

Linux - 用户和权限

作者: [douniwan](#)

原文链接: <https://ld246.com/article/1573485320483>

来源网站: [链滴](#)

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

                      

-l: 例出当前的设置。由非特权用户来确定他们的密码或帐号何时过期。 </p>

</blockquote>

<p> (3) /etc/group</p>

<p>查看组信息，组信息的显示有四个字段</p>

第一个字段：组名

第二个字段：组密码占位符 --> x

第三个字段：组 id

第四个字段：组成员

<p>一个用户只能有一个主要组，最多可以有 31 个附加组。主要组是用户创建文件时默认的所有组附加组主要用于权限管理。不论用户属于哪个组，用户都能拥有该组的权限。</p>

<p>特殊组 wheel: </p>

<p>在 Linux 中有一个特殊组 wheel，wheel 组就类似于一个管理员的组。在 linux 中，即使我们有统管理员 root 的权限，也不推荐用 root 用户登录。一般情况下用普通用户登录就可以了，在需要 root 权限执行一些操作时，再 su 登录成为 root 用户。但是，任何人只要知道了 root 用户的密码，就可以通过 su 命令来登录为 root 用户--这无疑为系统带来了安全隐患。所以，将普通用户加入到 wheel 组中，被加入的这个普通用户就成了管理员组内的用户了，然后可以修改配置文件使得只有 wheel 内的用户可以切换到 root 用户。</p>

<p> (4) id</p>

<p>id 命令可以用来查看某个用户是否存在与系统中。</p>

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

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

<p> (1) useradd</p> <p>通过 useradd 命令，我们可以为系统新建一个用户</p> <blockquote> <p>c< 备注 > 加上备注文字。备注文字会保存在 passwd 的备注栏位中。</p> <p>-d< 登入目录 > 指定用户登入时的起始目录。</p> <p>-D 变更预设值。</p> <p>-e< 有效期限 > 指定帐号的有效期限。</p> <p>-f< 缓冲天数 > 指定在密码过期后多少天即关闭该帐号。</p> <p>-g< 群组 > 指定用户所属的群组。</p> <p>-G< 群组 > 指定用户所属的附加群组。</p> <p>-m 自动建立用户的登入目录。</p> <p>-M 不要自动建立用户的登入目录。</p> <p>-n 取消建立以用户名称为名的群组。</p> <p>-r 建立系统帐号。</p> <p>-s 指定用户登入后所使用的 shell。</p> <p>-u 指定用户 ID。</p> </blockquote> <p>新建用户的过程中，系统做了什么？</p> <p>useradd 只是创建了一个用户名，如 (useradd + 用户名)，它并没有在/home 目录创建同名文件夹，也没有创建密码，除了添加一个新用户之外什么都没有。因此利用这个用户登录系统，是登录不了的，为了避免这样的情况出现，可以用 (useradd -m + 用户名) 的方式创建，它会在/home 目录下创建同名文件夹，然后利用 (passwd + 用户名) 为指定的用户名设置密码。</p> <p>网上很多人都说 useradd 用户后，会生成自己的家目录什么的，但是我在自己的 ubuntu 上发并没有，用这个命令创建一个用户后，还有很多需要自己去完善的東西，才能真正的使用这个用户。</p> <p> (2) adduser</p> <p>adduser，这个命令实际是一个 perl 脚本，是 useradd 等类似底层命令的更友好的前端，它会 原文链接: [Linux - 用户和权限](#)

