

# Linux 学习总结（二）Vim 使用及账号用户管理

作者: [wlgzs-sjl](#)

原文链接: <https://ld246.com/article/1604560638061>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p></p>

<h3 id="什么是Vim编辑器">什么是 Vim 编辑器</h3>

<p>Vim 是从 vi 发展出来的一个文本编辑器。代码补完、编译及错误跳转等方便编程的功能特别丰，在程序员中被广泛使用。</p>

<p>简单的来说，vi 是老式的字处理器，不过功能已经很齐全了，但是还是有可以进步的地方。</p>

<p>vim 则可以说是程序开发者的一项很好用的工具。</p>

<p>所有的 Unix Like 系统都会内建 vi 文书编辑器，其他的文书编辑器则不一定会存在。</p>

<p>连 vim 的官方网站 (<http://www.vim.org>) 自己也说 vim 是一个程序开发工具而不是文字处理软。</p>

<h3 id="三种使用模式">三种使用模式</h3>

<p>基本上 vi/vim 共分为三种模式，分别是<strong>命令模式 (Command mode) </strong>，<strong>输入模式 (Insert mode) </strong>和<strong>底线命令模式 (Last line mode) </stro g>。这三种模式的作用分别是：</p>

<h4 id="命令模式-">命令模式：</h4>

<p>用户刚刚启动 vi/vim，便进入了命令模式。</p>

<p>此状态下敲击键盘动作会被 Vim 识别为命令，而非输入字符。比如我们此时按下 i，并不会输入个字符，i 被当作了一个命令。</p>

<p>以下是常用的几个命令：</p>

<ul>

<li><strong>i</strong> 切换到输入模式，以输入字符。</li>

<li><strong>x</strong> 删除当前光标所在处的字符。</li>

<li><strong>:</strong> 切换到底线命令模式，以在最底一行输入命令。</li>

</ul>

<p>若想要编辑文本：启动 Vim，进入了命令模式，按下 i，切换到输入模式。</p>

<p>命令模式只有一些最基本的命令，因此仍要依靠底线命令模式输入更多命令。</p>

<h4 id="输入模式-">输入模式：</h4>

<p>在命令模式下按下 i 就进入了输入模式。</p>

<p>在输入模式中，可以使用以下按键：</p>

<ul>

<li><strong>字符按键以及 Shift 组合</strong>，输入字符</li>

<li><strong>ENTER</strong>，回车键，换行</li>

<li><strong>BACK SPACE</strong>，退格键，删除光标前一个字符</li>

<li><strong>DEL</strong>，删除键，删除光标后一个字符</li>

<li><strong>方向键</strong>，在文本中移动光标</li>

<li><strong>HOME</strong>/<strong>END</strong>，移动光标到行首/行尾</li>

<li><strong>Page Up</strong>/<strong>Page Down</strong>，上/下翻页</li>

<li><strong>Insert</strong>，切换光标为输入/替换模式，光标将变成竖线/下划线</li>

<li><strong>ESC</strong>，退出输入模式，切换到命令模式</li>

</ul>

<h4 id="底线命令模式">底线命令模式</h4>

<p>在命令模式下按下: (英文冒号) 就进入了底线命令模式。</p>

<p>底线命令模式可以输入单个或多个字符的命令，可用的命令非常多。</p>

<p>在底线命令模式中，基本的命令有 (已经省略了冒号)：</p>

<ul>

<li>q 退出程序</li>

<li>w 保存文件</li>

</ul>

<p>按 ESC 键可随时退出底线命令模式。</p>

<h4 id="Vim-按键说明">Vim 按键说明</h4>

<p>除了上面简易范例的 i, Esc, :wq 之外，其实 vim 还有非常多的按键可以使用。</p>

<p><strong>第一部分：一般模式可用的光标移动、复制粘贴、搜索替换等</strong></p>

<table>

```

<thead>
<tr>
<th>移动光标的方法</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>h 或 向左箭头键(←)</td>
<td>光标向左移动一个字符</td>
</tr>
<tr>
<td>j 或 向下箭头键(↓)</td>
