第二部分　单主机系统的管理

第5章　用户的管理

在Linux操作系统中，每一个文件和程序必须属于某一个“用户”。每一个用户都有一个唯一的身份标识叫做用户ID（User ID，UID）。每一个用户也至少需要属于一个“用户分组”，也就是由系统管理员建立的用户小团体。用户可以归属于多个用户分组。与用户一样，用户分组也有一个唯一的身份标识叫做用户分组ID（Group ID，GID）。

对某个文件或程序的访问是以它的UID和GID为基础的。一个执行中的程序继承了调用它的用户的权利和访问权限（本章后面将要介绍的set -UID命令可以建立改变这个规则的规定）。每位用户的权限可以被定义为下面两种中的任何一种：普通用户或者根用户。普通用户只能访问他们拥有的或者有权限执行的文件；分配给他们这样的权限是因为这个用户或者属于这个文件的用户分组，或者因为这个文件能够被所有的用户访问。根用户能够访问系统全部的文件和程序，而不论根用户是否拥有它们。根用户通常也被称为“超级用户”。

如果你对Windows NT很熟悉，就可以看出该系统的用户管理与Linux操作系统用户管理之间的异同。举例来说，Linux操作系统中的UID就相当于Windows中的SID。与Windows NT相对照，你就会发现Linux操作系统的安全模式其出发点是很简单的：或者你是一个根用户，或者不是一个根用户。NT中的普通用户可以分配到系统管理员（Administrator）的访问权限，但是Linux中的普通用户就没有办法采用相同的机制被授予根用户的优先权。你还将注意到在Linux操作系统中很明显地没有访问权限控制清单（Access Control Lists，ACL）这样的东西。哪一种机制更好呢？这要看你打算干什么和你问的是谁。

本章我们将学习在单主机系统上对用户进行管理的技术。对网络用户进行管理的内容我们将在第19章网络信息服务（NIS）中进行讲解。我们从深入那些保存着用户资料的数据库文件开始。从那里，我们将学习到一些对文件进行自动化管理的系统工具。

5.1　关于用户

在Linux操作系统中，任何东西都有一个所有者。也就是说，Linux系统中没有用户是无法存在的！最少它必须有一个根用户。而大多数的Linux发行版本中都有几种特别设置的用户，这些用户作为自备文档的工具在系统中运行得很好，因为每个用户都拥有全部与其用户名相关联的文件。举例来说，用户www就被设置为这样一个用户，它拥有与World Wide Web（万维网）服务相关联的全部文件。这些用户都按照这样一个方式配置：它们的访问权限只分配给经过挑选的一个很小的用户范围，因此你不必担心它们会被滥用。

用户的账户需要经过一些设置才能正常使用。本小节我们将介绍并讨论它们的作用。设置一个账户的实际过程将在本章稍后进行讲解。

5.1.1 用户登录子目录

每一个实际登录进入系统上机的用户都需要有地方保存那些专属于他的配置文件。这需要让用户工作在自己定制的操作环境中以免改变其他用户定制的操作环境——即使是两个用户同时登录上机也应该如此安排。这个地方就叫做用户登录子目录（ home directory）。在这个子目录中，用户不仅可以保存他的配置文件，还可以保存自己日常工作用到的各种文件。

出于一致性的考虑，大多数站点都从 /home开始安排用户登录子目录，并把每个用户的子目录命名为其上机使用的登录名。这样，如果你的登录名是 hdc，你的用户登录子目录就位于 /home/hdc。例外的情况是那些系统账户，比如每个根用户的账户。系统账户的登录子目录通常在“ / ”或者该账户指定的位置（比如：如果安装 Apache Web服务器，www账户可能就会把自己的登录子目录设置在 /usr/local/apache下）。根用户的登录子目录对大多数 UNIX操作系统的变体来说都是传统的“ / ”，许多Linux操作系统的安装把它设置为 /root。

把用户登录子目录安排在 /home下的决定完全是人为的——但是它确实便于对用户进行组织和管理。系统其实并不关心我们到底把用户登录子目录安排在什么地方，因为每个用户的位置是在口令文件（本章后面将会讲到）中定义说明的。你可能见过某些站点使用的是 /u；或者在/home子目录中分别建立 /home/engineering、/home/accounting、/home/admin等几个部分，然后再把用户安排到各部分中去（比如说，工程方面的 Bosze博士的登录子目录就是 /home/engineering/bosze）。

5.1.2 口令

每个账户都必须有一个口令，否则就根本不可能登录进入它。这对你的系统安全性十分关键——薄弱的口令会降低系统的安全性。

引入口令机制最初的理论是很有意思的，特别是至今我们还相当依赖它的作用。想法很简单：用不着把口令当作秘密保存在重重保护下的文件中，系统可以使用由 NSA（美国家安全局）开发的DES算法对口令进行加密，而加密后的数据可以让公众看到。在当时这么做是安全的，因为这个加密算法从计算的角度说很难被破译。即使到了现在，大多数人能够选择的最好办法还是“ 暴力 ”字典攻击，即让自动攻击系统使用一本大字典进行遍历尝试，这要寄希望于用户选择了某个英文单词作为他们的口令。有许多人想解开 DES算法本身，但是因为它本身就是一个公开的算法，每个人都可以对它进行研究，所以在它公开传播之前就已经被打造得更加刀枪不入了。

当用户在登录提示符处输入它们的口令时，输入的口令将由系统进行加密。再把加密后的数据与机器中用户的口令数据项进行比较。如果这两个加密数据匹配，就可以让这个用户进入系统。实际进行数据加密的算法从计算的角度看很简单，进行一次加密操作不会花费太多的时间。但是，对于一次字典遍历攻击来说，它所需要的成千上万次加密就不可避免地需要大量的时间了。除了加过密的口令之外，口令文件还可以保存用户的登录子目录、 UID、shell、真实姓名等信息；即使某个用户运行的应用程序具备访问这个文件的权限，也不必担心会降低系统的安全性。