交互性的方式建立新用户，使用它可以指定新用户的家目录，登录密码，是否加密主目录等等，它会

```
</p>
<blockquote>
<p>1: 建立一个新目录作为家目录</p>
<p>2: 建立同名新组</p>
<p>3: 把用户的主要组设为该组(除非命令选项覆盖以上默认动作，比如-disall-homdirecry 之类)</p>
<p>4: 从/etc/SKEL 目录下拷贝文件到家目录，完成初始化</p>
<p>5: 建立新用户的密码</p>
<p>6: 如果其存在的话，还会执行一个脚本。</p>
</blockquote>
<p>用户创建时，默认的属性（比如 UID, GID, 是否创建家目录，创建邮箱等）都是通过/etc/login
defs 文件控制的，修改此文件的属性，会影响以后创建的所有用户。也可以创建用户时指定参数修改
这样只对当前创建用户有效</p>
<p>另外，adduser 的可用参数与上面的 useradd 一致。</p>
<p><strong>总结：新建用户，系统应该做得事。</strong></p>
<ol>
<li>新建用户时，系统会将 /etc/skel 中的目录及文件拷贝到新建用户的家目录中</li>
<li>在 /var/spool/mail 中，新建用户名的邮箱</li>
<li>在 /etc 下的 passwd、shadow、group 文件中，增加用户信息</li>
</ol>
<p><strong> (3) userdel</strong></p>
<p>通过 userdel 可以删除系统中的用户。</p>
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span c
ass="highlight-cl">userdel -r 用户名 <span class="highlight-c1">#删除用户一定记得加 -r 参
!! </span>
</span></span></code></pre>
<ul>
<li>不加 -r 参数，只删除 passwd、shadow 和 group 文件中的用户信息，/home 目录下的文件不
除，/var/spool/mail/ 下的文件不删除</li>
<li>加 -r 参数，删除 passwd、shadow 和 group 文件中的用户信息，同时删除用户的家目录和邮
</li>
</ul>
<p><strong> (4) passwd</strong></p>
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span c
ass="highlight-cl">passwd add <span class="highlight-c1">#给a用户设置密码</span>
</span></span></code></pre>
<p><strong> (5) usermod</strong></p>
<p>usermod 用来修改用户帐号的各项设定。</p>
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span c
ass="highlight-cl">usermod <span class="highlight-o">[</span>-LU<span class="highlight
o">][</span>-c &lt;备注&gt;<span class="highlight-o">][</span>-d &lt;登入目录&gt;<span c
ass="highlight-o">][</span>-e &lt;有效期限&gt;<span class="highlight-o">][</span>-f &lt;
冲天数&gt;<span class="highlight-o">][</span>-g &lt;群组&gt;<span class="highlight-o">][
/span>-G &lt;群组&gt;<span class="highlight-o">][</span>-l &lt;帐号名称&gt;<span class="h
ghlight-o">][</span>-s &lt;shell&gt;<span class="highlight-o">][</span>-u &lt;uid&gt;<sp
n class="highlight-o">][</span>用户帐号<span class="highlight-o">]</span>
</span></span></code></pre>
<p>可用参数</p>
<blockquote>
<p>-c&lt; 备注 &gt; 修改用户帐号的备注文字。</p>
<p>-d 登入目录 &gt; 修改用户登入时的目录。</p>
<p>-e&lt; 有效期限 &gt; 修改帐号的有效期限。</p>
<p>-f&lt; 缓冲天数 &gt; 修改在密码过期后多少天即关闭该帐号。</p>
</blockquote>
```

<p>-g 群组 修改用户所属的群组。 </p>
<p>-G 群组 修改用户所属的附加群组。 </p>
<p>-l 帐号名称 修改用户帐号名称。 </p>
<p>-L 锁定用户密码，使密码无效。 </p>
<p>-s 修改用户登入后所使用的 shell。 </p>
<p>-u 修改用户 ID。 </p>
<p>-U 解除密码锁定。 </p>
<p>-a 默认情况下，当用户已经存在附加组时，再添加附加组则会把之前的附加组给替换了，加 -a 参数，则不替换原来的附加组，意味着该用户可以有多个附加组。 </p>

<p>虽然 usermod 和 passwd 这两个命令都可以锁定和解锁用户，但是还是有区别的。区别之一就是 passwd 命令操作完后会有提示。还有一个区别就是 passwd 的权限比 usermod 大，使用 usermod 锁定的用户可以用 passwd 来解锁，但是使用 passwd 锁定的用户不能用 usermod 来解锁 </p>

<h4 id="组管理">组管理</h4>

<p>(1) groupadd</p>

<p>groupadd 命令用于创建一个新的工作组，新工作组的信息将被添加到系统文件中。 </p>

<p>相关文件:</p>

/etc/group 组账户信息。

/etc/gshadow 安全组账户信息。

/etc/login.defs Shadow 密码套件配置。

<p>用法</p>

```
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl">groupadd <span class="highlight-o">[</span>-g gid <span class="highlight-o">[</span>-o<span class="highlight-o">]]</span> <span class="highlight-o">[</span>-r<span class="highlight-o">]</span> <span class="highlight-o">[</span>-f<span class="highlight-o">]</span> group</span></code></pre>
```