<td>光标向下移动一个字符</td>
</tr>
<tr>
<td>k 或 向上箭头键(↑)</td>
<td>光标向上移动一个字符</td>
</tr>
<tr>
<td>l 或 向右箭头键(→)</td>
<td>光标向右移动一个字符</td>
</tr>
<tr>
<td>[Ctrl] + [f]</td>
<td>屏幕『向下』移动一页，相当于 [Page Down]按键 (常用)</td>
</tr>
<tr>
<td>[Ctrl] + [b]</td>
<td>屏幕『向上』移动一页，相当于 [Page Up] 按键 (常用)</td>
</tr>
<tr>
<td>[Ctrl] + [d]</td>
<td>屏幕『向下』移动半页</td>
</tr>
<tr>
<td>[Ctrl] + [u]</td>
<td>屏幕『向上』移动半页</td>
</tr>
<tr>
<td>+</td>
<td>光标移动到非空格符的下一行</td>
</tr>
<tr>
<td>-</td>
<td>光标移动到非空格符的上一行</td>
</tr>
<tr>
<td>n<space></td>
<td>那个 n 表示『数字』，例如 20 。按下数字后再按空格键，光标会向右移动这一行的 n 个字符</td>
</tr>
<tr>
<td>0 或功能键[Home]</td>

```

<td>这是数字『0』：移动到这一行的最前面字符处 (常用)</td>
</tr>
<td>\$ 或功能键[End]</td>
<td>移动到这一行的最后面字符处(常用)</td>
</tr>
<td>H</td>
<td>光标移动到这个屏幕的最上方那一行的第一个字符</td>
</tr>
<td>M</td>
<td>光标移动到这个屏幕的中央那一行的第一个字符</td>
</tr>
<td>L</td>
<td>光标移动到这个屏幕的最下方那一行的第一个字符</td>
</tr>
<td>G</td>
<td>移动到这个档案的最后一行(常用)</td>
</tr>
<td>nG</td>
<td>n 为数字。移动到这个档案的第 n 行。例如 20G 则会移动到这个档案的第 20 行(可配合 :set nu
</td>
</tr>
<td>gg</td>
<td>移动到这个档案的第一行，相当于 1G 啊! (常用)</td>
</tr>
<td>n<Enter></td>
<td>n 为数字。光标向下移动 n 行(常用)</td>
</tr>

<th>删除、复制与粘贴</th>
<th></th>
</tr>

<td>x, X</td>
<td>在一行字当中，x 为向后删除一个字符 (相当于 [del] 按键)，X 为向前删除一个字符(相当于 [ba
kspace] 亦即是退格键) (常用)</td>
</tr>
<td>nx</td>
<td>n 为数字，连续向后删除 n 个字符。举例来说，我要连续删除 10 个字符，『10x』。</td>
</tr>

<tr>
<td>dd</td>
<td>删除光标所在的那一整行(常用)</td>
</tr>
<tr>
<td>ndd</td>
<td>n 为数字。删除光标所在的向下 n 行，例如 20dd 则是删除 20 行 (常用)</td>
</tr>
<tr>
<td>d1G</td>
<td>删除光标所在到第一行的所有数据</td>
</tr>
<tr>
<td>dG</td>
<td>删除光标所在到最后一行的所有数据</td>
</tr>
<tr>
<td>d\$</td>
<td>删除光标所在处，到该行的最后一个字符</td>
</tr>
<tr>
<td>d0</td>
<td>那个是数字的 0，删除光标所在处，到该行的最前面一个字符</td>
</tr>
<tr>
<td>yy</td>
<td>复制光标所在的那一行(常用)</td>
</tr>
<tr>
<td>nyy</td>
<td>n 为数字。复制光标所在的向下 n 行，例如 20yy 则是复制 20 行(常用)</td>
</tr>
<tr>
<td>y1G</td>
<td>复制光标所在行到第一行的所有数据</td>
</tr>
<tr>
<td>yG</td>
<td>复制光标所在行到最后一行的所有数据</td>
</tr>
<tr>
<td>y0</td>
<td>复制光标所在的那个字符到该行行首的所有数据</td>
</tr>
<tr>
<td>y\$</td>
<td>复制光标所在的那个字符到该行行尾的所有数据</td>
</tr>
<tr>
<td>p, P</td>
<td>p 为将已复制的数据在光标下一行贴上，P 则为贴在光标上一行！举例来说，我目前光标在第 20 行，且已经复制了 10 行数据。则按下 p 后，那 10 行数据会贴在原本的 20 行之后，亦即由 21 行始贴。但如果是按下 P 呢？那么原本的第 20 行会被推到变成 30 行。(常用)</td>
</tr>

