Linux培训系列

————第三讲

讲述不同的主题（包括：系统和因特网文档、Linux 权限模式、用户帐户管理以及登录环境调节），我们将使您的基本的 Linux 管理技能方面的知识趋于完善。

内容基础，语言简短简洁

红联Linux论坛是致力于Linux技术讨论的站点，目前网站收录的文章及教程基本能满足不同水平的朋友学习。

红联Linux门户： www.linux110.com
红联Linux论坛： www.linuxdiyf.com/bbs
下载:Linux电子书籍：
http://www.linux286.com/linux/linuxdzsj.htm

目录

系统和网络文档

系统和网络文档－Linux 系统文档的类型

发布时间 :2007-01-26 23:08:06

从本质上说，Linux 系统中有三种文档资源：手册页、信息页和 /usr/share/doc 中的应用程序随附的文档。在本章中，我们将揭示浏览这其中的每一种资源的方法，而不用“ 突破常规 ”地查找更多信息。

手册页

发布时间 :2007-01-26 23:08:37

手册页（manual pages，或“ man pages ”）是 UNIX 和 Linux 的参考文档的典型形式。理想的情况是，您可以在手册页中查找任何命令、配置文件或库例程的信息。实际上，由于 Linux 是自由软件，一些手册页没有编写，或者显得过时了。虽然如此，当您需要帮助的时候，手册页仍是您的首选。

要访问手册页，只需输入 man，后面跟上您要查询的主题。页面阅读器（pager）将被启动，那么当您完成阅读时，您需要按 q。例如，为了查找关于 ls 命令的信息，您要输入：

```
$ man ls
```

手册页（续）

发布时间 :2007-01-26 23:09:13

了解手册页的布局对于快速地转到您所需要的信息很有帮助。一般来说，您将在手册页中找到下面这些章节：

NAME 命令的名称和单行描述
SYNOPSIS 怎样使用命令
DEscr ī ptION 命令功能的深入讨论
EXAMPLES 怎样使用命令的建议
SEE ALSO 相关主题（通常是手册页）

手册页章节

发布时间 :2007-01-26 23:09:51

构成手册页的这些文件存储在 /usr/share/man 中（或者有些旧一点的系统存储在 /usr/man 中）。在该目录内，您将发现手册页被组织成下面这些章节：

man1 用户程序
man2 系统调用
man3 库函数
man4 特殊文件
man5 文件格式
man6 游戏
man7 其它

多个手册页

发布时间 :2007-01-26 23:10:35

有些主题在多个章节中存在。为了说明这一点，我们来使用 whatis 命令，它将显示一个主题所有可用的手册页：

```
$ whatis printf
printf          (1)  - format and print data
printf          (3)  - formatted output conversion
```

在这种情况下，man printf 将第 1 节（“ 用户程序 ”）中的页面作为缺省手册页。如果我们正在写 C 程序，我们可能对第 3 节（“ 库函数 ”）中的页面更感兴趣。您可以通过在命令行中指定章节来打开某一章节中的手册页，因此要打开 printf(3)，我们将输入：

```
$ man 3 printf
```

查找正确的手册页

发布时间 :2007-01-26 23:11:08

有时，对于给定的主题很难找到正确的手册页。在这种情况下，您可以试着使用 man -k 来搜索手册页的 NAME 这一节。请注意这是子串搜索，因此运行像 man -k ls 这样的命令将给出一大堆输出！下面是使用更具体的查询的一个示例：

```
$ man -k whatis
apropos          (1)  - search the whatis database for strings
makewhatis       (8)  - Create the whatis database
whatis           (1)  - search the whatis database for complete words
```

所有关于 apropos 的内容

发布时间 :2007-01-26 23:11:44

啊，前一屏的这个示例引出了两点更多的内容！第一，apropos 命令正好等价于 man -k。（事实上，我要让您知道一些小窍门。当您运行 man -k 时，它实际在幕后运行 apropos。）第二点是 makewhatis 命令，它扫描您的 Linux 系统上的所有手册页，并且为 whatis 和 apropos 构建数据库。通常这由 root 用户定期运行，从而使数据库保持更新：

makewhatis

要获取关于“ man ”及其参数的更多信息，您应该从它本身的手册页开始：

$ man man

MANPATH

发布时间 :2007-01-26 23:12:14

缺省情况下，man 程序将在 /usr/share/man、/usr/local/man、/usr/X11R6/man 以及还可能在 /opt/man 中查找手册页。有时，您可能发现您需要给该搜索路径添加一个附加项。如果是这样，只需在文本编辑器中编辑 /etc/man.conf，添加一行类似这样的内容：