但是后来出现了这样一个问题：莫尔定律关于计算机处理器速度每过 18个月就提高一倍的说法不断被事实所证明，家用计算机的运算速度也快得足以让程序在几天而不是几星期或者几年之内实现“ 暴力 ”字典攻击。字典越来越大，而程序越来越智能。口令的作用需要重

新考虑了。

影子（shadow）口令是对此局面的一个解决方案。在影子口令机制中，加过密的口令数据项从口令文件中被转移到另外一个名为 shadow的文件中。原来的口令文件还可以继续被系统上的所有用户访问，但真正加过密的口令数据项则只能够由根用户访问（ login提示符只有根用户权限才能运行）。为什么不规定口令文件只能由根用户访问呢？这没有那么简单。因为这么多年来口令文件一直是开放的，系统软件的其余部分又围绕着它发展，因此需要依赖口令文件总是能够被所有用户访问这个事实。改变这一点就会引起软件工作失常。

另外一个解决方案是改进用来对口令进行加密的算法。有些 Linux的发行版本追随着FreeBSD操作系统，开始使用MD5加密机制。这个算法增加了破译口令的复杂性，当它与影子口令方法联合使用的时候，效果就更好。（当然，还需要假设你让你的用户选用了好的口令！）

窍门 选择好的口令永远是问题的关键。你的用户肯定会问：“噢，全能的系统管理员，怎样才是好的口令？”这就是你的答案：一个非语言单词（不是英语、不是西班牙语、不是德语、不是人类使用语言的单词），最好大小写、数字和标点符号混用——换句话说，一个看起来就像是乱写出来的字符串。好了，这法子够棒。但是如果一个口令太难记忆，大多数人都会把它写下来放在一个容易看见的地方，这就完全违背了前面的初衷。所以最好还是让它比较容易记忆！我喜欢用的办法是：选择一句短语，然后把这句话里每一个单词的第一个字母拼在一起。这样，短语“ coffee is VERY GOOD for you and me ”（咖啡对你我都很有好处）就变成了 ciVGfyam。虽然拼出的口令不好记忆，但是这个短语还行。

5.1.3 shell

当用户登录进入系统时，他会希望有一个能够帮助自己出成果的操作环境。用户遇到的第一个程序叫做 shell。如果你比较熟悉 Windows世界，可以把这个东西等同于 command.com或者“ Program Manager ”（程序管理器）。

在UNIX世界里，大多数 shell都是基于文本的。我们将在第 6章更进一步讨论根用户使用的缺省shell，叫做Bourne Again Shell，简称BASH。Linux操作系统带有好几种shell供用户选用——你可以在/etc/shells文件中看到它们中的大多数。选择哪一种 shell最适合你就像选择最喜欢的啤酒一样——对你来说最好的对别人可不一定——而且几乎每个人都会为自己的选择找理由！

UNIX之所以如此有趣在于用户不必吊在 /etc/shells文件中列出的那些shell上。按照最严格的定义，每个用户的口令数据项中并没有定义需要运行哪个 shell，其中列出的是这个用户上机后第一个运行的程序是哪个。当然，大多数用户愿意第一个运行的程序是一个 shell，比如BASH。

5.1.4 启动上机脚本程序

在DOS下，我们已经习惯了在开机的时候有 autoexec.bat和config.sys文件自动运行。因为DOS是一个单用户系统，这两个程序不仅要完成系统功能如加载设备驱动程序，还要建立我们的工作环境。

而UNIX则是一个多用户环境。每个用户都可以拥有他或她自己的配置文件，这样系统就

像是为每一个用户定制好了似的，即使有其他人同时登录也是如此。配置文件是以 shell的“命令脚本程序”（script）的面目出现的——即当某个用户登录上机时由shell执行的一系列命令。对BASH而言，就是.bashrc文件（是的，在文件名的开始是一个句号——即以句号开始的文件名，它们也叫做“点文件“，在正常的文件清单中是看不见的，除非用户使用了特殊的参数列出它们）。你可以把命令脚本程序想像成批处理文件，只不过命令脚本程序功能更强大。.bashrc命令脚本程序与autoexec.bat文件的性质更相像。

当你建立了一个用户账号的时候，必须提供一套缺省的点文件让这个用户可以开始工作。如果你使用了Linux提供的工具，就不必费劲去建立这些文件，因为许多工具会为你自动完成这个任务。

5.1.5 电子邮件

建立一个新用户不仅仅意味着要建立这个用户的登录子目录和设置其操作环境，还意味着能够让这个用户收发电子邮件。在Linux中建立一个电子邮箱是很简单的，如果这个账户是你使用了Linux操作系统提供的工具建立的，你甚至都不必自己去做这些事！

电子邮箱保存在/var/spool/mail子目录中。每个用户都有一个基于他或她的登录名的电子邮箱。这样，如果某个用户的登录名是 jyom，他的电子邮箱就是/var/spool/mail/jyom。一个空电子邮箱是一个零长度的文件。所有电子邮箱都应该只属于它们对应的主人，其访问权限被设置为不允许别人读出其中的内容（进行这些设置的细节请参考第6章中讲述 chown、chmod和chgrp命令的内容）。

要想在系统中的什么地方建立一个零长度的文件，可以像下面这样使用 touch命令：

```
[ root@ford /root ] # touch myfile
```

上面的命令在当前子目录中建立了一个名为myfile的新文件。

5.2 用户数据库

如果你已经熟悉了Windows NT的用户管理，就肯定已经熟悉了处理用户数据库繁琐细节的“User Manager“（用户管理器）工具。这个工具很方便，但是它使开发你自己的系统管理工具的工作变得很需要技巧，因为读出或者访问用户信息的其他途径只有通过一系列 API调用才能实现。

与此形成鲜明对照的是Linux操作系统采用了UNIX传统的方法，把全部的用户信息保存为普通的文本文件。这样就允许你不必借助于任何其他的工具，只使用一个文本编辑器如pico就可以对用户信息进行修改，十分简便。在许多案例中，大型站点都开发了自己专用的用户管理工具来利用这些文本文件，这样它们不仅可以建立新账户，还可以自动对公司电话簿、Web网页等进行增改。

