

第 8 章用户和基本的帐户管理

(翻译中出现的任何问题或错误，请广大读者及时反馈给我：freebsdhandbook@163.com)

8.1 概要

FreeBSD 允许多个用户同时使用电脑。当然，这些用户中不是很多人同时坐在同一台电脑前，而是其他用户可以通过网络来使用同一台电脑以完成他们的工作。要使用系统，每一个人都要有一个帐户。

读完这章，你将了解到：

- 在一个 FreeBSD 系统上不同用户帐户之间的区别。
- 如何添加用户帐户。
- 如何删除用户帐户。
- 如何改变帐户细节，如用户的全名，或首选的 shell。
- 如何在每个帐户基础上设置限制，来控制象内存，CPU 时钟这样的资源。
- 如何使用组来使帐户管理更容易。

在阅读这章之前，你应当了解：

- 了解 UNIX 和 FreeBSD 的基础知识。

8.2 介绍

所有访问系统的用户必须通过帐户才能完成，所以在 FreeBSD 系统中用户和用户帐户的管理是非常重要的。有三种类型的帐户：超级用户、系统用户和普通用户。超级用户帐户通常叫做 root，可以毫无限制地管理系统。系统用户运行服务。最后，普通用户帐户给那些登陆系统，阅读邮件等的人们使用。

用户名

在登陆时需要键入用户名。用户名对于一台电脑来讲是唯一的；你不可以使用两个相同的用户名来登陆。有很多规则可以用来创建用户名，可以看看 `passwd` 的文档；你使用的用户名通常需要 8 个或更少的小写字母。

口令

每个帐户都有一个口令与它对应。口令可以是空的，这样不需要密码就可以访问系统。这通常不是一个好主意；每个帐户都应当要有一个密码。

用户 ID (UID)

UID 是系统用来识别用户的 0 到 65536 之间的数字。在它工作以前，允许你指定一个用户名的任何命令都会把它转换成 UID。这意味着你可以为不同的用户名使用多个帐户，但它们的 UID 是一样的。FreeBSD 会把这些帐户认定是一个人的。

组 ID (GID)

GID 是用来识别用户所在的组的 0 到 65536 之间的数字。组是一种用来控制用户访问资源的机制。它可以减少一些配置文件的大小。一个用户也可以属于多个组。

登陆类

登陆类是一个组机制的扩展，当把系统分配给用户时，它提供了额外的灵活性。

口令的定期转换

默认情况下，FreeBSD 不会强迫用户去改变他们的密码。你可以在每个用户的基础上强迫执行，当一个帐户过期了，可以强迫一些或所有的用户改变他们的密码。

帐户到期了

默认情况下，FreeBSD 不会终止帐户。如果你正在创建帐户，你要知道有一个有限的使用期限。例如，在学校里，你会为每个学生设立一个帐号，当帐号到期了，你可以重新指定它。帐户到期后，虽然帐户的目录和文件仍然存在，帐户就不能再使用了。

用户的全名

用户名可以唯一地识别 FreeBSD 的帐户，但不会反映用户的全名。这些信息可能与帐户是相关的。

主目录

主目录是用户用来启动的目录的完全路径。一个通常的规则是把所有用户的主目录都放在 `/home/username` 下。用户将会把他们的个人文件放在他们的主目录下，他们可以在那儿创建任何目录。

用户 shell

Shell 提供了用户用来操作系统的默认环境。有很多不同的 shell，有经验的用户会根据他们的经验来选择。

8.3 超级用户帐户

超级用户帐户通常叫做 `root`，可以重新配置和管理系统，在收发邮件、系统检查或编程时尽量不要使用 `root` 权限。

这是因为不象普通用户帐户，超级用户能够毫无限制地操作系统，超级用户帐户的滥用可能会引起无法想象的灾难。普通的用户帐户不会由于出错而破坏系统。所以通常要尽可能地使用普通帐户，除非你需要额外的特权。

另外，在使用超级帐户时要再三检查命令，因为一个额外的空格或缺少某个字符的命令都可能会引起数据丢失。当你要改为超级用户的时候，你的普通用户的安全措施将不起作用。所以，你在阅读了这章后要做的第一件事是在平时使用的时候，创建一个没有特权的用户帐户。无论你使用的是多用户还是单用户的机器，这样的申请都是相同的。在这一章的后面，我们将讨论如何创建一个额外的帐户和如何在普通用户和超级用户之间进行切换。