MANPATH /opt/man

从这一点向前，将找到 /opt/man/man* 目录中的所有手册页。请记住您将需要重新运行 makewhatis，从而将这些新手册页添加到 whatis 数据库中。

GNU 信息

发布时间 :2007-01-26 23:12:46

手册页的一个缺点是它们不支持超文本，因此您不能容易地从一个地方跳到另一个地方。GNU 的工作者们意识到了这个缺点，因此他们发明了另一种文档格式：“ 信息 ”页。许多 GNU 程序带有信息页形式的扩展文档。您可以用“ info ”命令开始阅读信息页：

$ info

以这种方式调用 info 将在系统上生成可用页面的索引。您可以用箭头键在上面来回移动，使用 enter 键进入链接（用星号表明），按 q 退出。这些键是基于 Emacs 的，因此如果您对这种编辑器很熟悉，那么您应该能够很容易地浏览。

GNU 信息（续）

发布时间 :2007-01-26 23:13:18

您也可以在命令行中指定信息页：

$ info diff

为了获取关于使用 info 阅读器更多的信息，请试着阅读它的信息页。您应该能够自己学会使用我已经提到的几个键进行浏览：

$ info info

/usr/share/doc

发布时间 :2007-01-26 23:13:54

Linux 系统上还有最后一种帮助资源。许多程序还带有其它格式的附加文档：文本、PDF、Postscr ī pt、HTML，这里仅举出几种。在 /usr/share/doc 中（或者旧一些的系统上的 /usr/doc）看一看。您将发现一个很长的目录列表，其中每个目录都带有您系统上的某个应用程序。搜索该文档通常可以发现一些在手册页或信息页中找不到的精品，比如教程或附加的技术文档。快速检查将发现这里有大量有用的阅读材料：

```
$ cd /usr/share/doc
$ find . -type f | wc -l
  7582
```

哎呀！今晚您的家庭作业就是阅读这些文档的一半（3791）。等着明天测验哦。

Linux 文档计划

发布时间 :2007-01-26 23:14:25

除系统文档之外，因特网上有很多优秀的 Linux 参考资料。“ Linux 文档计划 ”（Linux Documentation Project ）是一群志愿者将完整的免费 Linux 文档系列放在一起的行动。该计划的存在是为了将 Linux 文档的不同片断放在容易搜索和使用的地方。

LDP 概述

发布时间 :2007-01-26 23:15:07

LDP 由下面这些方面组成：

指南 — 更长、更深入的书籍，比如 The Linux Programmer's Guide
HOWTO — 特定主题的帮助，比如 DSL HOWTO
FAQ — 带有回答的“ 常见问题 ”（Frequently Asked Questions），比如 Brief Linux FAQ
手册页 — 个别命令的帮助（它们与您在 Linux 上使用 man 命令时所得到的手册页相同）

邮件列表

发布时间 :2007-01-26 23:15:39

邮件列表很可能提供了 Linux 开发者相互合作的最重要的因素。项目一般由相隔遥远的贡献者制定，他们甚至可能在地球的两端。对于一个项目，邮件列表为每个开发者提供一种方法，使他们可与所有其他开发者联系，还可以通过电子邮件举行小组讨论。最有名的开发邮件列表之一是“ Linux Kernel Mailing List ”，在 http://www.tux.org/lkml/ 中有描述。

更多的关于邮件列表的内容

发布时间 :2007-01-26 23:16:21

除了开发之外，邮件列表还可以提供提问以及从博学的开发者，或者甚至从其他用户那里得到回答的方法。例如，单独的分发包经常给新来者提供邮件列表。您可以检查您的分发包的 Web 站点，以获取所提供的关于邮件列表的信息。

如果您花时间阅读过前一屏上链接的 LKML FAQ，您可能已经注意到邮件列表订户通常对于被重复问到的问题不太友好。在写问题之前，搜索一下所给邮件列表的归档文件总是很明智的。这很可能也将节省您的时间！

新闻组

发布时间 :2007-01-26 23:17:10

因特网“ 新闻组 ”类似于邮件列表，但是它基于叫做 NNTP（“ Network News Transfer Protocol ”，网络新闻传输协议）的协议，而不使用电子邮件。为了参加新闻组，您需要使用 NNTP 客户端程序，比如 slrn 或 pan。其主要的优点是您可以只参加您想参加的讨论，而不会总有邮件发到您的信箱中。