但是那些第一次采用UNIX方式进行工作的用户和用户分组还是有可能并更愿意停留在Linux发行版本提供的基本的用户管理工具上。我们将在本章的后面讨论那些工具。现在，我们先来看看Linux的文本文件的结构。

5.2.1 /etc/passwd文件

/etc/passwd文件保存着用户的登录名、加过密的口令数据项、用户 ID（UID）、缺省的用

户分组ID（GID）、姓名（有时也叫做GECOS）、用户登录子目录以及登录后使用的shell。这个文件的每一行保存一个用户的资料，而用户资料的每一个数据项采用分号分隔。如下所示：

```
sshah ：boQavhhaCKaXg  ：100 ：102 ：Steven Shah：/home/sshah  ：/bin/tcsh
```

一般说来，用户的登录名不应该超过八个字符。如果你工作在一个超大型的UNIX环境中，更要注意遵守这一点；因为不同的UNIX操作系统在处理长用户名的时候其方法是不一样的。

我们已经在本章的前面讨论过口令数据项的细节。在上面列出的代码中，可以实际看到一个经过DES算法加密的口令是什么样的（第一个分号后面的数据）。许多站点是通过修改这个加过密的口令数据项来禁用某个账户的。这样做了以后，当这个被禁用了账户的用户输入他的口令时，就不会与口令文件中的值相匹配（为了这个目的而改变口令最保险的方法是在这个数据项中插入一个星号*。就上面的这个例子，那个数据项可以被修改为boQavhhaCKaXg*）。

窍门 采用这种方法禁用账户的时候，除了加上一个星号字符之外，如果再加上一个说明为什么禁用这个账户原因的字符串就更有帮助了。举例来说，如果你发现某个用户正在下载盗版软件，就可以禁用掉他的账户，并把他的加密口令数据项修改为boQavhhaCKaXg*caught pirating。

用户ID（UID）对每一个用户来说都必须是唯一的，只有UID等于0时可以例外。任何拥有0值UID的用户都具有根用户（系统管理员）访问权限，因此具备对系统的完全控制。通常，UID是这个特殊值的用户的登录名是“root”。允许任何其他用户或者用户名拥有0值UID都是不好的做法。这就明显区别于Windows NT中的管理模式：在Windows NT中，可以有任意数量的用户具有系统管理员级别的优先权。

注意 有些Linux操作系统的发行版本保留数值-1（或者65535）作为“nobody”用户的UID。

用户姓名可以是任何格式的文本数据项。虽然不可打印字符可以用在这个字符串中，但是这样做被认为是很不好的。另外，用户姓名也不允许延伸到多个文本行。

注意 虽然用户完整的口令数据项不可以延伸到多个文本行，但是它却可以多于80个字符。

用户的登录子目录数据部分请参考本章前面所讨论过的内容。最后一个数据项，用户登录后使用的shell，也是如此。那么，一个系统完整的口令文件可能会是下面的这个样子：

```
root:AgQ/IJgASeW1M:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/adm:
lp:*:4:7:lp:/var/spool/lpd:
sync:*:5:0:sync:/sbin:/bin/sync
shutdown:*:6:0:shutdown:/sbin:/sbin/shutdown
halt:*:7:0:halt:/sbin:/sbin/halt
mail:*:8:12:mail:/var/spool/mail:
news:*:9:13:news:/var/spool/news:
uucp:*:10:14:uucp:/var/spool/uucp:
operator:*:11:0:operator:/root:
```

```
games:*:12:100:games:/usr/games:
gopher:*:13:30:gopher:/usr/lib/gopher-data:
ftp:*:14:50:FTP User:/home/ftp:
pop:*:15:15:APOP Admin:/tmp:/bin/tcsh
nobody:*:99:99:Nobody:/:
sshah:Kss9Ere9b1Ejs:500:500:Steve Shah:/home/sshah:/bin/tcsh
hdc:bfCAblvZBIbFM:501:501:H. D. Core:/home/hdc:/bin/bash
jyom:*:502:502:Mr. Yom:/home/jyom:/bin/bash
```

5.2.2 /etc/shadow文件

家用电脑的速度开始让黑客们能够比较任意地对口令文件实现字典攻击，这就导致了把加过密的口令从/etc/passwd文件分离出去的做法。/etc/passwd依然保持对全部用户都可读，但是保存在/etc/shadow文件中的口令则只对那些具有根用户优先权的程序如登录程序等可读。

除了加过密的口令，/etc/shadow文件中还包含着口令失效期和账户是否已被禁用等方面的信息。/etc/shadow文件中每一行的格式包含着如下所示的几个部分：

- 登录名。
- 加过密的口令。
- 从1970年1月1日起计算，该口令修改后已经过去了多少天。
- 需要再过多少天才能修改这个口令。
- 需要再过多少天这个口令必须被修改。
- 需要在这个口令失效之前多少天对用户发出提示警告。
- 口令失效多少天之后禁用这个账户。
- 从1970年1月1日起计算，该口令已经被禁用了多少天。
- 保留域。

每个用户的数据占用一行，彼此用冒号隔开。如下面的例子：

```
sshah ：boQavhhaCKaXg  ：10750 ：0：99999 ：7：-1：-1：134529868
```

数值为-1的数据项其含义是无限。如果在表示需要再过多少天这个口令必须被修改的那个数据项位置处的数值是-1，就等于明确地表示那个用户永远都不必修改他的口令。

5.2.3 /etc/group文件

正如你所知道的，每个用户至少会属于一个用户分组，也就是他缺省的用户分组。在需要的情况下，用户还可以分配到其他的分组中去。/etc/passwd文件中包含着每个用户缺省的分组ID（GID）。在/etc/group文件中，这个GID被映射到该用户分组的名称以及同一分组中的其他成员去。/etc/group文件中每一行的格式如下所示：