8.4 系统用户

系统用户可能要使用诸如 DNS，mail,web 等的服务。使用帐户的原因就是为了安全。如果所有的服务都由超级用户来运行，那他们就可以不受约束地做任何事情。系统用户可以是后台程序、操作员、`bind` 或新闻。系统管理员经常创建 `httpd` 来运行 web 服务器。

8.5 用户帐户

用户帐户是让真实用户访问系统的主要方式。这些帐户把用户和环境分隔开，能阻止用户破坏系统或其它用户，在不影响其它用户的情况下定置它们的环境。

每个人访问你的系统必须要有它们自己的唯一帐户。这允许你可以找到谁做了些什么，阻止人们破坏其它用户的设置和阅读其它人的邮件等等。

每个用户能够设置它们自己的环境，以利于它们通过改变 shell、编辑器、键盘绑定和语言等适应使用这个系统。

8.6 修改帐户

`pw` 是一个功能强大，灵活修改帐户的方式，但还是建议用 `adduser` 来创建新的帐户，使用 `rmuser` 来删除帐户。`chpass` 允许系统管理员和普通用户来调整口令，shell 和个人信息。`Passwd` 是个更普通的改变口令的方式。

8.6.1 增加用户

`adduser` 是个简单的增加新用户的命令。它为用户创建 `passwd` 和 `group`，也创建它们的主目录，从 `/usr/share/skel` 拷贝文件形成一个默认的 `.files` 的文件，另外给用户发送一个带欢迎信息的邮件。

创建一个初始的配置文件，使用 `adduser -s -config_create`。下面，我们配置 `adduser` 的默认设置，创建我们第一个用户帐户。

例 8-1，改变 `adduser` 的配置

```
# adduser -v

Use option “ -silent ” if you don ’ t want to see all warnings and questions.

Check /etc/shells

Check /etc/master.passwd

Check /etc/group

Enter your default shell: csh date no sh tcsh [sh]: tcsh

Your default shell is: tcsh -> /usr/local/bin/tcsh

Enter your default HOME 分区: [/home]:

Copy dotfiles from: /usr/share/skel no [/usr/share/skel]:

Send message from file: /etc/adduser.message no

[/etc/adduser.message]: no

Do not send message

Use passwords (y/n) [y]: y

Write your changes to /etc/adduser.conf? (y/n) [n]: y

Ok, let ’ s go.
```

Don't worry about mistakes. I will give you the chance later to correct any input.

Enter username [a-z0-9_-]: **jru**

Enter full name []: **J. Random User**

Enter shell csh date no sh tcsh [tcsh]:

Enter home directory (full path) [/home/jru]:

Uid [1001]:

Enter login class: default []:

Login group jru [jru]:

Login group is "jru". Invite jru into other groups: guest no

[no]: **wheel**

Enter password []:

Enter password again []:

Name: jru

Password: ****

Fullname: J. Random User

Uid: 1007

Gid: 1007 (jru)

Class:

Groups: jru wheel

HOME: /home/jru

Shell: /usr/local/bin/tcsh

OK? (y/n) [y]: **y**

Added user "jru"

Copy files from /usr/share/skel to /home/jru

Add another user? (y/n) [y]: **n**

Goodbye!

#

总的来讲，我们把默认的 shell 设置成 tcsh，关闭欢迎邮件。然后，保存配置，接着创建一个 jru 的帐户，并且确信 jru 在 wheel 组里面。

注意：你输入的口令是不会显示出来的，而只会显示星号。确保输入两次密码时，不

要输错。从现在起，只要使用 `adduser`，你不必改变默认设置。如果程序要求你改变默认设置，先退出程序，然后执行程序时加上 `-s` 选项。

8.6.2 `rmuser`

`rmuser` 能从系统中删除用户，包括超越用户数据库的任何线索。`rmuser` 执行下面的步骤：

- 1, 删除用户的 `crontab` 记录
- 2, 删除属于用户的 `at` 工作
- 3, 杀掉所有属于用户的所有线程
- 4, 删除本地密码文件中的用户
- 5, 删除用户的主目录
- 6, 删除来自 `/var/mail` 的属于用户的邮件
- 7, 删除所有诸如 `/tmp` 的临时文件存储区中的文件
- 8, 最后，删除在 `/etc/group` 中所有属于组的用户名

注意：如果一个组变成空，而组名和用户名一样，组将被删除；`rmuser` 不能用来删除超级用户的帐户。

例 8-2. `rmuser` interactive account removal

```
# rmuser jru

Matching password entry:

jru:*:1000:1000::0:0:J. Random User:/home/jru:/usr/local/bin/tcsh

Is this the entry you wish to remove? y

Remove user 's home directory (/home/jru)? y

Updating password file, updating databases, done.

Updating group file: trusted (removing group jru—personal group is empty) done.

Removing user 's incoming mail file /var/mail/jru: done.

Removing files belonging to jru from /tmp: done.

Removing files belonging to jru from /var/tmp: done.

Removing files belonging to jru from /var/tmp/vi.recover: done.

#
```