<tr>
<td>J</td>
<td>将光标所在行与下一行的数据结合成同一行</td>
</tr>
<tr>
<td>c</td>
<td>重复删除多个数据，例如向下删除 10 行，[ 10cj ]</td>
</tr>
<tr>
<td>u</td>
<td>复原前一个动作。(常用)</td>
</tr>
<tr>
<td>[Ctrl]+r</td>
<td>重做上一个动作。(常用)</td>
</tr>
</tbody>
</table>
<table>
<thead>
<tr>
<th>搜索替换</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>/word</td>
<td>向光标之下寻找一个名称为 word 的字符串。例如要在档案内搜寻 vbird 这个字符串，就输入 /bird 即可! (常用)</td>
</tr>
<tr>
<td>?word</td>
<td>向光标之上寻找一个字符串名称为 word 的字符串。</td>
</tr>
<tr>
<td>n</td>
<td>这个 n 是英文按键。代表重复前一个搜寻的动作。举例来说，如果刚刚我们执行 /vbird 去向搜寻 vbird 这个字符串，则按下 n 后，会向下继续搜寻下一个名称为 vbird 的字符串。如果是执行 ?vbird 的话，那么按下 n 则会向上继续搜寻名称为 vbird 的字符串! </td>
</tr>
<tr>
<td>N</td>
<td>这个 N 是英文按键。与 n 刚好相反，为『反向』进行前一个搜寻动作。例如 /vbird 后，按下 N 则表示『向上』搜寻 vbird 。</td>
</tr>
</tbody>
</table>
<p><strong>第二部分：一般模式切换到编辑模式的可用的按钮说明</strong></p>
<table>
<thead>
<tr>
<th>进入输入或取代的编辑模式</th>
<th></th>

i, I	进入输入模式(Insert mode): i 为『从目前光标所在处输入』, I 为『在目前所在行的第一个空格符处开始输入』。(常用)
a, A	进入输入模式(Insert mode): a 为『从目前光标所在的下一个字符处开始输入』, A 为『从光所在行的最后一个字符处开始输入』。(常用)
o, O	进入输入模式(Insert mode): 这是英文字母 o 的大小写。o 为『在目前光标所在的下一行处输新的一行』; O 为在目前光标所在处的上一行输入新的一行! (常用)
r, R	进入取代模式(Replace mode): r 只会取代光标所在的那一个字符一次; R 会一直取代光标所的文字, 直到按下 ESC 为止; (常用)
[Esc]	退出编辑模式, 回到一般模式中(常用)

**第三部分: 一般模式切换到指令行模式的可用的按钮说明**

指令行的储存、离开等指令	
:w	将编辑的数据写入硬盘档案中(常用)
:w!	若文件属性为『只读』时, 强制写入该档案。不过, 到底能不能写入, 还是跟你对该档案的档权限有关啊!
:q	离开 vi (常用)
:q!	若曾修改过档案, 又不想储存, 使用 ! 为强制离开不储存档案。

<p>&lt;td&gt;注意一下啊，那个惊叹号 (!) 在 vi 当中，常常具有『强制』的意思 ~ &lt;/td&gt;</p>
<p>&lt;td&gt;:wq&lt;/td&gt;</p> <p>&lt;td&gt;储存后离开，若为 :wq! 则为强制储存后离开 (常用)&lt;/td&gt;</p>
<p>&lt;td&gt;ZZ&lt;/td&gt;</p> <p>&lt;td&gt;这是大写的 Z 喔！若档案没有更动，则不储存离开，若档案已经被更动过，则储存后离开！ &lt;/td&gt;</p>
<p>&lt;td&gt;:w [filename]&lt;/td&gt;</p> <p>&lt;td&gt;将编辑的数据储存成另一个档案（类似另存新档）&lt;/td&gt;</p>
<p>&lt;td&gt;:r [filename]&lt;/td&gt;</p> <p>&lt;td&gt;在编辑的数据中，读入另一个档案的数据。亦即将『filename』这个档案内容加到游标所在行面&lt;/td&gt;</p>
<p>&lt;td&gt;:n1,n2 w [filename]&lt;/td&gt;</p> <p>&lt;td&gt;将 n1 到 n2 的内容储存成 filename 这个档案。&lt;/td&gt;</p>
<p>&lt;td&gt;:! command&lt;/td&gt;</p> <p>&lt;td&gt;暂时离开 vi 到指令行模式下执行 command 的显示结果！例如『:! ls /home』即可在 vi 当中看 /home 底下以 ls 输出的档案信息！&lt;/td&gt;</p>
<p>&lt;td&gt;:set nu&lt;/td&gt;</p> <p>&lt;td&gt;显示行号，设定之后，会在每一行的前缀显示该行的行号&lt;/td&gt;</p>
<p>&lt;td&gt;:set nonu&lt;/td&gt;</p> <p>&lt;td&gt;与 set nu 相反，为取消行号！&lt;/td&gt;</p>

