h` 会输出 1G 与 25K。  
-h 选项使用的单位基数是 1024，而 -H 选项使用的单位基数是 1000。
- c 于最后输出合计占用容量。这是测量目录时的默认行为，但是 -c 选项让你在测量文件时也有相同效果。
- L 跟随符号链接，并测量链接目标的大小。
- s 只显示工作目录所占总空间。

**file [options] files** file

---

`/usr/bin` `stdin stdout -file --opt --help --version`

file 命令可依据文件内容来猜测文件的类型：

```
$ file /etc/hosts /usr/bin/who letter.doc
/etc/hosts: ASCII text
/usr/bin/who: ELF 32-bit LSB executable, Intel 80386 ...
letter.doc: Microsoft Office Document
```

不同于其他某些操作系统，Linux 不记录文件类型。虽然很多文件都有扩展名，但即便是应用程序本身，也不见得依据扩展名来辨识文件类型。基本上，Linux 文件的扩展名只是给用户的提示，而非内容类型的决定性因素。

## 常用选项

- b 省略文件名（最左栏）。
- i 以 MIME 类型名称（例如“text/plain”、“audio/mpeg”）来代替正常的输出信息。
- f *file\_withfnlist* 读取 *file\_withfnlist* 文件，将该文件中的每一行视为一个文件名，然后汇报这些文件名所指文件的类型。
- L 跟随符号链接，汇报链接目标的文件类型，而不是告诉你那是符号链接。

-z 如果受测文件为压缩文件（参阅91页“15. 文件压缩”），则检验解压缩后的内容，然后汇报其内容类型，而不是告诉你那是压缩文件。

## touch [options] files

coreutils

/bin

stdin stdout -file --opt --help --version

touch命令可改变文件的两项时间戳属性：最后一次修改时间（上次修改文件内容的时间）以及最后一次访问时间（上次是何时被读取）。

```
$ touch myfile
```

你也可以指定任意的时间值：

```
$ touch -d "November 18 1975" myfile
```

如果目标文件不存在，则touch会自动创建一个新文件。这种方法常用来创建新的空文件。

## 常用选项

-a 只改变访问时间。

-m 只改变修改时间。

-c 不自动产生新文件。

-d *timestamp* 设定文件的时间戳。touch接受多种*timestamp*格式，从“12/28/2001 3pm”到“28-May”（当年指定日期的午夜时刻）到“next tuesday 13:59”到“0”（今日午夜）。使用stat可看到touch的修改结果。完整的*timestamp*格式说明，请参阅info touch。

-t *timestamp* 效果类似于-d选项，但是*timestamp*的格式限定为[[CC]YY]MMDDhhmm[.ss]。其中的CC是两位数的世纪（译注8），YY是两位数的年份，MM是两位数的月份，DD是两位数的日期，hh是两位数的小时，mm是两位数的分钟，ss是两位数的秒钟。举例来说，-t 20041207150047代表2004年12月7日15:00:47。

---

译注8：正确说法是世纪数减1。例如，21世纪是写成20。

---

*/bin* stdin stdout -file --opt --help --version

chown ( change owner ) 命令可设定文件或目录的拥有权。

```
$ chown smith myfile myfile2 mydir
```

*user\_spec* 参数有下列几种可能性：

- 一个用户名称 ( 或 UID 数值 )。只设定拥有者。
- 一个用户名称 ( 或 UID 数值 )，其后跟着一个冒号 ( : ) 与一个组名称 ( 或 GID 数值 )。设定拥有者与所属组。
- 一个用户名称 ( 或 UID 数值 )，其后跟着一个冒号 ( : )。设定拥有者，以该用户的登录组为本文件的所属组。
- 一个冒号后面接着一个组名称 ( 或 GID 数值 )。只设定所属组。
- `--reference=file`。以 *file* 的拥有者与所属组作为本文件的拥有者与所属组。

### 常用选项

`--dereference` 跟随符号链接，并作用在链接目标。

`-R` 递归作用在整个子目录的每一个文件与子目录上。

---

*/bin* stdin stdout -file --opt --help --version

chgrp ( change group ) 命令可设定文件与目录的组拥有权。

```
$ chgrp smith myfile myfile2 mydir
```

*group\_spec* 参数有下列几种形式：

- 一个组名称或 GID 数值。
- `--reference=file`，以另一个文件的所属组作为本文件的所属组。

关于组的概念，请参阅第 134 页“26. 组管理”。

## 常用选项

- `--dereference` 跟随符号链接，作用在链接对象，而非链接文件本身。
- `-R` 递归改变整个子目录的组拥有权。

**chmod [options] permissions files**

coreutils

/bin

stdin stdout -file --opt --help --version

`chmod` (change mode) 命令可设定文件与目录的访问模式。并非每个文件都可被任何人访问（你知道，这不是 Windows 95），而 `chmod` 就是用来确保这点的工具。访问权限分为读、写、执行三种，用户分成拥有者、同组、其他人三类。三种用户各有三种权限，构成 9-bit 访问模式。

`permissions` 参数有三种形式：

- `--reference=file`，引用另一个文件的访问模式。
- 四位数的八进制数字，以位形式指出文件的绝对权限 (absolute permissions)：最左侧的数字表示特殊属性，往右的数字依次表示拥有者、所属组、其他人的权限。图 3 以 0640 为例，分别指出各数字表示的意义。