8.6.3 pw

Pw 是一个用来创建，删除，修改，显示用户和组的命令行工具，它还有系统用户和组文件编辑器的功能。Pw 有一个非常强大的命令行设置选项，但新用户可能会觉得它比这儿讲的其它命令要复杂得多。

8.6.4 chpass

Chpass 可以改变用户的密码，shells，和个人信息的数据库信息。只有超级用户才能改变其它用户的信息。除了可选择的用户名，不需要任何选项，chpass 显示一个包含用户信息的编辑器，而且可以试图改变在用户数据库中的信息。

例如 8-3. Interactive chpass by Superuser

```
#Changing user database information for jru.
```

```
Login: jru
```

```
Password: *
```

```
Uid [#]: 1000
```

```
Gid [# or name]: 1000
```

```
Change [month day year]:
```

```
Expire [month day year]:
```

```
Class:
```

```
Home directory: /home/jru
```

```
Shell: /usr/local/bin/tcsh
```

```
Full Name: J. Random User
```

```
Office Location:
```

```
Office Phone:
```

```
Home Phone:
```

```
Other information:
```

The normal user can change only a small subsection of this information, and only for themselves.

例如 8-4. Interactive chpass by Normal User

```
#Changing user database information for jru.
```

```
Shell: /usr/local/bin/tcsh
```

Full Name: J. Random User

Office Location:

Office Phone:

Home Phone:

Other information:

8.6.5 passwd

passwd 是改变你自己的密码的常用方法。

注意:在改变密码前用户必须键入原来的密码。当使用者离开他们的控制台时，可以阻止一个没有经过认证的人改变他们的密码。

例如 8-5. passwd

```
% passwd
```

```
Changing local password for jru.
```

```
Old password:
```

```
New password:
```

```
Retype new password:
```

```
passwd: updating the database...
```

```
passwd: done
```

```
# passwd j ru
```

```
Changing local password for jru.
```

```
New password:
```

```
Retype new password:
```

```
passwd: updating the database...
```

```
passwd: done
```

8.7 受限制的用户

如果你运行一个多用户系统，你不信任的用户对系统所作的修改可能会损坏你的系统。FreeBSD 提供了系统管理员限制用户访问系统资源的方法。这些限制通常被分成两种：磁盘配额和其他资源限制。磁盘配额为系统管理员提供了一个告诉文件系统给用户使用多少磁盘空间的方法。而且，它还提供了一种快速检查用户所使用的磁盘数量而不需要时刻计算的方法。

法。配额将在第 11 章讨论。其他资源限制包括限制 CPU、memory 的数量和用户可能会使用的其他资源。这些是通过对登陆进行分类来完成的，下面将作讨论。

登陆的类由 `/etc/login.conf` 文件来定义。比较精确的表述超出了本章的范围，但 `login.conf` 的参考文档会有比较细致的描述。

资源限制与普通的登陆限制是有区别的。首先，对于每一种限制，有软限制和硬限制之分。一个软限制可能被用户或应用程序调整过了，但不会超越硬限制。越往后可能会越降低，但不会升高。第二，绝大多数资源限制会分配每个处理给一个特殊的用户。

下面就是绝大多数资源限制的例子：

`coredumpsize`

很明显，由程序产生的核心文件大小的限制在磁盘使用上是从属于其他限制的（如，文件大小，或磁盘配额）。然而，既然用户自己无法产生核心文件，而且经常不删除它们，设置这个可以减少由于一个大型应用程序的崩溃所造成的大量磁盘空间的浪费。

`cputime`

这是一个用户程序所能消耗掉的最大的 CPU 时钟数量。一些不理想的进程会被内核杀掉。

注意：这是一个有关 CPU 消耗的时钟的限制问题，不是在使用 `top` 和 `ps` 命令时屏幕上显示的 CPU 消耗的百分比。

`filesize`

这是用户可以处理的一个文件的最大值。不象磁盘配额，这个限制是对单个文件强制执行的。

`maxproc`

这是一个用户可以运行的最大的进程数。这包括前台和后台处理。很明显，这不可能比系统指定的限制要大。当然，如果设置得太小可能会削弱用户的处理能力：可能需要多次登陆或执行多个管道。一些任务，象编译一些大的程序，也可能产生多进程。（象 `make`, `cc`, 和其他一些预处理程序）。