最有影响力的新闻组的讨论是 comp.os.linux。您可以在 LDP 站点 http://www.linuxdoc.org/linux/#ng 上浏览该列表。

和邮件列表一样，新闻组讨论经常被归档。一个很受欢迎的新闻组归档站点是 Deja News。

供应商和第三方 Web 站点

发布时间 :2007-01-26 23:17:50

各种 Linux 分销商的 Web 站点经常提供更新的文档、安装说明、硬件兼容性／不兼容性声明以及其它支持, 如知识库搜索工具。例如：

Redhat Linux
Debian Linux
Gentoo Linux
SuSE Linux
Caldera
Turbolinux

Linux 权限模型

Linux 权限模型－一个用户、一个组

发布时间 :2007-01-26 23:18:54

在这一章，我们将来看一看 Linux 权限和所有权模型。我们已经看到每个文件属于一个用户和一个组。这正是 Linux 中权限模型的核心。您可以在 ls -l 清单中查看用户和组：

```
$ ls -l /bin/bash
-rwxr-xr-x    1 root     wheel      430540 Dec 23 18:27 /bin/bash
```

在这个特殊的示例中，/bin/bash 可执行文件属于 root 用户，并且在 wheel 组中。Linux 权限模型通过允许给每个文件系统对象设置三种独立的权限级别来工作 — 它们为文件的所有者、文件的组以及所有其他用户。

理解“ ls -l ”

发布时间 :2007-01-26 23:19:28

我们来看一看我们的 ls -l 输出，检查一下这个清单的第一栏：

```
$ ls -l /bin/bash
-rwxr-xr-x    1 root     wheel      430540 Dec 23 18:27 /bin/bash
```

第一个字段 -rwxr-xr-x 包含该特殊文件的权限的符号表示。该字段中的首字符（-）指定该文件的类型，本例中它是一个常规文件。其它可能的首字符还有：

“ d ” 目录
“ l ” 符号链接
“ c ” 字符专门设备文件
“ b ” 块专门设备文件
“ p ” 先进先出
“ s ” 套接字

三个三元组

发布时间 :2007-01-26 23:20:01

```
$ ls -l /bin/bash
-rwxr-xr-x    1 root     wheel      430540 Dec 23 18:27 /bin/bash
```

该字段的其余部分由三个三元组字符组成。第一个三元字符组代表文件所有者的权限，第二个代表文件的组的权限，第三个代表所有其他用户的权限：

"rwx"
"r-x"
"r-x"

上面，r 表示允许读（查看文件中的数据），w 表示允许写（修改文件以及删除），x 表示允许“ 执行 ”（运行程序）。将所有这些信息放在一起，我们可以发现每个人都能够读该文件的内容和执行该文件，但是只允许文件所有者（root 用户）可以以任何方式修改该文件。因此，虽然一般用户可以复制该文件，但是只允许 root 用户更新或删除它。

我是谁？

发布时间 :2007-01-26 23:20:35

在我们看怎样改变文件的用户所有权和组所有权之前，我们首先来看一看怎样得知您当前的用户标识和组成员资格。除非最近您使用过 su 命令，否则您当前的用户标识是您用来登录系统的用户标识。但是，如果您经常使用 su，您可能不记得您当前有效的用户标识。要查看用户标识，输入 whoami：

```
# whoami
root
# su drobbins
$ whoami
drobbins
```

我在哪一组？

发布时间 :2007-01-26 23:21:08

要看看您属于哪一组，使用 group 命令：

```
$ groups
drobbins wheel audio
```

在上面的示例中，我是 drobbins、wheel 和 audio 组的成员。如果您想看看其他用户在什么组，指定他们的用户名作为参数：

```
$ groups root daemon
root : root bin daemon sys adm disk wheel floppy dialout tape video
daemon : daemon bin adm
```

改变用户和组所有权

发布时间 :2007-01-26 23:21:42

为了改变文件或其它文件系统对象的所有者或组，分别使用 chown 或 chgrp。这两个命令都要一个用户名或组名作参数，后面跟上一个或多个文件名。

```
# chown root /etc/passwd
# chgrp wheel /etc/passwd
```