- 用户分组名。
- 加过密的用户分组口令。
- 用户分组ID号（GID）。
- 以逗号分隔的成员用户清单。

每一个数据域还是以冒号隔开的，其中的数据项看起来应该如下所示：

```
project ：baHrE1KPNjrPE  ：102 ：sshah, hdc
```

与/etc/passwd文件一样，分组定义文件对整个系统来说也必须是可读的，因为这样应用程序才能测试用户与用户分组之间的关联性。用户分组的名称不允许超过 8个字符，每个分组的GID也必须是唯一的。最后，以逗号分隔的成员用户清单中列出的只是那些不属于这个特定分组的用户。

如果想建立一个没有口令的用户分组，可以使用下面的方法设置数据项：

```
project ：baHrE1KPNjrPE ： 102 ：sshah, hdc
```

如果打算建立一个用户分组，并且不希望允许任何人把他的工作分组修改为这个分组的话（适用于这样的情况：应用程序需要其自己的分组，但是不存在合法的理由让一个用户工作在该分组中），可以在口令数据域中放上一个星号。如下所示：

```
project ：*：102
```

5.3 用户管理工具

拥有口令数据库文件最奇妙的优点是任何人都可以编写出他们自己的管理工具程序。事实上，许多站点的系统管理员已经这样做了，目的是把他们的工具程序与其组织内部结构的其他部分整合到一起。他们可以从一个表格中建立一个新的用户，并且能够对公司电话与电子邮件地址、LDAP服务器、Web网页等方面进行修改更新。当然，并不是所有的人都想“有自己的车开”——因此Linux操作系统附带了几个已经编写好的工具程序为你干那些活。

我们将在本小节中对来自命令行界面和图形化用户界面的两类用户管理工具程序进行讨论。学会使用两种方法当然是最好的了，因为你不可能知道自己会在哪一天在什么样的环境中添加用户。

5.3.1 使用命令行进行用户管理

可以从下列的六种命令行工具程序中进行选择，用来执行 GUI工具程序完成同样的动作：useradd、userdel、usermod、groupadd、groupdel和groupmod。使用GUI工具程序明显的优点是使用简便。但它的缺点是用它执行的动作无法自动地进行。在这种情况下，命令行工具程序就相当简便了。

请注意：非Red Hat的Linux发行版本与这里使用的工具程序的参数可能略有不同。如果想了解你特定安装到底与之有什么不同，请阅读相关程序的使用手册页。

1. useradd命令

正如其名称的含义， useradd命令允许你一次把一个用户添加到系统中去。与 GUI工具程序不同，它没有任何交互性的提示。反之，全部参数都必须在命令行上定义好。下面是这个工具程序的使用方法：

```
useradd [-c comment] [-d homedir] [-e expire date] [-f inactive time]
[-g initial group][-G group[,...]] [-m [-k skeleton dir]] [-M]
[-s shell] [-u uid [-o]] [-n] [-r] login
```

不要被这么长的参数表吓到！我们将依次介绍它们中的每一个，并讨论它们之间的联系。

在扎进这些参数之前，请注意方括号中的所有内容都是可选的。这样，我们可以发出如下所示的一个最简单的命令

```
[ root@ford /root ] # useradd sshah
```

来添加一个登录名为sshah的用户。任何没有定义的参数都使用了缺省值（要想查看这些参数的缺省值，请执行useradd -D命令；马上我们就将讨论到如何改变这些缺省值）。表5-1列出了这个命令的参数和对它们的说明。

表5-1 useradd命令的参数及其说明

参 数	说 明
-c comment	允许你在GECOS域中设置用户的姓名。与其他命令行参数一样，如果它的设置值中有空格，就必须在文本两端加上引号 —— 比如说，如果想把用户的姓名设置为Steven Shah，就必须把参数定义为-c “ Steven Shah ”
-d homedir	缺省的情况下，用户的登录子目录被定义为 /home/login。这样如果我的登录名是sshah，我的登录子目录就是/home/shah。在建立一个新用户的时候，该用户的登录子目录是和用户账户一起建立的。因此，如果你想把缺省值修改为另外一个位置，就可以定义新的位置 —— 比如说，-d /home/sysadm/sshah
-e expire-date	在经过了一段时期之后，账户有可能失效。缺省的情况下，账户永远都不会失效。如果你想指定一个日期，一定要按照 MM/DD/YY的格式进行（本系统中，对 2000年请使用00做为设置值） —— 使用-e 04/01/00设置为在2000年4月1日失效
-f inactive-time	定义口令失效后，新账户还能够使用的天数。 0（零）值表示要立刻禁用这个账户。-1值表示即使在口令失效之后也不禁用这个帐户 —— 比如说：-f 3允许每个用户在口令失效还可以再使用3天。这个参数的缺省值是-1
-g initial group	使用这个参数可以在口令文件中定义用户缺省的分组。你可以使用那个分组的一个编号或者名称；但如果你使用的是每个用户分组的名称，那么这个用户分组必须已经在/etc/group进行了定义 —— 比如-g project
-G group [, . . .]	允许你把新用户设置到其他的分组中去。如果你使用了 -G参数，就必须至少指定一个额外的用户分组。另外，你还可以使用逗号分隔定义多个用户分组（举例来说，如果你打算把用户加到 project和admin用户分组中去，就可以使用 -G projetc,admin）
-m [-k skel-dir]	缺省情况下，系统将自动地建立用户的登录子目录。这个参数明确地建立用户的登录子目录名称。建立这个子目录的部分工作是把缺省的配置文件拷贝到这个子目录中去。这些文件缺省情况下是从 /etc/skel子目录中拷贝过去的。使用第 2个参数-k skel dir可以改变缺省设置（你必须使用了 -m参数才能使用-k参数）。举例来说，如果想指定/etc/adminskel子目录，我们需要使用-m -k /etc/adminskel
-M	如果已经使用了-m参数，就不能再使用 -M参数，反之亦然。这个参数告诉系统不要建立用户的登录子目录
-n	Red Hat Linux把建立一个与新用户同名的新用户分组作为用户添加过程的一部分。使用这个参数可以禁止这种行为
-s shell	用户的登录shell是一个用户登录进入一个系统之后运行的第一个程序。它通常是一个命令行操作环境，除非你是从 X-Windows登录屏幕登录的。缺省情况下，它将是Bourne Shell（/bin/bash）。有些人喜欢使用其他的shell如Turbo C Shell（/bin/tcsh）。这个参数可以让你随意选择新用户登录之后运行的 shell（shell的清单保存在/etc/shells文件中）
-u uid	缺省的情况下，程序将会自动地找出下一个可用的 UID并使用它。如果出于某种原因你需要强制让某个新用户的 UID是一个特殊的数值，则可以使用这个参数。请记住对全部用户来说，他们各自的 UID必须是唯一的
Login	最后，唯一 “ 不是 ” 可选项的参数！你必须指定新用户的登录名

