https://www.jianshu.com/p/b587d91b8127

**Linux探索之旅 | 第二部分第五课：用户和权限，有权就任性**

[](https://www.jianshu.com/u/44339a8a9afa)

[程序员联盟](https://www.jianshu.com/u/44339a8a9afa) 关注

2016.11.09 06:30\* 字数 8876 阅读 4825评论 34喜欢 90

作者 谢恩铭 转载请注明出处  
公众号「[程序员联盟](https://link.jianshu.com/?t=https://github.com/frogoscar/ProgrammerLeague/blob/master/qrcode.jpg)」（微信号：ProgrammerLeague ）  
原文：[http://www.jianshu.com/p/b587d91b8127](https://www.jianshu.com/p/b587d91b8127" \t "_blank)

[《Linux探索之旅》](https://www.jianshu.com/nb/5470392)全系列

**内容简介**

1. 第二部分第五课：用户和权限，有权就任性
2. 第二部分第六课预告：Nano文本编辑器和终端配置

**用户和权限，有权就任性**

上一课 [Linux探索之旅 | 第二部分第四课：文件操纵，股掌之中](https://www.jianshu.com/p/45904376a66f) 中，我们学了不少很有用的基础知识。

今天的标题也挺任性的，小编自认是一个很本分的人（真的吗？），起这种标题也是“情非得已”。

第二部分的前几课我们讲解了终端，命令行，文件和目录，文件操作。都算是Linux的重要基础。

这一课我们来讲一个极为重要的内容，可以说是Linux基础中的重点：用户和权限。

我们知道Linux是一个多用户系统，所以可以实现多人同时不同地登录一个Linux系统，而每个人有不同的权限，这些权限规定了每个人能够做什么，不能够做什么。

就好比一个市政厅，虽然可以有很多人同时在里面办公，但是不同层级的人能够做的事是不一样的。

正因为多用户这个特性，所以更要严加管理每个用户，以防止互相干扰，甚至越权行为。因为万一有居心叵测的用户拥有了高度权限，对所有用户岂不是一种威胁么。

所以，Linux系统中的每个用户都有自己独立的账号，而且其账号受着权限的制约。

是不是觉得跃跃欲试，我保证这一课会很有意思的。而且再提一次，这一课极为重要！

**sudo命令：以root身份运行命令**

在我们安装Ubuntu系统的时候，到了最后几步，安装程序会让我们创建自己的用户名。比如，小编就创建了名为oscar的用户。

在大多数的Linux发行版中（Ubuntu是一个Linux发行版），我们都建议初学者创建一个有限权利的账户。小编的oscar这个用户就是有限权限的账户。

也许你会问：“等一下，有没有搞错？这个Linux系统是我安装的，我还不能想干什么就干什么了？为什么我的权利是有限的？那我还怎么任性啊？”

是的，而且这是一种安全机制。当然了，当我们在终端中使用命令行的时候，我们随时可以说：我想要切换到什么都可以做的用户的身份。但是，默认情况下，我们是没有为所欲为的权限的，这会比较安全。

因为有些命令会对Linux系统的稳定性和安全性造成威胁。具有有限的权限，意味着我们不能随随便便就运行一些可怕的命令，例如那个“草木为之含悲，风云因而变色”的命令：