<p>可用参数</p>

<blockquote>

<p>-g: 指定新建工作组的 id; </p>

<p>-r: 创建系统工作组，系统工作组的组 ID 小于 500; </p>

<p>-K: 覆盖配置文件 "/ect/login.defs"; </p>

<p>-o: 允许添加组 ID 号不唯一的工作组。 </p>

<p>-f,--force: 如果指定的组已经存在，此选项将失明了仅以成功状态退出。当与 -g 一起使用，并指定的 GID_MIN 已经存在时，选择另一个唯一的 GID (即-g 关闭) 。 </p>

</blockquote>

<p>(2) groupdel</p>

<p>需要从系统上删除群组时，可用 groupdel(group delete)指令来完成这项工作。倘若该群组中包括某些用户，则必须先删除这些用户后，方能删除群组。 </p>

<p>用法</p>

```
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl">groupdel <span class="highlight-o">[</span>群组名称<span class="highlight-o">]</span></code></pre>
```

<p>可选参数</p>

<blockquote>

<p>-r, --remove 删除主目录和邮件池</p>

</blockquote>

<p>(3) groupmod</p>

<p>需要更改群组的识别码或名称时，可用 groupmod 指令来完成这项工作。 </p>

<p>用法</p>

```
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl">
```



```
ass="highlight-cl">groupmod <span class="highlight-o">[</span>-g &lt;群组识别码&gt; &lt;o&gt;<span class="highlight-o">][</span>-n &lt;新群组名称&gt;<span class="highlight-o">]
</span>群组名称<span class="highlight-o">]</span>
</span></span></code></pre>
```

<p>可选参数</p>

<blockquote>

<p>-g < 群组识别码 > 设置欲使用的群 id。</p>

<p>-o 重复使用群组 id。</p>

<p>-n < 新群组名称 > 设置欲使用的群组名称。</p>

</blockquote>

<p> (4) groupmems</p>

<p>更改和查看组成员 (只有 root 可以执行) </p>

<p>用法</p>

```
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl">groupmems <span class="highlight-o">[</span>options<span class="highlight-o">]
</span> <span class="highlight-o">[</span>-g GROUPNAME<span class="highlight-o">]
</span></span></code></pre>
```