举例来说，如果打算建立这样一个用户：他的姓名是 H. D. Core，同时属于admin和support用户分组（缺省的用户分组是admin）、喜欢使用Turbo C Shell并希望使用登录名“ hdc ”，请使用下面这样的命令：

```
[ root@fordd /root ] # useradd -c " H. D. core " -g admin -G sopport -s
/bin/tcsh hdc
```

2. userdel命令

userdel命令实现的操作正好与useradd命令相反 —— 它删除现有的用户。这个命令只有一个可选参数和一个必要参数：

```
userdel [ -r ] username
```

在执行这个命令的时候，如果只指定了用户的登录名 —— 比如说userdel sshah，那么在/etc/passwd文件和/etc/shadow文件中的有关数据项以及 /etc/group文件中的关联数据项都将被自动删除。如果使用了可选参数 —— 比如说userdel -r sshah，那么在其登录子目录中归这个用户所有的全部文件也将被删除。

3. usermod命令

usermod命令允许你修改系统中现有的某个用户，它的工作原理与 useradd命令差不多。完整的命令行使用方法如下所示：

```
usermod [-c comment] [-d homedir] [-m] [-e expire date]
[-f inactive time] [-g initial group]
[-G group[,...]] [-l login] [-s shell]
[-u uid] login
```

使用这个命令的时候指定的每一个参数都将改变这个用户的某个属性。这个命令的参数除了-l之外，其余的都与useradd命令中对应的参数作用相同。

-l参数允许你改变用户的登录名，它和 -u参数在使用中必须引起足够的重视。在修改用户的登录名或者UID的时候，必须确认该用户当时没有登录上机或者运行任何进程。如果在用户已经登录上机或者正在运行进程的时候修改这些信息会引起不可预见的结果。

下面是一个使用usermod命令对用户hdc进行修改的例子，我们打算把她的姓名域由“ H. D. H ”修改为“ H. D. Core ”。

```
[ root@ford/root ] # usermod -c " H. D. Core " hdc
```

4. groupadd命令

对用户分组进行操作的命令类似于对用户进行操作的命令；但是，它们并不作用于单个的用户，而是作用于/etc/group文件中列出的用户分组。请注意改变用户分组的属性并不会自动改变用户的属性。举例来说，如果你删除了一个 UID为100的用户分组，而某个用户缺省的分组正好是100，那么这个缺省的用户分组数据项不会改变，反映不出这个用户分组已经被删除的事实。

groupadd命令把用户分组添加到/etc/group文件中。它的命令行操作格式如下所示：

```
groupadd [ -g gid ] [ -r ] [ -f ] group
```

表5-2列出了命令参数和有关的说明。

举例来说，假设你打算添加一个GID为800的新分组research，可以输入下面的命令：

```
[ root@ford /root ] # groupadd -g 800 research
```

表5-2 groupadd命令的命令行参数和它们的说明

参 数	说 明
-g *gid*	把新用户分组的GID指定为gid。缺省的情况下，这个值被自动选定为找到的第一个可用值
-r	缺省情况下，Red Hat会自动搜索第一个大于499的GID值。-r参数告诉groupadd命令正在添加的用户分组是一个系统分组，需要使用第一个小于499的可用数值
-f	在添加新用户分组的时候，如果准备添加的用户分组已经存在，Red Hat Linux会自动退出执行，并且没有错误信息。使用这个参数，在退出执行之前，这个命令不会修改用户分组的设置值。在进行命令脚本程序编程的时候，如果你打算在用户分组已经存在时让命令脚本程序继续执行下去，这个参数就非常有用了
group	这个参数是必需的。它定义了你打算添加的用户分组名称为 group

5. groupdel命令

比userdel命令还直接，groupdel命令删除在/etc/group文件中定义的用户分组。这个命令唯一的使用方法是：

```
groupdel group
```

其中的group是准备删除的用户分组的名称。举例来说，如果打算删除 research分组，需要发出下面的命令：

```
[ root@ford /root ] # groupdel research
```

6. groupmod命令

groupmod命令修改某个现有用户分组的属性。这个命令的参数如下所示：

```
groupmod -g gid -n group-name group
```

其中的-g参数允许改变用户分组的GID值，-n参数允许你给用户分组起一个新名字，当然还需要把现有用户分组的名称作为最后一个参数。

举例来说，如果用户分组 superman打算把它的名称修改为 batman，需要使用下面的命令来进行设置：

```
[ root@ford /root ] # groupmod -n batman superman
```

5.3.2 使用LinuxConf进行用户管理

LinuxConf工具软件包是一个功能非常强大的配置工具，可以用来执行许多不同的任务。它的特色之一就是建立、删除和修改用户信息的能力。

启动LinuxConf之前，必须以根用户身份登录进入系统并运行 X-Windows环境。如果你正在使用的是GNOME和Enlightenment（Red Hat的缺省设置），应该在左下角的菜单中看到LinuxConf菜单项。如果它没有出现在那里（或者你并没有使用缺省的窗口管理器），可以从某个终端窗口中输入linuxconf命令启动LinuxConf。