rm -rf /

我们在上一课讲过，rm -rf / 这个命令会删除你的所有东西。所以江湖上人人闻之色变，除了颤抖还是颤抖。

接下来，我们先了解一下Linux中用户是如何组织的，然后再来学习如何变身为超级用户。

**Linux下的用户组织**

在Linux中，理论上说来，我们可以创建无数个用户。但是这些用户是被划分到不同的群组里面的。

有一个用户，名叫root，是一个很特殊的用户。字大管家，号超级用户，江湖诨号“有钱任性”。因为在Linux系统中，它可以做任何事情。

我们用下图来演示一下Linux中的用户组织的一般可能形式：

上图中的群组，可以有好多不同的，我们举了“家人”和“朋友”两个例子而已。

在我们装完Linux系统后，假如我们创建的个人用户叫做oscar，那么这时系统中只有两个用户：root和oscar。

root用户是默认会创建的，因为是系统的大管家，超级用户。

oscar则是小编自己创建的用户。

我们只在必要时才会切换为root身份。这一课中你将学习到的几个命令须要root身份才能运行。

其他时候，我们都是用有限权限的用户：oscar（当然，这是小编的情况，你的个人用户可能叫xiaoming）。

这一简单的保护措施大大地提高了Linux系统的安全性，有效防止误操作或是病毒的攻击。因为如果你是以个人用户身份登录系统，被病毒控制了电脑，那么它也只能做有限的事，不能为所欲为。但假如你以root用户登录系统，那么劫持你的病毒就可以胡作非为了，甚至毁了你的系统。

在Windows系统中，默认是以系统大管家或管理员账户登录，叫做Administrator。这也是Windows系统较Linux系统更加容易被病毒攻击得逞的原因。

例外：Ubuntu系统是Linux系统中一个特例，它默认不允许以root用户登录系统。root用户存在，但是你不能直接登录。我们接下来会学习如何间接登录root。Ubuntu的开发者解释说他们考虑到对于初学者，能登录root用户太危险。

秘：但其实我们还是可以通过设置来允许Ubuntu直接登录root的。自己可以百度/Google。我就不把此方法告诉大家了，免得毁了系统来扔我鸡蛋。

**sudo命令：暂时成为root**

默认地，我们是以个人用户身份登录Ubuntu系统。在小编的情况，是oscar这个用户。

在Ubuntu中默认不允许在开机时直接以root用户登录，那么我们要如何来运行一些只有root才有权利运行的命令呢？

别担心，我们可以使用sudo这个命令暂时成为root，当一天的皇帝也是很开心的嘛。

sudo是英语Substitute User DO的缩写，substitute是“替换，代替，替身”的意思，user是“用户”的意思，do就是“做”的意思。所以连在一起就是“替换用户来执行...”的意思。

因此，如果我们要运行只有root才可以运行的命令，那么可以在此命令前面加上sudo命令，如下：

sudo command

终端会提示你输入密码，至少第一次会要求输入密码。此密码就是你个人用户的密码，在小编的情况，就是我的oscar这个账户的密码。

比如，我们可以用sudo命令配合date命令运行一下（放心，运行date命令不会有什么危险）：

sudo date

可以看到，在我们输入 sudo date后，回车，终端要求我们输入个人用户的密码，就是那一行：

[sudo] password for oscar:

意思是输入oscar这个用户的密码（password是英语“密码”的意思）。

小编输入了oscar账户的密码，虽然输入密码时是看不到任何字符的，这是为了安全。

输入密码之后，回车，date命令的运行结果就显示给我们了，还记得我们的date命令么？是的，它用于显示当前日期和时间。

当然了，date这个命令并一定要是root用户才能执行，所以没什么特别，只是给大家一个印象，如何用sudo命令暂时切换为root身份。

**sudo su命令：一直成为root**

有的人当一天皇帝就知足了，但有的人想要当得更久一些，甚至《向天再借五百年》，怎么办呢？

只需要使用 sudo su 这个组合命令，就可以一直成为root用户了。

终端会要求我们输入个人用户的密码，就是那一行：

[sudo] password for oscar:

可以看到，运行了sudo su之后，我们的命令行提示符从 oscar@oscar-laptop:~$ 变成了 root@oscar-laptop:/home/oscar#

如果你还记得我们在命令行那一课里讲到的如何阅读命令行提示符，你就会很容易明白此时我们已经完成了“屌丝逆袭”：

从oscar升级为root用户了！（从第一个root可以看出，还有结尾处的#也表明此时是root身份）。

这个时候就要小心了，因为此时你已经可以为所欲为了。

当然了，如果你感觉有点不适应，讶异于幸福来得太突然，还留恋屌丝的身份，那么可以退出root用户，重新回到个人用户身份，用 exit命令就可以完成（exit是英语“退出，离开”的意思）：