### 账号管理

#### 简介

Linux 系统是一个多用户多任务的分时操作系统，任何一个要使用系统资源的用户，都必须首先系统管理员申请一个账号，然后以这个账号的身份进入系统。

用户的账号一方面可以帮助系统管理员对使用系统的用户进行跟踪，并控制他们对系统资源的访问；另一方面也可以帮助用户组织文件，并为用户提供安全性保护。

每个用户账号都拥有一个唯一的用户名和各自的口令。

用户在登录时键入正确的用户名和口令后，就能够进入系统和自己的主目录。

实现用户账号的管理，要完成的工作主要有如下几个方面：

- 用户账号的添加、删除与修改。
- 用户口令的管理。



</li>用户组的管理。 </li>

</ul>

#### 用户账号的管理</h4>

<p>用户账号的管理工作主要涉及到用户账号的添加、修改和删除。 </p>

<p>添加用户账号就是在系统中创建一个新账号，然后为新账号分配用户号、用户组、主目录和登录hell等资源</p>

<blockquote>

<p>添加账号 useradd</p>

</blockquote>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">useradd 选项 用户名</span></span></code></pre>
```

<p>参数说明： </p>

<ul>

<li>选项：

<ul>

<li>-c comment 指定一段注释性描述。 </li>

<li>-d 目录 指定用户主目录，如果此目录不存在，则同时使用-m 选项，可以创建主目录。 </li>

<li>-g 用户组 指定用户所属的用户组。 </li>

<li>-G 用户组，用户组 指定用户所属的附加组。 </li>

<li>-m 使用者目录如不存在则自动建立。 </li>

<li>-s Shell 文件 指定用户的登录 Shell。 </li>

<li>-u 用户号 指定用户的用户号，如果同时有-o 选项，则可以重复使用其他用户的标识号。 </li>

</ul>

</li>

<li>用户名 :</li>

<li>指定新账号的登录名。 </li>

</ul>

<p>测试： </p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"># 此命令创建了一个用户test，其中-m选项用来为登录名test产生一个主目录 /home/test</span></span><span class="highlight-line"><span class="highlight-cl">[root@iz2zeenexi kssoufermazz home]# useradd -m test</span></span></code></pre>
```