您还可以用 chown 命令的另一种形式同时设置所有者和组：

```
# chown root.wheel /etc/passwd
```

除非您是超级用户，否则您不可以使用 chown，然而任何人都可以使用 chgrp 来将文件的组所有权改为他们所属的组。

递归的所有权改变

发布时间 :2007-01-26 23:22:11

chown 和 chgrp 都有一个 -R 选项，该选项可以用来告诉它们递归地将所属权和组改变应用到整个目录树中。例如：

chown -R drobbins /home/drobbins

介绍 chmod

发布时间 :2007-01-26 23:22:40

chown 和 chgrp 可以用来改变文件系统对象的所有者和组，而另一个程序 — 叫做 chmod — 用来改变我们可以在 ls -l 清单中看到的 rwx 权限。chmod 带有两个或多个参数：“ mode ”，描述怎样改变权限，后面跟将会受到影响的文件或文件列表：

$ chmod +x scr ī ptfile.sh

在上面的示例中，我们的“ mode ”是 +x。您可能会猜到，+x 模式告诉 chmod，使该特殊文件对于用户、组以及其它任何人都是可执行的。

如果我们想要除去一个文件的所有执行权限，我们应该这样做：

$ chmod -x scr ī ptfile.sh

用户／组／其他粒度

发布时间 :2007-01-26 23:23:08

到此，我们的 chmod 示例已经影响到了所有三个三元组 — 用户、组和所有其他用户。通常，一次只修改一个或两个三元组很方便。要这样做，只需要在 + 或 - 符号之前，给您想要修改的特定的三元组指定符号字符。对于“ 用户 ”三元组使用 u，对于“ 组 ”三元组使用 g，对于“ 其他／每个人 ”使用 o：

$ chmod go-w scr ī ptfile.sh

我们刚除去了组和所有其他用户的写权限，而保留“ 所有者 ”权限不动。

重新设置权限

发布时间 :2007-01-26 23:23:36

除了交替打开和关闭权限位以外，我们还可以一起重新设置它们。通过使用 = 操作符，我们可以告诉 chmod 我们要指定权限和取消别的权限：

$ chmod =rx scr ī ptfile.sh

上面，我们只设置了所有的“ read ”和“ execute ”位，没有设置所有的“ write ”位。如果您仅仅想重新设置特定的三元组，您可以像下面这样，在 = 之前指定该三元组的符号名：

$ chmod u=rx scr ī ptfile.sh

数字模式

发布时间 :2007-01-26 23:24:09

直到现在为止，我们使用了叫做“ 符号 ”的模式来用 chmod 指定权限的改变。然而，指定权限还有一种普遍使用的方法 — 使用 4 位八进制数。使用叫做数字权限语法的语法，每一位代表一个权限三元组。例如，在 1777 中，777 设置本章我们所讨论的“ owner ”、“ group ”和“ other ”标志。1 用来设置专门的权限位，我们将在本章的结束部分讲到。这个图表说明了怎样解释第二到四位（777）：

模式 数字
rwx 7
rw- 6
r-x 5
r-- 4
-wx 3
-w- 2
--x 1
--- 0

数字权限语法

发布时间 :2007-01-26 23:24:40

当您需要给一个文件指定所有权限时，数字权限语法特别有用，比如在下面的示例中：

```
$ chmod 0755 scr ī ptfile.sh
$ ls -l scr ī ptfile.sh
-rwxr-xr-x    1 drobbins drobbins        0 Jan  9 17:44 scr ī ptfile.sh
```

在该示例中，我们使用了 0755 模式，它展开为一个完整的权限设置“ -rwxr-xr-x ”。

umask

发布时间 :2007-01-26 23:25:11

当进程创建了新文件时，它指定新文件应该具有的权限。通常，所请求的模式是 0666（每个人可读和可写），它比我们希望的具有更多的权限。幸运的是，不管什么时候创建了新文件，Linux 将参考叫做“ umask ”的东西。系统用 umask 值来将初始指定的权限降低为更合理、更安全的权限。您可以通过在命令行中输入 umask 来查看您当前的 umask 设置：

```
$ umask
0022
```

Linux 系统上，umask 的缺省值一般为 0022，它允许其他人读您的新文件（如果他们可以得到它们），但是不能进行修改。

umask（续）

发布时间 :2007-01-26 23:25:50

为了在缺省的情况下使新文件更安全，您可以改变 umask 设置：

$ umask 0077

umask 将确保组和其他用户对于新创建的文件绝对没有任何权限。那么，umask 怎样工作呢？与文件的“ 常规 ”权限不同，umask 指定应该关闭哪一个权限。我们来参阅一下我们的“ 模式到数字 ”映射表，从而使我们可以理解 0077 的 umask 的意思是什么：