可以看到，当我们身为root时，只要运行exit命令，就随时可以回到个人用户。

在Ubuntu以外的其他Linux发行版，也许切换到root账户，不是用sudo su，而是只需要su就可以了。当然我们比较建议再加一个横线，用

su -

这样不仅能切换为root，还可以直接定位到root的家目录。而且可以直接使用root才能用的那些命令。

**用户管理的命令**

既然你已经学会了如何暂时或一劳永逸地成为root用户，那么我们就可以来学习一些只有root用户才能运行的命令了。

adduser命令就是其中之一。

**adduser命令：添加新用户**

adduser这个命令，我们很容易理解其作用，因为完全可以顾名思义：add是英语“添加”的意思，user是英语“用户”的意思，因此adduser就是用于添加用户。

用法也很简单，adduser命令后接我们要创建的用户名。

如果你使用你的个人用户身份来运行这条命令，终端会提示你没有权限：

可以看到，我们在oscar这个用户的身份下，运行 adduser thomas，表示我们想要创建一个新用户，叫做thomas。

但是不能运行，终端提示“adduser: Only root may add a user or group to the system”。

这句英语翻出来就是“adduser：只有root用户才能往系统里添加用户或群组”。

太看不起人了，我不变身为钢铁侠你以为我只是托尼.斯塔克是吧？

所以我只能穿上sudo su这套“钢衣”，先变身为钢铁侠，再来运行adduser thomas这个命令。

可以看到，我们成为root之后，运行adduser thomas就没问题了，显示thomas用户被添加，用户名是thomas，群组是thomas，创建/home/thomas这个thomas的家目录，然后终端命令停在这一行：

Enter new UNIX password：

意思是“请为thomas这个用户创建一个密码”。

我们随便填写，然后回车，会让我们再输入一遍密码确认。然后接着是一些配置信息，比如thomas这个用户的全名，房间号，工作电话，家庭电话，其他。我们通通可以不填写，用回车键略过，到了最后终端会提示：

Is the information correct ? [Y/n]

意思是“以上这些信息都正确吗？[正确请输入y/不正确输入n]”。

我们输入y，回车。thomas用户就正式创建好了。可以看一下此时/home目录里的内容，发现多了一个 thomas 目录。也就是thomas这个用户的家目录。如下图：

**passwd命令：修改密码**

如果之后你对设定的密码不满意，那么可以用passwd命令来修改当前密码。

passwd命令是password这个英语单词的缩写，表示“密码”。

用法也类似adduser，只要在其后加上需要修改密码的那个用户名，例如：

passwd thomas

Enter new UNIX password的意思是“输入新的密码”

Retype new UNIX password的意思是“再输入一遍密码”

输入两次相同的新密码之后，可以看到提示：password updated successfully，意思是“密码更新成功”。

**deluser：删除用户**

既然有添加用户的命令，那么也一定有删除账户的命令。

是的，我们可以用deluser命令来删除已创建的账户。

deluser是delete和user的缩写，delete是英语“删除”的意思，user是“用户”的意思。

用法同adduser，在后面接你要删除的用户名。例如：

deluser thomas

可以看到，运行deluser thomas这个命令，终端不会提示你确认是否删除，而是直接删除了用户thomas。

所以，deluser这个命令还是要谨慎使用。

因为如果你删除的是你的个人用户，例如小编的情况，假如我 deluser oscar。

那么就会陷入囧境：在下次系统启动时你就不能以oscar登录了，而Ubuntu默认又不允许root登录，你就会不知所措了。

单单用deluser命令，不加参数的话，只会删除用户，但是不会删除在/home目录中的用户家目录。如果你想要连此用户的家目录也一并删除，可以加上 --remove-home 这个参数，如下：

deluser --remove-home thomas

这样，不仅删除了thomas这个用户，连/home/thomas这个目录也会删除。

注意：adduser和deluser命令只是Debian一族（包括Ubuntu）才有的命令。其他的LInux发行版，一般来说，添加用户和删除用户是用useradd和userdel命令。  
而且，用useradd添加用户之后，在默认的情况下，该账号是暂时被封锁的， 也就是说，该账号是无法登录，须要用passwd命令来给新创建的用户设置密码之后才可以使用。

**群组管理的命令**

在这课的开头，我们说了，Linux中每一个用户都属于一个特定的群组。

那你要问了：“那么我们刚才创建的thomas是属于哪个群组呢？我们以前创建的个人用户oscar又属于哪个群组呢？我们之前都没配置呀。”