<p>增加用户账号就是在/etc/passwd 文件中为新用户增加一条记录，同时更新其他系统文件如/etc/hadow, /etc/group 等。 </p>

<blockquote>

<p>Linux 下如何切换用户</p>

</blockquote>

<p>1.切换用户的命令为：su username【username 是用户名】 </p>

<p>2.从普通用户切换到 root 用户，还可以使用命令：sudo su</p>

<p>3.在终端输入 exit 或 logout 或使用快捷方式 ctrl+d，可以退回到原来用户，其实 ctrl+d 也是行的 exit 命令</p>

<p>4.在切换用户时，如果想在切换用户之后使用新用户的工作环境，可以在 su 和 username 之间加，例如：【su - root】 </p>

<p>\$ 表示普通用户</p>

<p>#表示超级用户，也就是 root 用户</p>

<blockquote>

<p>删除帐号</p>

</blockquote>

<p>如果一个用户的账号不再使用，可以从系统中删除。 </p>

<p>删除用户账号就是要将/etc/passwd 等系统文件中的该用户记录删除，必要时还删除用户的主目。 </p>

<p>删除一个已有的用户账号使用 userdel 命令，其格式如下： </p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">userdel 选项 用户名
</span> </span> </code> </pre>
<p>常用的选项是 <strong>-r</strong>, 它的作用是把用户的主目录一起删除。</p>
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">[root@iz2zeeneximkssoufermazz home]# userdel -r test
</span> </span> </code> </pre>
<p>此命令删除用户 kuangshen 在系统文件中 (主要是/etc/passwd, /etc/shadow, /etc/group 等的记录, 同时删除用户的主目录。</p>
<blockquote>
<p>修改帐号</p>
</blockquote>
<p>修改用户账号就是根据实际情况更改用户的有关属性, 如用户号、主目录、用户组、登录 Shell 。</p>
<p>修改已有用户的信息使用 usermod 命令, 其格式如下: </p>
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">usermod 选项 用户名
</span> </span> </code> </pre>
<p>常用的选项包括-c, -d, -m, -g, -G, -s, -u 以及-o 等, 这些选项的意义与 useradd 命令中的选项样, 可以为用户指定新的资源值。</p>
<p>例如: </p>
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">[root@iz2zeeneximkssoufermazz home]# usermod -s /bin/ksh -d /home/z -g developer t
st
</span> </span> </code> </pre>
<p>此命令将用户 test 的登录 Shell 修改为 ksh, 主目录改为/home/z, 用户组改为 developer。</p>
<blockquote>
<p>用户口令的管理</p>
</blockquote>
<p>用户管理的一项重要内容是用户口令的管理。用户账号刚创建时没有口令, 但是被系统锁定, 无使用, 必须为其指定口令后才可以使使用, 即使是指定空口令。</p>
<p>指定和修改用户口令的 Shell 命令是 passwd。超级用户可以为和其他用户指定口令, 普通用户只能用它修改自己的口令。</p>
<p>命令的格式为: </p>
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">passwd 选项 用户名
</span> </span> </code> </pre>
<p>可使用的选项: </p>
<ul>
<li>-l 锁定口令, 即禁用账号。</li>
<li>-u 口令解锁。</li>
<li>-d 使账号无口令。</li>
<li>-f 强迫用户下次登录时修改口令。</li>
</ul>
<p>如果默认用户名, 则修改当前用户的口令。</p>
<p>例如, 假设当前用户是 test, 则下面的命令修改该用户自己的口令: </p>
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">$ passwd
</span> </span> <span class="highlight-line"> <span class="highlight-cl">Old password:****
*
</span> </span> <span class="highlight-line"> <span class="highlight-cl">New password:***
***
</span> </span> <span class="highlight-line"> <span class="highlight-cl">Re-enter new pas
```

word:\*\*\*\*\*

```
</span></span></code></pre>
```

<p>如果是超级用户，可以用下列形式指定任何用户的口令：</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"># passwd test
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">New password:***
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">Re-enter new password:*****
```

```
</span></span></code></pre>
```

<p>普通用户修改自己的口令时，passwd 命令会先询问原口令，验证后再要求用户输入两遍新口令。如果两次输入的口令一致，则将这个口令指定给用户；而超级用户为用户指定口令时，就不需要知道口令。</p>

<p>为了系统安全起见，用户应该选择比较复杂的口令，例如最好使用 8 位长的口令，口令中包含有写、小写字母和数字，并且应该与姓名、生日等不相同。</p>

<p>为用户指定空口令时，执行下列形式的命令：</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"># passwd -d test
```

```
</span></span></code></pre>
```

<p>此命令将用户 test 的口令删除，这样用户 test 下一次登录时，系统就不再允许该用户登录了。</p>

<p>passwd 命令还可以用 -l(lock) 选项锁定某一用户，使其不能登录，例如：</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"># passwd -l test
```

```
</span></span></code></pre>
```

### 用户组管理</h3>

<p>每个用户都有一个用户组，系统可以对一个用户组中的所有用户进行集中管理。不同 Linux 系统用户组的规定有所不同，如 Linux 下的用户属于与它同名的用户组，这个用户组在创建用户时同时创建。</p>

<p>用户组的管理涉及用户组的添加、删除和修改。组的增加、删除和修改实际上就是对/etc/group 文件的更新。</p>

<blockquote>

<p>增加一个新的用户组使用 groupadd 命令</p>

</blockquote>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">groupadd 选项 用户组
```

```
</span></span></code></pre>
```

<p>可以使用的选项有：</p>

<ul>

<li>-g GID 指定新用户组的组标识号 (GID) 。</li>

<li>-o 一般与-g 选项同时使用，表示新用户组的 GID 可以与系统已有用户组的 GID 相同。</li>

</ul>

<p>实例 1：</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"># groupadd group1
```

```
</span></span></code></pre>
```

<p>此命令向系统中增加了一个新组 group1，新组的组标识号是在当前已有的最大组标识号的基础上加 1。</p>

<p>实例 2：</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"># groupadd -g 101 group2
```

```
</span></span></code></pre>
```

<p>此命令向系统中增加了一个新组 group2，同时指定新组的组标识号是 101。</p>

<blockquote>

<p>如果要删除一个已有的用户组，使用 groupdel 命令</p>

</blockquote>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">groupdel 用户组
```

```
</span> </span> </code> </pre>
```

<p>例如： </p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"># groupdel group1
```

```
</span> </span> </code> </pre>
```

<p>此命令从系统中删除组 group1。 </p>

<blockquote>

<p>修改用户组的属性使用 groupmod 命令</p>

</blockquote>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">groupmod 选项 用户组
```

```
</span> </span> </code> </pre>
```