<p>可选参数</p>

<blockquote>

<p>-a 用户名 将一个用户添加到组成员列表</p>

<p>-d 用户名 从组成员列表中删除用户</p>

<p>-g 组名 超级用户可以指定修改哪个组的组成员列表</p>

<p>-l -g 组名 列出组成员</p>

<p>-p -g 组名 从组成员列表中删除所有用户</p>

</blockquote>

<p>sudo 是一种权限管理机制, 依赖于/etc/sudoers, 其定义了授权给哪个用户可以以管理员的身能够执行什么样的管理命令; </p> <p>格式: sudo -u USERNAME COMMAND</p> <p>默认情况下, 系统只有 root 用户可以执行 sudo 命令。需要 root 用户通过使用 visudo 命令编辑 sudo 的配置文件/etc/sudoers, 才可以授权其他普通用户执行 sudo 命令。</p> <p>su 为 switch user, 即切换用户的简写。</p> <p>格式为两种: </p> <p>su -l USERNAME (-l 为 login, 即登陆的简写) </p> <p>su USERNAME</p> <p>如果不指定 USERNAME (用户名), 默认即为 root, 所以切换到 root 的身份的命令即为: su -oot 或 su -, su root 或 su。</p> <p>su USERNAME, 与 su - USERNAME 的不同之处如下: </p> <p>su - USERNAME 切换用户后, 同时切换到新用户的工作环境中。</p> <p>su USERNAME 切换用户后, 不改变原用户的工作目录, 及其他环境变量目录。</p> <p>su -, su -l 或 su --login 命令改变身份时, 也同时变更工作目录, 以及 HOME, SHELL, USE, LOGNAME。此外, 也会变更 PATH 变量。用 su -命令则默认转换成成 root 用户了。</p> <p>而不带参数的“su 命令”不会改变当前工作目录以及 HOME,SHELL,USER,LOGNAME。只是拥有了 root 的权限而已。</p> <p>su -使用 root 的密码,而 sudo su 使用用户密码</p> <p>在 Linux 中有严格的权限等级, 不同的用户可能对与同一文件拥有不同权限。那么怎么查看文件权限呢? 我们一般可以使用 <code>ls -l</code> 查看当前目录下的各个文件的权限。</p> <p>ls -l 命令的输出结果类似如下: </p> ``` <pre><code class="language-bash highlight-chroma"> ``` 原文链接: [Linux - 用户和权限](#)

```
ass="highlight-cl">drwxr-xr-x <span class="highlight-m">2</span> huny huny <span clas
="highlight-m">4096</span> 11月 <span class="highlight-m">10</span> 20:01 Demo
</span></span><span class="highlight-line"><span class="highlight-cl">drwxr-xr-x <span
lass="highlight-m">3</span> huny huny <span class="highlight-m">4096</span> 11月 <
pan class="highlight-m">10</span> 14:30 GpssWorld
</span></span><span class="highlight-line"><span class="highlight-cl">-r--r--rw- <span c
ass="highlight-m">1</span> huny huny <span class="highlight-m">0</span> 11月 <sp
n class="highlight-m">7</span> 21:13 <span class="highlight-nb">test</span>
</span></span><span class="highlight-line"><span class="highlight-cl">-rw-r--r-- <span c
ass="highlight-m">1</span> huny huny <span class="highlight-m">1080</span> 11月 <
pan class="highlight-m">6</span> 20:52 vimTest.txt
</span></span><span class="highlight-line"><span class="highlight-cl">drwxr-xr-x <span
lass="highlight-m">3</span> huny huny <span class="highlight-m">4096</span> 11月 <
pan class="highlight-m">5</span> 09:29 安卓实验课
</span></span><span class="highlight-line"><span class="highlight-cl">-rw-r--r-- <span c
ass="highlight-m">1</span> huny huny <span class="highlight-m">178333</span> 11月
span class="highlight-m">7</span> 21:35 深度截图_20191107213501.png
</span></span></code></pre>
```

<p>在这个输出中，前面的最前面的一个 <code>- </code> 和 <code>d</code> 代表文件的不类型，后面的 <code>rwrxr-x</code> 这种形式的表示文件的权限情况。<code>rwrxr-x</code> 这里面其实是九个字符，前三个代表了文件属主的权限，中间三个代表了文件属组的权限，后面一个代表了其他用户的权限。然后后面的数字代表了文件的链接数。接着的 huny huny，其中的第一个文件所属的用户，第二个是所属的组。接着是文件的大小，时间，和文件名。</p>

<p>在 Linux 中一共有七中文件类型：</p> - -: 普通文件 - d: 目录文件 - b: 块设备文件 - c: 字符特殊文件 - l: 符号连接（类似与 windows 的快捷方式） - f: 命名管道 - s: 套接字文件 <p>知道什么符号代表什么文件过后，我们就可以去看该文件的权限类型了（同意权限针对不同的文类型表达的意思不一样。）</p> <p>Linux 中的权限可以用字符来表示，也可以用数字来表示</p> <thead> |<tr> | | <th>权限</th> | <th>对应数字</th> | <th>意义</th> | |</tr> | | </tbody> |<tr> | | <td>r</td> | <td>4</td> | <td>可读</td> | |</tr> | | |<tr> | | <td>w</td> | 原文链接: [Linux - 用户和权限](#)

2	可写
x	1
可执行	

对目录来说

r: 可以查询目录下的文件 (ls ll 命令)

w: 具有修改目录结构的权限, 比如新建文件和目录, 删除此目录下文件和目录 (cp mv touch m 命令)

x: 目录有执行权限但是不能运行, 可以进入目录 (cd 命令)

对文件来说最高权限是 x, 对于目录来说最高权限是 w, 一般给目录赋予权限 0, 5 (rx), 7 (rw), 赋予 4, 1, 6 都是没有意义的。对于文件能否删除, 首先要对目录具有执行权限, 同时对文也具有执行权限。

如何修改文件权限

chmod

chmod 命令可以用来变更文件或目录的权限。

用法: `chmod [options] mode files`

mode : 权限设定字串, 格式如下:

`[ugoa...][[+ -=][rwxX]...][,...]`

其中:

- u 表示该文件的拥有者, g 表示与该文件的拥有者属于同一个群体(group)者, o 表示其他以外人, a 表示这三者皆是。u 用户 g 组 o 其它 a 所有用户(默认)
- + 表示增加权限、- 表示取消权限、= 表示唯一设定权限。
- r 表示可读取, w 表示可写入, x 表示可执行, X 表示只有当该文件是个子目录或者该文件已经设定过为可执行。

其他参数说明:

- c : 若该文件权限确实已经更改, 才显示其更改动作 (-c, --changes 只输出被改变文件的信息)
- f : 若该文件权限无法被更改也不要显示错误讯息 (--silent, --quiet, 当 chmod 不能改变文件格式时, 不通知文件的用户)
- v : 显示权限变更的详细资料
- R : 对目前目录下的所有文件与子目录进行相同的权限变更(即以递归的方式逐个变更)--recursive 可递归遍历子目录, 把修改应用到目录下所有文件和子目录
- help : 显示辅助说明、输出帮助信息。
- version : 显示版本
- reference=filename 参照 filename 的权限来设置权限
- v, --verbose 无论修改是否成功, 输出每个文件的信息

<p>例子: </p>

```
<pre> <code class="language-bash highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> chmod u+r 文件名 <span class="highlight-c1"> #给属主加读权限</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> chmod a+x 文件名
<span class="highlight-c1"> #给所有用户和其他人加执行权限</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> chmod g-w file1
<span class="highlight-c1"> #file1的属组去掉写权限</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> chmod <span class="highlight-nv"> o</span> <span class="highlight-o"> =</span> x 文件名 <span class="highlight-c1"> #其他用户只有执行权限</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> chmod <span class="highlight-nv"> a</span> <span class="highlight-o"> =</span> - 文件名 <span class="highlight-c1"> #去除所有用户的权限</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> chmod <span class="highlight-nv"> ug</span> <span class="highlight-o"> =</span> rw,o <span class="highlight-o"> =</span> r file <span class="highlight-c1"> #给属主属组读写权限, 给其他用户读权限</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> chmod <span class="highlight-m"> 644</span> file <span class="highlight-c1"> #给属组读写权限, 其
用户和属组为读权限</span>
</span> </span> </code> </pre>
```

<h4 id="chown">chown</h4>

<p>用来改变文件或目录的所有者和所属用户组</p>

<p>用法: <code>chown [-cfhvR] [--help] [--version] user[:group] file...</code> </p>

<blockquote>

<p>参数说明: </p>

- user : 新的文件拥有者的使用者 ID
- group : 新的文件拥有者的使用者组(group)
- -c : 显示更改的部分的信息
- -f : 忽略错误信息
- -h : 修复符号链接
- -v : 显示详细的处理信息
- -R : 处理指定目录及其子目录下的所有文件
- --help : 显示辅助说明
- --version : 显示版本

</blockquote>

<p>例子: </p>

```
<pre> <code class="language-bash highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> chown root:root vimTest.txt <span class="highlight-c1"> #将vimTest.ja
a的属主和属组改为root: root</span>
</span> </span> </code> </pre>
```

<h4 id="chgrp">chgrp</h4>

<p>可以单独更改属组</p>

<p>用法: <code>chgrp [-cfhRv][--help][--version][所属群组][文件或目录...] 或 chgrp [-cfhRv][--help][--reference=<参考文件或目录>][--version][文件或目录...]</code> </p>

<blockquote>

<p>参数说明: </p>

<p> -c 或--changes 效果类似"-v"参数, 但仅回报更改的部分。 </p>

<p> -f 或--quiet 或--silent 不显示错误信息。 </p>

<p> -h 或--no-dereference 只对符号连接的文件作修改, 而不更动其他任何相关文件。 </p>

<p> -R 或--recursive 递归处理, 将指定目录下的所有文件及子目录一并处理。 </p>

<p> -v 或--verbose 显示指令执行过程。</p>

<p> --help 在线帮助。</p>

<p> --reference=< 参考文件或目录 > 把指定文件或目录的所属群组全部设成和参考文
或目录的所属群组相同。</p>

<p> --version 显示版本信息。</p>

</blockquote>

<p>例子: </p>

```
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl">chgrp root vimTest.txt      <span class="highlight-c1">#将vimTest.java  
属组改为root</span>  
</span></span></code></pre>
```