事实上，如果你不设置用户的群组，那么它默认会创建一个和它的用户名一样的群组，并且把用户划归到这个群组。

我们可以用ls -l命令来看一下/home目录下的内容：

可以看到，我们的oscar用户和thomas用户的家目录分别是/home/oscar和/home/thomas。

在每一行的各列都有不同意义，我们之前的课程里有讲解过（[Linux探索之旅 | 第二部分第三课：文件和目录，组织不会亏待你](https://www.jianshu.com/p/4146c47b5865)）。

所以，第三列表示文件或目录的所有者，第四列表示文件或目录的所在群组。

可以注意到：oscar这个目录的所有者是oscar，群组是oscar；thomas这个目录的所有者是thomas，群组是thomas。

我们也注意到其他的两个：image.bnp这个文件的所有者是root，群组是root；swaroop这个目录的所有者是root，群组是root。所以其实root这个群组是存在的，root用户就属于这个群组。

但是，把用户分在不同的群组，到底有何意义呢？

在用户不多的时候，我们会觉得一个用户属于一个群组（比如默认是与用户名相同的群组名）是挺不错的。但是一旦用户一多，你可能就想要创建群组了。

我们来学习如何管理群组。当然，群组还有权限的考量因素，我们这课之后会讲权限。

当然了，群组管理的命令也需要root身份。

**addgroup：创建群组**

addgroup是add和group的缩写，add是英语“添加”的意思，group是英语“群组”的意思。所以addgroup命令用于添加一个新的群组。

用法也很简单，和adduser命令类似，后接需要创建的群组名。例如：

addgroup friends

创建一个名为friends的群组，friends是英语“朋友”的意思，也是美剧《老友记》，哈哈。

在上图中，我们看到，用addgroup命令创建了一个新的群组，名叫friends，而且成功了。Done是英语“完成”的意思。

很不错，不过目前friends这个群组还是空的，因为还没有往里面添加用户呢。

**usermod命令：修改用户账户**

usermod是user和modify的缩写，user是英语“用户”的意思，modify是“修改”的意思。usermod命令用于修改用户的账户。

usermod命令有好多参数，可以实现不同的功能。不过我们暂时只需要记得它的两个参数：

* -l：对用户重命名，但是/home目录中的用户家目录名不会改变，需要手动修改。
* -g：修改用户所在群组

用法很简单，假如我要将thomas这个用户放到我刚创建的friends这个群组里，可以这样写：

usermod -g friends thomas

我们知道，用户thomas之前的群组是thomas，默认的。运行完usermod -g friends thomas之后，thomas的群组就会变成friends了，但是在 /home/thomas这个目录的信息不变，仍旧显示群组是thomas。

我们怎么知道用户thomas的群组已经改变为friends了呢？

我们可以用groups命令，这个命令可以获知一个用户属于哪个（些）群组。

用法很简单，后接用户名就可以了，当然用户要存在才行。

可以看到，thomas的群组是friends；root的群组就是root；oscar的群组有好几个，说明小编加入了很多“组织”。

上图中，我们确实看到了，/home/thomas这个目录的信息没变，第四列表示群组的依旧是thomas。

当然我们也可以一次将一个用户添加到多个群组，就用 -G 参数（大写的G）。用法如下：

usermod -G friends,happy,funny thomas

以上命令把thomas添加到friends，happy和funny三个群组。记得群组名之间要用逗号分隔，而且没有空格。

注意：使用usermod时要小心，因为配合-g或-G参数时，它会把用户从原先的群组里剔除，加入到新的群组。如果你不想离开原先的群组，又想加入新的群组，可以在-G参数的基础上加上-a参数，a是英语append的缩写，表示“追加”。例如：

usermod -aG happy thomas

以上命令就把thomas追加到群组happy里了，这样thomas就属于两个群组：friends和happy

可以用groups命令测试一下。

注意：groups命令如果单独用，不加任何参数，会显示当前用户所在群组。

记得，追加群组的时候，一定要用大写的G这个参数，不能用小写的g这个参数，即使只追加一个群组。

**delgroup命令：删除群组**

delgroup是delete和group的缩写，delete是英语“删除”的意思，group是英语“群组”的意思。所以delgroup命令用于删除一个已存在的群组。