<p>常用的选项有： </p>

<ul>

<li>-g GID 为用户组指定新的组标识号。 </li>

<li>-o 与-g 选项同时使用，用户组的新 GID 可以与系统已有用户组的 GID 相同。 </li>

<li>-n 新用户组 将用户组的名字改为新名字</li>

</ul>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"># 此命令将组group2的组标识号修改为102。
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl">groupmod -g 102
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"># 将组group2的
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl">groupmod -g 100
```

```
</span> </span> </code> </pre>
```

<blockquote>

<p>切换组</p>

</blockquote>

<p>如果一个用户同时属于多个用户组，那么用户可以在用户组之间切换，以便具有其他用户组的权。 </p>

<p>用户可以在登录后，使用命令 newgrp 切换到其他用户组，这个命令的参数就是目的用户组。例： </p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">$ newgrp root
```

```
</span> </span> </code> </pre>
```

<p>这条命令将当前用户切换到 root 用户组，前提条件是 root 用户组确实是该用户的主组或附加组</p>

<blockquote>

<p>/etc/passwd</p>

</blockquote>

<p>完成用户管理的工作有许多种方法，但是每一种方法实际上都是对有关的系统文件进行修改。 </p>

<p>与用户和用户组相关的信息都存放在一些系统文件中，这些文件包括/etc/passwd, /etc/shadow, /etc/group 等。 </p>

<p>下面分别介绍这些文件的内容。 </p>

<p><strong>/etc/passwd 文件是用户管理工作涉及的最重要的一个文件。 </strong> </p>

<p>Linux 系统中的每个用户都在/etc/passwd 文件中有一个对应的记录行，它记录了这个用户的一

基本属性。</p>

<p>这个文件对所有用户都是可读的。它的内容类似下面的例子：</p>

<p></p>

<p>从上面的例子我们可以看到，/etc/passwd 中一行记录对应着一个用户，每行记录又被冒号(:)分为 7 个字段，其格式和具体含义如下：</p>

<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">用户名:口令:用户标识号:组标识号:注释性描述:主目录:登录Shell</span></span></code></pre>

<ol>

<li>"用户名"是代表用户账号的字符串。</li>

</ol>

<p>通常长度不超过 8 个字符，并且由大小写字母和/或数字组成。登录名中不能有冒号(:)，因为冒在这里是分隔符。</p>

<p>为了兼容起见，登录名中最好不要包含点字符(.)，并且不使用连字符(-)和加号(+)打头。</p>

<ol start="2">

<li>"口令" 一些系统中，存放着加密后的用户口令字。</li>

</ol>

<p>虽然这个字段存放的只是用户口令的加密串，不是明文，但是由于/etc/passwd 文件对所有用户可读，所以这仍是一个安全隐患。因此，现在许多 Linux 系统（如 SVR4）都使用了 shadow 技术，真正的加密后的用户口令字存放到/etc/shadow 文件中，而在/etc/passwd 文件的口令字段中只存一个特殊的字符，例如 "x" 或者 "\*" 。</p>

<ol start="3">

<li>"用户标识号" 是一个整数，系统内部用它来标识用户。</li>

</ol>

<p>一般情况下它与用户名是一一对应的。如果几个用户名对应的用户标识号是一样的，系统内部将它们视为同一个用户，但是它们可以有不同的口令、不同的主目录以及不同的登录 Shell 等。</p>

<p>通常用户标识号的取值范围是 0~65 535。0 是超级用户 root 的标识号，1~99 由系统保留，为管理账号，普通用户的标识号从 100 开始。在 Linux 系统中，这个界限是 500。</p>

<ol start="4">

<li>"组标识号" 字段记录的是用户所属的用户组。</li>

</ol>

<p>它对应着/etc/group 文件中的一条记录。</p>

<p>5) "注释性描述" 字段记录着用户的一些个人情况。</p>

<p>例如用户的真实姓名、电话、地址等，这个字段并没有什么实际的用途。在不同的 Linux 系统中这个字段的格式并没有统一。在许多 Linux 系统中，这个字段存放的是一段任意的注释性描述文字，作 finger 命令的输出。</p>

<p>6) "主目录"，也就是用户的起始工作目录。</p>

<p>它是用户在登录到系统之后所处的目录。在大多数系统中，各用户的主目录都被组织在同一个特的目录下，而用户主目录的名称就是该用户的登录名。各用户对自己的主目录有读、写、执行（搜索权限，其他用户对此目录的访问权限则根据具体情况设置。</p>