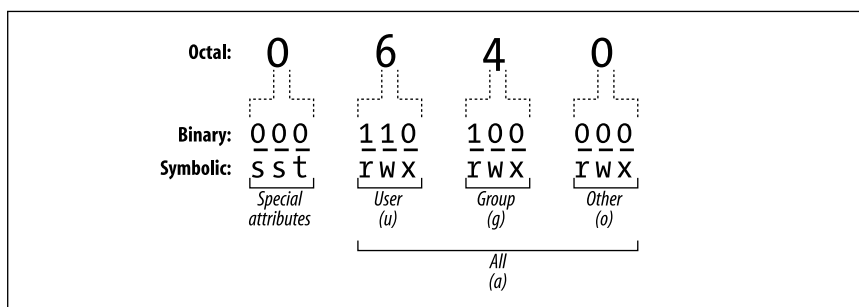


图 3：文件权限位

- 一个或多个以逗号相隔的字符串，这些字符串描述绝对权限或相对权限（相对于文件原有的权限）。

对于第三种形式，每个字符串可包含三个部分，分别是作用域 (scope)、命令 (command)、权限 (permission)，分述如下：

作用域（可省略）

u ( user ) 表示拥有者 , g ( group ) 表示所属组 , o ( other ) 表示非拥有者  
也非组成员的任何其他人 , a ( all ) 表示全体用户（默认）。

命令

+ 表示增加权限 , - 表示移除权限 , = 表示设定绝对权限（忽略原本的权限）。

权限

r 表示可读（文件）或可显示内容（目录）；w 表示可修改文件内容或是在目录下新增文件；x 代表可执行（对于文件）或是可进入（对于目录）；X 代表有条件执行（稍后解释）；u 继承“拥有者”的权限；g 继承所属组的权限；o 继承“其他人”的权限；s 用于 setuid 或 setgid；t 代表粘贴位（sticky bit）。

举例来说，ug+rw 将使得拥有者与所属组成员获得可读可写的权限，而 a-x（或只有 -x）将移除所有人的可执行权限。u=r 将撤消原本的任何权限，然后使拥有者具有可读权限。你可以用逗号将权限描述字符串结合在一起，例如：ug+rw,a-x。

setuid 和 setgid 仅适用于可执行文件（程序或脚本程序）。假设我们有一个可执行文件 *F*，其拥有者是 smith，所属组为 friends。假设 *F* 文件的 setuid（set user ID）有效，则任何人在运行 *F* 程序的这段期间，将会暂时“成为”smith，并继承该身份的所有权限。

同理，若 *F* 文件的 setgid（set group ID）权限位被设定，则任何能够运行 *F* 程序代码的人在该程序的运行期间，将暂时成为 *F* 所属组的成员。正如你的想象，setuid 和 setgid 对系统安全的影响甚巨，所以，除非你真的知道自己在做什么，否则不应该贸然设定 setuid 或 setgid 位。一次失误运行 chmod +s，就有可能给整个系统带来莫大的风险。

条件性执行权限（X）的作用类似于 x，唯一差别在于它只对可执行文件与目录有效，否则没有作用。

常用选项

-R 递归改变整个子目录的访问模式。

如果你曾经使用过其他 Unix 系统，或许会惊讶 Linux 文件竟然有访问模式之外的扩展属性。对于存放在 ext2 或 ext3 文件系统（Fedora 默认使用）中的文件，你可以用 `chattr`（change attribute）来改变它们的扩展属性，用 `lsattr` 来显示它们。

如同 `chmod`，`chattr` 提供三种属性操作，分别是增加（+）、撤销（-）以及设定（=）。

属性	意义
a	仅供附加（Append-only）：容许增添内容到文件末端，但是不容许修改其原有内容。仅有 root 有权限设定此属性
A	不记录访问时间。读取文件时，不自动更新其访问时间（atime）
c	自动压缩：写入数据时自动压缩；读取数据时自动解压缩
d	不转储（dump）：要求 <code>dump</code> 程序在制作备份时，忽略此文件（参阅 106 页“18. 备份”）
i	不可变（Immutable）：不能删除文件，也不能改变文件内容。仅有 root 有权限设定此属性
j	日志记录数据（Journaled data），仅 ext3 文件系统有此属性
s	安全删除（secure deletion）：文件被删除时，将磁盘上的文件数据都归零
S	同步更新（synchronous update）：发生变动时，立刻将数据写回磁盘，其效果就像是保存之后手动运行一次 <code>sync</code> （见第 101 页“17. 磁盘与文件系统”）
u	禁删（Undeletable）：文件不可被删除

## 常用选项

`-R` 递归处理整个子目录。

使用 `chattr` 设定扩展属性之后，可用 `lsattr` (list attributes) 来显示扩展属性的设定值。`lsattr` 的输出列表使用与 `chattr` 相同的属性字母。例如，下列文件不可变 (immutable)，而且不容许删除 (undeletable)：

```
$ lsattr myfile
-u--i--- myfile
```

### 常用选项

- R 递归列出整个子目录。
- a 列出包含隐藏文件 (文件名第一个字符为“.”) 在内的所有文件。
- d 遇到目录时不显示其内容，而显示目录本身的扩展属性。

没指定文件时，`lsattr` 列出当前工作目录下所有文件的扩展属性。

---