用法很简单，后接想要删除的群组名：

delgroup happy

就删除了happy这个群组。

再用groups命令测试，发现thomas只属于friends群组了。因为happy这个群组被删除了嘛。

注意：addgroup和delgroup命令只是Debian一族（包括Ubuntu）才有的命令。其他的LInux发行版，一般来说，添加用户和删除用户是用groupadd和groupdel命令。

**修改文件的所有者和群组**

只有root用户可以修改一个文件的所有者和群组。

比如说，小编自己的用户oscar的家目录有一个文件，file.txt，是我创建的。

我们用ls -l命令来看一下它的信息：

可以看到，file.txt的所有者和群组都是oscar。

现在我决定，把这个文件转让给thomas，也就是让file.txt的所有者变为thomas，怎么做呢？

**chown命令：改变文件的所有者**

此命令也需要root身份才能运行。

chown是change和owner的缩写，change是英语“改变”的意思，owner是英语“所有者”的意思。

因此chown命令用于改变文件的所有者。

用法也很简单，后接新的所有者的用户名，再接文件名。例如：

chown thomas file.txt

可以看到，用chown命令，把file.txt文件的所有者改为thomas之后，file.txt的所在群组是不变的，还是oscar。

正所谓“身在曹营心在汉”。那么如何使它“身在曹营心也在曹营”呢？

就要用到chgrp命令了。

**chgrp命令：改变文件的群组**

chgrp是change和group的缩写，change是英语“改变”的意思，group是英语“群组”的意思。

chgrp命令用于改变文件的群组。

用法也很简单，后接新的群组名，再接文件名。例如：

chgrp thomas file.txt

好了，这下file.txt的所有者和群组都是thomas了。

其实，chown命令也可以改变文件的群组，用法如下：

chown thomas:friends file.txt

这句命令就把file.txt这个文件的所有者改为thomas，群组改为friends了。用法也很简单，就是在所有者和群组之间用冒号隔开。

**-R参数：递归设置子目录和子文件**

chown命令的-R参数非常有用，还记得以前我们有些命令也会使用-R参数么？

是的，R是recursive的缩写，表示“递归”。所以如果chown命令配上-R参数，就会使得被修改的目录的所有子目录和子文件都改变所有者（或者连群组也改变，如果用上述冒号的方法来同时修改所有者和群组）。

例如，假如我突然变得“很坏”，想要把用户thomas的家目录的所有子目录和文件都占为己有。我可以这么做：

chown -R oscar:oscar /home/thomas

这样不但使/home/thomas这个目录的所有者和群组都变成oscar，而且其子目录和子文件也都是如此。

可以看到，/home/thomas都归我（oscar）所有了。

**chmod命令：修改访问权限**

好了，这一节我们要攻坚这一课最难的部分了：访问权限。

**权限的原理**

在Linux系统里，每个文件和目录都有一列权限属性。这一列访问权限指明了谁有读的权利，谁有修改的权利，谁有运行的权利。

我们其实早就见过访问权限了，是的，就在我们运行ls -l命令的时候，显示的每个文件或目录的第一列信息就是访问权限。

比如我们在/home/oscar/linux\_c目录下运行ls -l命令试试（当然你可以在你的家目录或其他目录运行也可以，因为小编的家目录东西太多了，比较乱）：

上图中文件信息的第一列比较复杂，我们可以看到不少 d，r，w，l，x等字母。如果不细分的话，这些我们可以通称为文件访问权限符。

以下列出我们看到的字母的含义：

* d：是英语directory的缩写，表示“目录”。就是说这是一个目录。
* l：是英语link的缩写，表示“链接”。就是说这是一个链接。
* r：是英语read的缩写，表示“读”。就是说可以读这个文件。
* w：是英语write的缩写，表示“写”。就是说可以写这个文件，也就是可以修改。
* x：是英语execute的缩写，表示“执行，运行”。就是说可以运行这个文件。