`memorylocked`

这是一个进程可能会被锁定到主内存中的最大内存数量。一些比较大型的程序，象

amd ,这样做，在遇到问题时，他们得到的巨大交换量无法传递给系统进行处理。

memoryuse

这是在给定时间内一个进程可能消耗的最大的内存数量。它包括核心内存和交换内存。在限制内存消耗方面，这个不是一个完全的限制，但它是一个好的开始。

openfiles

这是一个进程可以打开的最大的文件数。在 FreeBSD 中，文件可能用来表现套接字和 IPC 通道；然而，注意不要把这个设置得太小。对这个更深入的限制是由 kern.maxfiles sysctl 来定义的。

sbsize

这是网络内存数量的限制。这可以通过创建许多套接字来生成一些针对老式的 DoS 的攻击的回应，但它通常被用来限制网络通讯。

stacksize

这是一个进程堆栈可能达到的最大值。这个不能单独地限制一个程序可能使用的内存数量，而是要和其他的限制一起配合。

在设置资源限制时，有一些其他的事情需要记住。下面是一些通常的技巧，建议，和各种注意事项。

1. 系统启动的进程会被指派给后台的登陆类。
2. 虽然来自系统的 */etc/login.conf* 文件是一个对于绝大多数的限制做合理配置的资源文件，但只有你，系统管理员，才能知道什么对你的系统才是最适当的。限制设得太高可能会把你的系统开放得太大而被人滥用，而设置得太低可能会处理时效率很低。
3. X 视窗系统的用户可能要比其他用户使用更多的资源。X11 本身就要使用很多资源，但它也可以让用户同时运行更多的程序。
4. 记住许多限制会被应用于单独的处理进程，不是所有的用户。例如，设置 openfiles 为 50 意味着用户运行的每个进程可能最高只能打开 50 个文件。然而，用户可以打开的文件的总的大小是根据 maxproc 的值逐步增加的 openfiles 的值。这也适用于内存的消耗。有关资源限制，登陆类的更深入信息可以看看相关的联机手册：
cap_mkdb, getrlimit, login.conf。

8.8 私有化用户

本地化是由系统管理员或用户设置的一个环境,它可以用来调整不同的语言,字符设置,时钟标准等。这将在第 14 章本地化-I18N/L10N 使用与设置作详细讨论。

8.9 组

组简单来讲就是许多用户的列表。组可以通过他们的组名和他们的编号来识别。在 FreeBSD (和其他绝大多数的 Unix 系统)中,这两个要素通常被内核用来决定一个允许被执行的进程是否是它的用户 ID,还是它所属的组的列表。不象用户 ID,一个进程有一个与它相关联的组的列表。你可能听说过一些有关一个用户或进程的组 ID 的事情;在大多数情况下,这只意味着在列表中的第一个组。

与组 ID 地图对应的组名在 */etc/group* 中。这是一个用四个冒号来界定的文本文件。第一部分是组名,第二部分是加密的密码,第三部分是组 ID,第四部分是以逗号分割的成员列表。它可以用手工的方式进行编辑。对于更完整的描述,可以参看 *group* 的参考页。如果你不想手工编辑 */etc/group*,你可以使用 *pw* 命令来增加和编辑组。例如,要增加一个叫 *teamtwo* 的组,确信它存在:

例如 8-6. 使用 *pw* 增加一个组:

```
# pw groupadd teamtwo
# pw groupshow teamtwo
teamtwo: *:1100:
```

上面的 1100 数字是组 *teamtwo* 的组 ID。在这儿, *teamtwo* 没有成员,那它也就没有多大用处。

例如 8-7. 使用 *pw* 在组中添加一些成员:

```
# pw groupmod teamtwo -M jru
# pw groupshow teamtwo
teamtwo: *:1100:jru
```

使用 *-M* 参数为了用逗号划分开一个组成员中的用户的列表。你可能知道密码文件也会为每个用户包含一个组;当使用 *pw* 来询问组成员的时候,在密码文件中的组会自动被添加到组列表中,而不会出现在成员列表中。如果你想知道一个用户属于哪个组,你可以使用 *id* 命令:

例如 8-8. 使用 `id` 来决定组成员

```
% id jru
```

```
uid=1001(jru) gid=1001(jru) groups=1001(jru), 1100(teamtwo)
```

正如你所看到的，`jru` 是组 `jru` 和 `teamtwo` 的成员。

有关 `pw` 的更多信息，可以参看它的联机手册，更多有关 `/etc/group` 格式的信息，可参考 `group` 的联机手册。