<p>7) 用户登录后，要启动一个进程，负责将用户的操作传给内核，这个进程是用户登录到系统后运的命令解释器或某个特定的程序，即 Shell。</p>

<p>Shell 是用户与 Linux 系统之间的接口。Linux 的 Shell 有许多种，每种都有不同的特点。常用有 sh(Bourne Shell), csh(C Shell), ksh(Korn Shell), tcsh(TENEX/TOPS-20 type C Shell), bash(Bou ne Again Shell)等。</p>

<p>系统管理员可以根据系统情况和用户习惯为用户指定某个 Shell。如果不指定 Shell，那么系统使 sh 为默认的登录 Shell，即这个字段的值为/bin/sh。</p>

<p>用户的登录 Shell 也可以指定为某个特定的程序（此程序不是一个命令解释器）。</p>

<p>利用这一特点，我们可以限制用户只能运行指定的应用程序，在该应用程序运行结束后，用户就动退出了系统。有些 Linux 系统要求只有那些在系统中登记了的程序才能出现在这个字段中。</p>

<p>8) 系统中有一类用户称为伪用户（pseudo users）。</p>

<p>这些用户在/etc/passwd 文件中也占有一条记录，但是不能登录，因为它们的登录 Shell 为空。



们的存在主要是方便系统管理，满足相应的系统进程对文件属主的要求。

常见的伪用户如下所示：

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">伪用户含义
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">bin 拥有可执行的  
户命令文件
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">sys 拥有系统文件
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">adm 拥有帐户文件
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">uucp UUCP使用
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">lp lp或lpd子系统  
用
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">nobody NFS使用
```

```
</span></span></code></pre>
```

```
<blockquote>
```

```
<p>/etc/shadow</p>
```

```
</blockquote>
```

**1、除了上面列出的伪用户外，还有许多标准的伪用户，例如：audit, cron, mail, usen t 等，它们也都各自为相关的进程和文件所需要。**

由于/etc/passwd 文件是所有用户都可读的，如果用户的密码太简单或规律比较明显的话，一台通的计算机就能够很容易地将它破解，因此对安全性要求较高的 Linux 系统都把加密后的口令字分离来，单独存放在一个文件中，这个文件是/etc/shadow 文件。有超级用户才拥有该文件读权限，这就证了用户密码的安全性。

**2、/etc/shadow 中的记录行与/etc/passwd 中的一一对应，它由 pwconv 命令根据/tc/passwd 中的数据自动产生**

它的文件格式与/etc/passwd 类似，由若干个字段组成，字段之间用":"隔开。这些字段是：

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">登录名:加密口令:最后一次修改时间:最小时间间隔:最大时间间隔:警告时间:不活动时间:失效时间:  
志
```

```
</span></span></code></pre>
```

```
<ol>
```

```
<li>"登录名"是与/etc/passwd 文件中的登录名相一致的用户账号</li>
```

```
<li>"口令"字段存放的是加密后的用户口令字，长度为 13 个字符。如果为空，则对应用户没有口令  
登录时不需要口令；如果含有不属于集合 { ./0-9A-Za-z }中的字符，则对应的用户不能登录。</li>
```

```
<li>"最后一次修改时间"表示的是从某个时刻起，到用户最后一次修改口令时的天数。时间起点对不  
的系统可能不一样。例如在 SCO Linux 中，这个时间起点是 1970 年 1 月 1 日。</li>
```

```
<li>"最小时间间隔"指的是两次修改口令之间所需的最小天数。</li>
```

```
<li>"最大时间间隔"指的是口令保持有效的最大天数。</li>
```

```
<li>"警告时间"字段表示的是从系统开始警告用户到用户密码正式失效之间的天数。</li>
```

```
<li>"不活动时间"表示的是用户没有登录活动但账号仍能保持有效的最大天数。</li>
```

```
<li>"失效时间"字段给出的是一个绝对的天数，如果使用了这个字段，那么就给出相应账号的生存期  
期满后，该账号就不再是一个合法的账号，也就不能再用来登录了。</li>
```

```
</ol>
```

```
<blockquote>
```

```
<p>/etc/group</p>
```

```
</blockquote>
```