启动LinuxConf之后，就可以在下拉菜单中看到用户管理部分，如图 5-1所示。

从这个窗口中，可以执行三个基本的操作：添加一个新用户、修改一个现有用户以及删除一个用户。

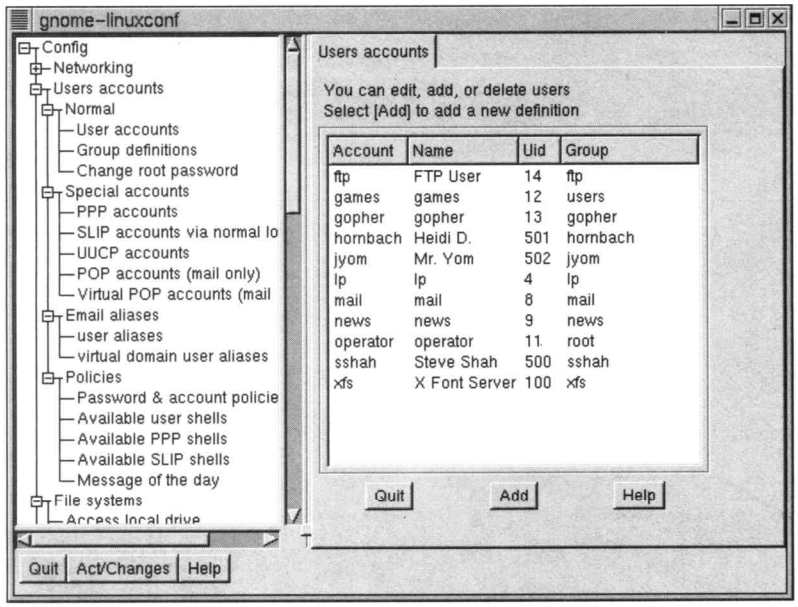

图5-1 LinuxConf窗口

1. 添加一个用户

如果想添加一个用户，请先单击窗口底部的“ Add ”（添加）按钮，窗口如图5-2所示。

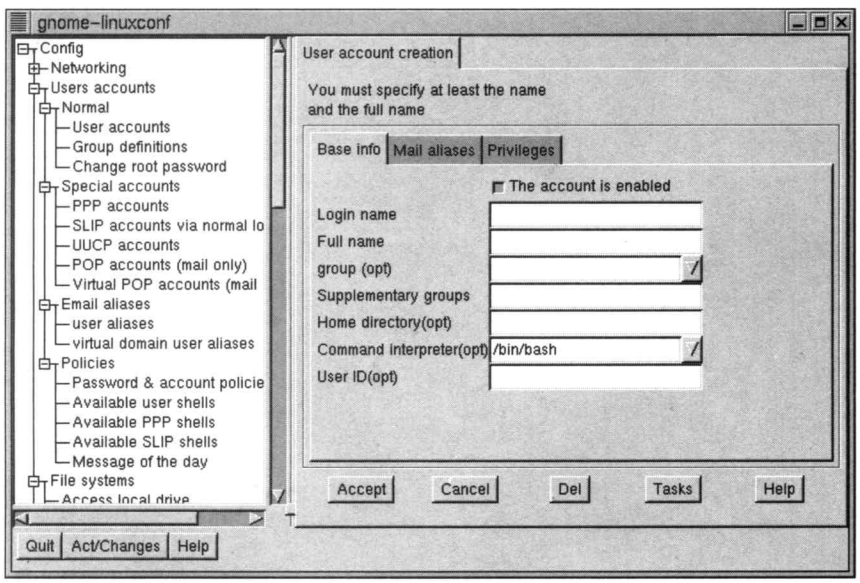

图5-2 使用LinuxConf添加一个用户

除了出现“ (opt) ”（可选项）字样的数据域之外，其他的每一个数据域都必须填写。其中包括新账户的登录名和用户的真实姓名，所有其他的数据域都有相应的缺省值。这些参数与useradd程序中的各个数据项是一一对应的。

除了这些基本的用户属性之外，LinuxConf还可以设置一些参数，它们是用户对某些应用程序的访问权限和电子邮件设置（如果用户需要把自己的电子邮件转发到其他地址的话）。这些选

项可以通过窗口中的标签进行设置，图5-3给出了“ Privileges ”（优先权）设置标签的画面。

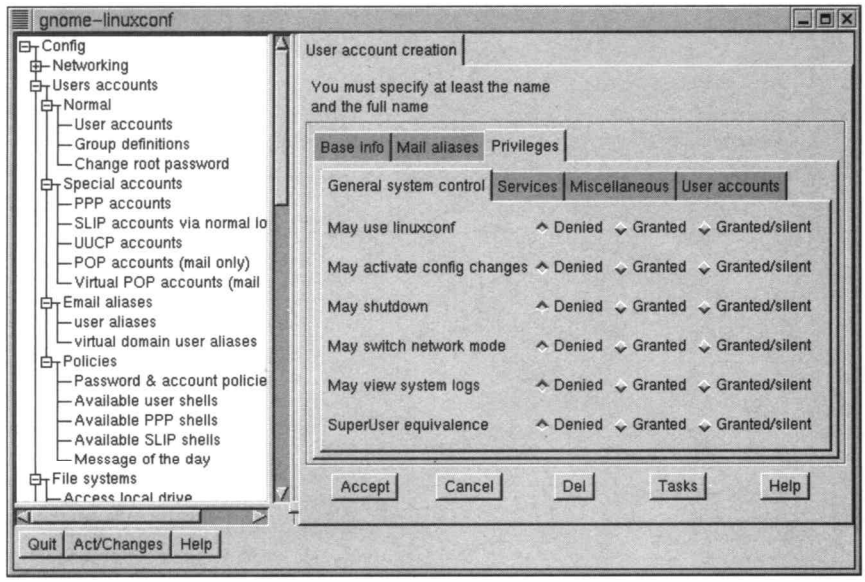

图5-3 使用LinuxConf添加一个用户时的“ Privileges ”（优先权）标签

根据需要选好适当的参数之后，单击窗口底部的“ Accept ”（接受）按钮。下一个窗口让你设置用户的初始口令。口令设置完成后，就返回到现有用户清单的画面，新添加的用户也出现在那里了。