如果x权限在一个目录上，那么表示的是这个目录可以被读，也就是可以打开此目录来看其子目录和子文件，如果它同时有r权限的话。

如果相应位置有字母，表示有相应权限；如果相应位置是一个短横 -，则表示没有相应权限。

为什么我们看到这一排有好多个重复出现的r，w和x呢？

那是因为访问权限是按照用户来划分的：

如上图，除开第一个表示文件或目录属性的符号（此处是d，表示目录。如果是l，则是链接。还有其他字母，我们暂时不深究。如果是短横-，那么是普通文件。），其他的9个符号被划分为三组，从左到右分别表示：

* 第一组rwx表示文件的所有者对于此文件的访问权限。
* 第二组rwx表示文件所属的群组的其他用户对于此文件的访问权限。
* 第三组rwx表示除前两组之外的其他用户对于此文件的访问权限。

我们用一个具体的文件来作为例子分析一下：

可以看到，renamed\_file这个文件的访问权限是

-rw-r--r--

我们从左到右来分析这些符号都表示什么：

* -：第一个短横表示这是一个普通文件。如果此处是d，那么表示目录；如果是l，那么表示链接，等等。
* rw-：表明了文件的所有者（此处是oscar）对文件有读，写的权限，但是没有运行的权限。也很好理解，因为这是一个普通文件，默认没有可执行的属性。记住：如果有w权限（写的权限），那么表明也有删除此文件的权限。
* r--：表明文件所在的群组（此处是oscar）的其他用户（除了oscar之外）只可以读此文件，但不能写也不能执行。“可远观而不可亵玩焉”。
* r--：表示其他用户（除去oscar这个群组的用户）只可以读此文件，但不能写也不能执行。

综上所述，renamed\_file这个文件是一个普通文件，不是一个目录，也不是链接文件，它的所有者oscar可以读写它，但不能执行；其他的用户只能读。

那么root呢？对于此文件root用户的访问权限是什么呢？

记住：root是超级管家，它有所有权限，"只有它想不到的，没有它做不到的"。  
它可以读、写、运行任意文件。

**chmod命令：修改文件的访问权限**

既然我们已经学会了如何查看和理解文件的访问权限，我们就来学习如何修改文件的访问权限吧。

我们要用到chmod命令，这个命令也是Linux中常用的命令。

毕竟“一朝权倾天下有”，“争权夺利”谁不喜欢啊。开个玩笑...

开始讲解之前，要说明一点，chmod命令不需要是root用户才能运行。只要你是此文件的所有者，你就可以用chmod来修改文件的访问权限。

chmod是change和mode的缩写，change是英语“改变”的意思，mode是“模式”的意思。chmod命令用于修改文件的各种访问权限。

chmod这个命令充满魅力，因为它的用法不止一种，好像一个千面女郎，令人着迷。

最常见的用法应该是数字式的。

**用数字来分配权限：chmod的绝对用法**

我们接下来要做一些加法，大家准备好了吗？不要让小学数学老师哭晕在体育办公室哦。什么？你的小学数学是语文老师教的，好，算你厉害...

不要怕，只是做一些极为简单的加法，我们只要心算就可以了。

事实上，Linux系统对每种权限（r，w和x）分配了对应的数字：

| **权限** | **数字** |
| --- | --- |
| r | 4 |
| w | 2 |
| x | 1 |

所以，如果我们要合并这些权限，就需要做简单的加法了：将对应的数字相加。

假如我们要分配读，写权限，那么我们就要用4+2，就等于6。数字6表示具有读和写权限。

以下是可能的组合形式：

| **权限** | **数字** | **计算** |
| --- | --- | --- |
| --- | 0 | 0 + 0 + 0 |
| r-- | 4 | 4 + 0 + 0 |
| -w- | 2 | 0 + 2 + 0 |
| --x | 1 | 0 + 0 + 1 |
| rw- | 6 | 4 + 2 + 0 |
| -wx | 3 | 0 + 2 + 1 |
| r-x | 5 | 4 + 0 + 1 |
| rwx | 7 | 4 + 2 + 1 |