用户组的所有信息都存放在/etc/group 文件中。

将用户分组是 Linux 系统中对用户进行管理 & 控制访问权限的一种手段。

每个用户都属于某个用户组；一个组中可以有多个用户，一个用户也可以属于不同的组。

当一个用户同时是多个组中的成员时，在/etc/passwd 文件中记录的是用户所属的主组，也就是录时所属的默认组，而其他组称为附加组。

用户要访问属于附加组的文件时，必须首先使用 newgrp 命令使自己成为所要访问的组中的成员

用户组的所有信息都存放在/etc/group 文件中。此文件的格式也类似于/etc/passwd 文件，由号(:)隔开若干个字段，这些字段有：

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">组名:口令:组标识号:组内用户列表
</span></span></code></pre>
<ol>
<li>"组名"是用户组的名称，由字母或数字构成。与/etc/passwd 中的登录名一样，组名不应重复。</li>
<li>"口令"字段存放的是用户组加密后的口令字。一般 Linux 系统的用户组都没有口令，即这个字段
一般为空，或者是*。</li>
<li>"组标识号"与用户标识号类似，也是一个整数，被系统内部用来标识组。</li>
<li>"组内用户列表"是属于这个组的所有用户的列表/b]，不同用户之间用逗号(,)分隔。这个用户组可
是用户的主组，也可能是附加组。</li>
</ol>
<h3 id="磁盘管理">磁盘管理</h3>
<h4 id="概述">概述</h4>
<p>Linux 磁盘管理好坏直接关系到整个系统的性能问题。</p>
<p>Linux 磁盘管理常用命令为 df、du。</p>
<ul>
<li>df：列出文件系统的整体磁盘使用量</li>
<li>du：检查磁盘空间使用量</li>
</ul>
<blockquote>
<p>df</p>
</blockquote>
<p>df 命令参数功能：检查文件系统的磁盘空间占用情况。可以利用该命令来获取硬盘被占用了多
空间，目前还剩下多少空间等信息。</p>
<p>语法：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">df [-ahikHTm] [目录或文件名]
</span></span></code></pre>
<p>选项与参数：</p>
<ul>
<li>-a：列出所有的文件系统，包括系统特有的 /proc 等文件系统；</li>
<li>-k：以 KBytes 的容量显示各文件系统；</li>
<li>-m：以 MBytes 的容量显示各文件系统；</li>
<li>-h：以人们较易阅读的 GBytes, MBytes, KBytes 等格式自行显示；</li>
<li>-H：以 M=1000K 取代 M=1024K 的进位方式；</li>
<li>-T：显示文件系统类型, 连同该 partition 的 filesystem 名称 (例如 ext3) 也列出；</li>
<li>-i：不用硬盘容量，而以 inode 的数量来显示</li>
</ul>
<blockquote>
<p>du</p>
</blockquote>
<p>Linux du 命令也是查看使用空间的，但是与 df 命令不同的是 Linux du 命令是对文件和目录磁
使用的空间的查看，还是和 df 命令有一些区别的，这里介绍 Linux du 命令。</p>
<p>语法：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">du [-ahskm] 文件或目录名称
</span></span></code></pre>
<p>选项与参数：</p>
<ul>
<li>-a：列出所有的文件与目录容量，因为默认仅统计目录底下的文件量而已。</li>
<li>-h：以人们较易读的容量格式 (G/M) 显示；</li>
<li>-s：列出总量而已，而不列出每个各别的目录占用容量；</li>
<li>-S：不包括子目录下的总计，与 -s 有点差别。</li>
</ul>
```

<li>-k : 以 KBytes 列出容量显示; </li>  
<li>-m : 以 MBytes 列出容量显示; </li>  
</ul>  
<blockquote>  
<p>磁盘挂载与卸载</p>  
</blockquote>  
<p>根文件系统之外的其他文件要想能够被访问, 都必须通过“关联”至根文件系统上的某个目录来  
现, 此关联操作即为“挂载”, 此目录即为“挂载点”, 解除此关联关系的过程称之为“卸载” </p>  
<p>Linux 的磁盘挂载使用 mount 命令, 卸载使用 umount 命令。</p>  
<p>磁盘挂载语法: </p>  
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">mount [-t 文件系统] [-L Label名] [-o 额外选项] [-n] 装置文件名 挂载点  
</span></span></code></pre>  
<p>磁盘卸载命令 umount 语法: </p>  
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">umount [-fn] 装置文件名或挂载点  
</span></span></code></pre>  
<p>选项与参数: </p>  
<ul>  
<li>-f : 强制卸载! 可用在类似网络文件系统 (NFS) 无法读取到的情况下; </li>  
<li>-n : 不升级 /etc/mstab 情况下卸载。</li>  
</ul>  
<h3 id="写在最后">写在最后</h3>  
<p>学习内容以及总结参考 <strong>B 站狂神说 Java</strong></p>  
<p><a href="https://www.bilibili.com/video/BV187411y7hF">https://www.bilibili.com/video/  
V187411y7hF</a></p>