2. 修改一个用户

修改一个用户的设置值相当简单。在LinuxConf的开始窗口中选中准备要修改设置值的用户。屏幕上将出现类似于图5-4所示的窗口，它类似于添加用户时的窗口，但是已经是完全填写好了的。窗口中的所有数据域都可以进行修改，全部修改完成之后，单击“ Accept ”（接受）按钮。

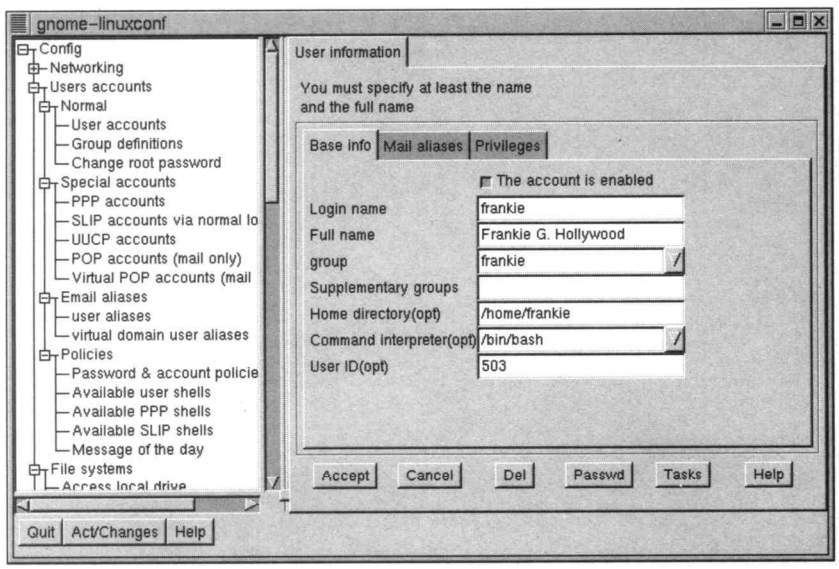

图5-4 使用LinuxConf修改用户信息

注意 在修改某个用户的登录名或者UID之前，一定要确认那个用户当时没有登录上机或者运行任何进程。

3. 删除一个用户

如果想删除一个用户，在图5-1所示画面的用户清单中单击打算删除的用户名。屏幕上出现一个用户信息窗口，窗口底部有一个“ Del ”（删除）按钮。单击这个按钮，就开始对这个用户进行删除操作（如图5-5所示）。

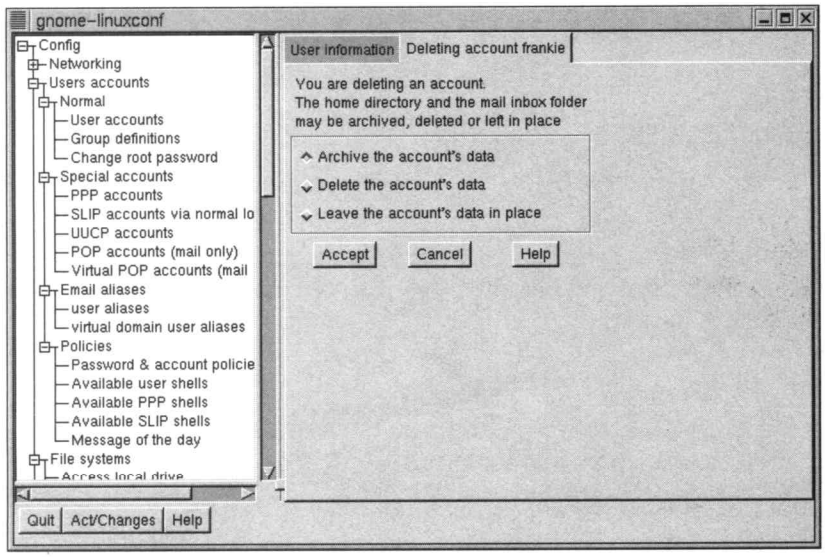

图5-5 使用LinuxConf删除用户账号

决定了如何处理该用户的登录子目录之后，单击“ Accept ”（接受）按钮回到用户清单画面。如果选择了对该用户的登录子目录进行归档，就将在 /home/oldaccounts子目录中找到一个包含着其数据的压缩归档文件。

4. 添加一个用户分组

如果想添加一个用户分组，请单击LinuxConf开始窗口左边的“ Group Definition ”（用户分组定义）菜单项。在屏幕窗口的右边将出现一个当前用户分组的清单（如图 5-6所示）。

如果单击“ Add ”（添加）按钮，将看到一个窗口，在其中填上新建用户分组的名字。虽然窗口中显示有好几个数据域，必须要填写的只有新建用户分组的名字，其余的数据域都有由系统提供的缺省设置值。完成输入之后，单击“ Accept ”（接受）按钮返回到用户分组清单的画面，新添加的用户分组也出现在那里了。

5. 修改用户分组

与对一个用户进行修改的操作类似，如果打算对某个用户分组进行修改，请在用户分组清单窗口中选中那个分组的名字，打开一个类似用户分组添加的窗口。在窗口中将各数据项都填写为该分组当前的设置值。修改相应的数据项，然后单击“ Accept ”（接受）按钮让所做的修改生效。

6. 删除一个用户分组

如果想删除一个用户分组，在图5-6所示的用户分组清单中选中打算删除的用户分组名称。

屏幕将显示出这个用户分组的信息资料，在其底部有一个标识为“ Del ”（删除）的按钮。单击“ Del ”（删除）按钮，打开一个窗口，要求对操作进行确认。单击“ Yes ”（确认）按钮，返回到用户分组清单画面，可以看到所删除的用户分组已经不在其中了。