不难吧？

所以，对于访问权限的三组（所有者的权限，群组用户的权限，其他用户的权限），我们只要分别做加法就可以了，然后把三个和连起来。

例如：640分别表示：

* 文件的所有者有读和写的权限。
* 文件所在群组的其他用户具有读的权限。
* 除此之外的其他用户没有任何权限。

因此，我们可以给的最宽泛的权限就是 777：所有者，群组用户，其他用户都有读，写和运行的权限。这样，所有人就都可以对此文件“为所欲为”了。

相反，如果权限是000，那么没有人能对此文件做什么。当然，除了root，root可以做任何事。

我们现在来修改renamed\_file的权限试试：

chmod 600 renamed\_file

可以看到，我们的renamed\_file文件的访问权限被修改为了

rw-------

正好是600。

所以现在只有oscar可以读和写此文件，其他人都不能做什么。当然，除了root之外。

**用字母来分配权限：chmod的相对用法**

除了用数字，我们也可以用另一种方式来分配文件的访问权限：用字母。

原理是类似的，但是有时用字母的方式更加精巧，因为不需要一次性把三组权限都写出来。

我们需要知道不同的字母代表什么：

* u：user的缩写，是英语“用户”的意思。表示所有者。
* g：group的缩写，是英语“群组”的意思。表示群组用户。
* o：other的缩写，是英语“其他”的意思。表示其他用户。
* a：all的缩写，是英语“所有”的意思。表示所有用户。

当然了，和这些字母配合的还有几个符号：

* +：加号，表示添加权限。
* -：减号，表示去除权限。
* =：等号，表示分配权限。

接下来，我们举例说明如何使用：

#文件file.txt的所有者增加读和运行的权限。

chmod u+rx file.txt

#文件file.txt的群组其他用户增加读的权限。

chmod g+r file.txt

#文件file.txt的其他用户移除读的权限。

chmod o-r file.txt

#文件file.txt的群组其他用户增加读的权限，其他用户移除读的权限。

chmod g+r o-r file.txt

#文件file.txt的群组其他用户和其他用户均移除读的权限。

chmod go-r file.txt

#文件file.txt的所有用户增加运行的权限。

chmod +x file.txt

#文件file.txt的所有者分配读，写和执行的权限；群组其他用户分配读的权限，不能写或执行；其他用户没有任何权限。

chmod u=rwx,g=r,o=- file.txt

**-R参数：递归地修改访问权限**

-R参数可是“死性不改”，上一课可以配合cp命令来递归拷贝文件，这一课又来“捣蛋”：

chmod配合-R参数可以递归地修改文件访问权限。

假如我要只允许oscar这个用户能读，写，运行/home/oscar这个目录的所有文件（当然，root不算，root可以做任何事），该怎么做呢？

chmod -R 700 /home/oscar

就是这么简单。

终于结束了，长舒一口气。

**总结**

1. 在Linux系统中，每一个用户都有自己独立的账户。
2. 用户是被分组的。
3. 存在一个超级用户，也就是一般说的大管家，它有权利做任何事情，它就是root。它有权利安装软件，并且对系统设置做修改。
4. 有一些命令要能运行须要先切换到root身份，我们可以用sudo这个命令来实现此切换。比如以下这些命令就需要在前面再加一个sudo：adduser（用于添加新用户），deluser（用于删除用户），chmod（用于修改文件的权限）。
5. 我们可以用chmod命令来修改文件的访问权限。有三种权限：r（读权限），w（写权限）和x（运行权限）。

**第二部分第五课预告**

今天的课就到这里，一起加油吧！

下一课我们学习：[Linux探索之旅 | 第二部分第六课：Nano文本编辑器和终端配置](https://www.jianshu.com/p/74e39da071a0)

我是 [谢恩铭](https://www.jianshu.com/u/44339a8a9afa)，在巴黎奋斗的软件工程师。  
我创建了程序员联盟的 [知识星球](https://www.jianshu.com/p/9c3826859dd2)，欢迎加入。  
热爱生活，喜欢游泳，略懂烹饪。  
人生格言：「向着标杆直跑」

[上一篇](https://www.jianshu.com/p/45904376a66f)

[目录](javascript:;)

[下一篇](https://www.jianshu.com/p/74e39da071a0)

**文章还行的话记得点个“❤”，谢谢！**

赞赏支持

[Linux探索之旅](https://www.jianshu.com/nb/5470392)