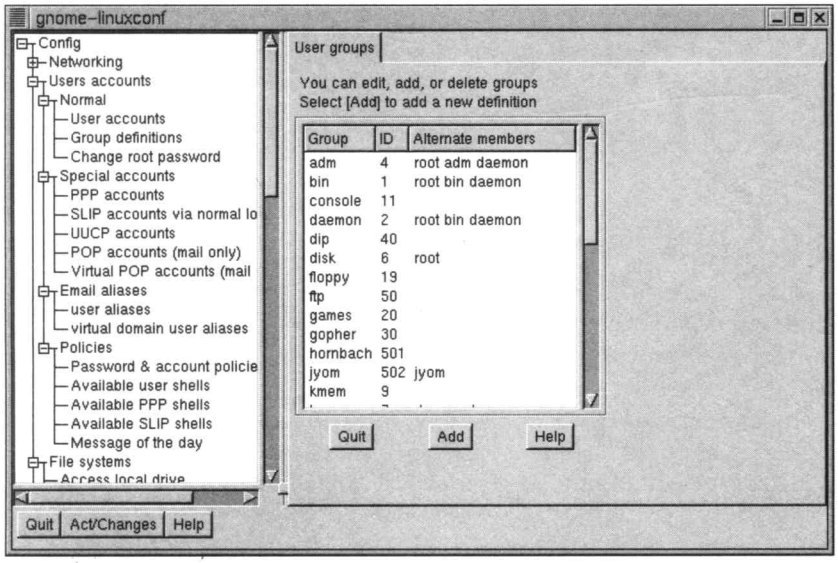

图5-6 使用LinuxConf添加一个用户分组

5.4 SetUID和SetGID程序

一般情况下，当用户运行一个应用程序的时候，这个程序将继承该用户所具有的全部权利（或者限制）。如果用户不能够读取/var/log/messages文件，那么他运行的程序也不能读取该文件。请注意这个权限可能会与该程序文件（通常叫做二进制文件）所有者所具有的权限有所不同。举例来说，ls程序（它用来产生子目录中的文件清单列表）是归根用户所有的，它的访问权限被设置为每一个用户都能够执行。又例如，某用户 sshah运行了ls命令，限制这份ls命令的是分配给用户sshah的访问权限而不是根用户。

但是这里有一个例外。可以采用对一个叫做 SetUID的二进制位进行设置的方法使程序按照其所有者的访问权限运行，不再受运行它的用户的访问权限的限制。我们还使用 ls命令作为例子，如果把它的SetUID位设置为on，并且让这个命令的二进制文件归属于根用户，那么就意味着如果用户sshah运行了ls命令，这份ls命令就是以根用户的访问权限运行的，不再受到用户sshah访问权限的限制。 SetGID位的作用也是如此，只不过它并不作用于文件的所有者而是作用于文件用户分组的设置情况。

如果想激活SetUID或者SetGID位，需要使用chmod命令，我们将在第6章中详细介绍这个命令。如果想把某个程序设置为 SetUID状态，在打算分配给它的访问权限数值前面加上一个数字4。如果想把某个程序设置为SetGID状态，在打算分配给它的访问权限数值前面加上一个数字2。举例来说，如果想把/bin/ls设置成SetUID置位了的程序（随便说一句，这并不是好主意），需要输入下面的命令：

```
[ root@ford /root ] # chmod 4755 /bin/ls
```

5.5 如果没有文件的所有权

本小节的标题有一些误导。不管怎么说，一个文件总会归属于某位用户。一个更加精确的标题应该是“当某个文件所有者的 UID与/etc/passwd文件中的某个数据项不匹配的时候”——但是我的编辑不让这么写。

当一个用户被建立之后，它就得到了一个新的、唯一的 UID值。该用户建立的任何文件都归这个用户所有。出于简单化的考虑， Linux并没有使用用户名而是使用了 UID来设置文件的所有权。然后系统使用/etc/passwd文件在用户的UID和登录名之间做一个映射，这样就使子目录列表操作更容易阅读。

那么当用户从/etc/passwd文件中被删除后但是属于他的文件还依然存在的时候会发生什么样的事情呢？实际上不会发生什么事情。我们最能够观察到的现象将发生在问题中的文件进行子目录列表操作的时候。列表中不会出现文件的所有者，它将显示为一个号码。这个号码代表着拥有该文件的UID。如果有一个新用户在被建立的时候使用了与老用户相同的 UID，这个相同的UID将被显示为所有者，使得新用户看起来就像是拥有着那些文件一样。因为会出现这种情况，所以当删除某个用户的账号时，确保同时删除了该用户拥有的全部文件是十分重要的。

5.6 小结

在本章中介绍了Linux操作系统中用户的含义。你在这里阅读到的东西对其他 UNIX操作系统的变体也是同样适用的，这样就使得在不同的 UNIX操作系统中对大量用户进行管理要比在NT/UNIX系统中要容易得多。

我们讨论过的重要内容主要有以下几个方面：

- 每一个用户都将获得唯一的UID。
- 每一个用户分组都将获得唯一的GID。
- /etc/passwd文件把UID映射到用户名。
- Linux操作系统对加密口令有多种处理方法。
- Linux操作系统附带有帮助你对用户进行管理的工具软件。
- 如果你准备用自己的工具软件对用户数据库进行管理，现在就已经掌握了这么做的格式。

对一个来自Windows 95/98/NT环境的系统管理员来说，这些改变是相当巨大的。并且在刚开始的时候会有一些不适应。但是不必担心， UNIX的安全模式相当简明易懂，这样你将会迅速掌握那些操作的原理。

如果你对自行编写用户管理工具软件感兴趣，就一定要阅读一些介绍 Perl命令脚本编程语言的书籍。这个语言对表格类型的数据（比如 /etc/passwd文件）进行操作处理是极为适合的。因为它所具备的网络功能和对 NT的支持， Linux甚至可以让你编写出跨平台的用户管理工具，使你不仅可以建立和设置UNIX账号，还可以建立和设置NT账号。因为有太多关于Perl语言的书籍，每一种的角度都稍有不同，对编程背景水平的考虑也稍有不同，所以很难具体推荐哪一本书。最好是花一些时间在当地的书店里好好地浏览一下。