模式 数字
rwx 7
rw- 6
r-x 5
r-- 4
-wx 3
-w- 2
--x 1
--- 0

使用该表，0077 的最后三位扩展为 ---rwxrwx。现在，请记住 umask 告诉系统禁用哪个权限。根据推断，我们可以看到将关闭所有“ 组 ”和“ 其他 ”权限，而“ 用户 ”权限将保留不动。

介绍 suid 和 sgid

发布时间 :2007-01-26 23:26:24

当您最初登录时，将启动一个新的 shell 进程。您已经知道，但是您可能还不知道这个新的 shell 进程（通常是 bash）使用您的用户标识运行。照这样，bash 程序可以访问所有属于您的文件和目录。事实上，作为用户，我们完全依靠其它程序来代表我们执行操作。因为您启动的程序继承了您的用户标识，因此它们不能访问任何不允许您访问的文件系统对象：

介绍 suid 和 sgid（续）

发布时间 :2007-01-26 23:26:56

例如，一般用户不能直接修改 passwd 文件，因为“ write ”标志已经对除“ root 用户 ”以外的每个用户关闭：

```
$ ls -l /etc/passwd
-rw-r--r--    1 root     wheel        1355 Nov  1 21:16 /etc/passwd
```

但是，一般用户确实需要在他们需要改变其密码的任何时候，能够修改 /etc/passwd（至少间接地）。但是，如果用户不能修改该文件，究竟怎样完成这个工作呢？

suid

发布时间 :2007-01-26 23:27:27

幸好，Linux 权限模型有两个专门的位，叫做 “ suid ” 和 “ sgid ” 。当设置了一个可执行程序的 “ suid ” 这一位时，它将代表可执行文件的所有者运行，而不是代表启动程序的人运行。

现在，回到 /etc/passwd 问题。如果看一看 passwd 可执行文件，我们可以看到它属于 root 用户：

```
$ ls -l /usr/bin/passwd
-rwsr-xr-x    1 root     wheel       17588 Sep 24 00:53 /usr/bin/passwd
```

您还将注意到，这里有一个 s 取替了用户权限三元组中的一个 x。这表明，对于这个特殊程序，设置了 suid 和可执行位。由于这个原因，当 passwd 运行时，它将代表 root 用户执行（具有完全超级用户访问权），而不是代表运行它的用户运行。又因为 passwd 以 root 用户访问权运行，所以能够修改 /etc/passwd 文件，而没有什么问题。

suid/sgid 告诫说明

发布时间 :2007-01-26 23:27:56

我们看到了 suid 怎样工作，sgid 以同样的方式工作。它允许程序继承程序的组所有权，而不是当前用户的程序所有权。

这里有一些关于 suid 和 sgid 的其它的但是很重要的信息。首先，suid 和 sgid 占据与 ls -l 清单中 x 位相同的空间。如果还设置了 x 位，则相应的位表示为 s（小写）。但是，如果没有设置 x 位，它将表示为 S（大写）。

另一个很重要的提示：在许多环境中，suid 和 suid 很管用，但是不恰当地使用这些位可能使系统的安全遭到破坏。最好尽可能地少用“ suid ”程序。passwd 命令是为数不多的必须使用“ suid ”的命令之一。

改变 suid 和 sgid

发布时间 :2007-01-26 23:28:25

设置和除去 suid 与 sgid 位相当简单。这里，我们设置 suid 位：

```
# chmod u+s /usr/bin/myapp
```

此处，我们从一个目录除去 sgid 位。我们将看到 sgid 位怎样影响下面几屏中的目录：

```
# chmod g-s /home/drobbins
```

权限和目录

发布时间 :2007-01-26 23:28:52

到此为止，我们从常规文件的角度来看权限。当从目录的角度看权限时，情况有一点不同。目录使用同样的权限标志，但是它们被解释为表示略微不同的含义。

对于一个目录，如果设置了“ read ”标志，您可以列出目录的内容；“ write ”表示您可以在目录中创建文件，“ execute ”表示您可以进入该目录并访问内部的任何子目录。没有“ execute ”标志，目录内的文件系统对象是不可访问的。没有“ read ”标志，目录内的文件系统对象是不可查看的，但是只要有人知道磁盘上对象的完整路径，就仍然可以访问目录内的对象。

目录和 sgid

发布时间 :2007-01-26 23:29:24

如果启用了目录的“ sgid ”标志，在目录内创建的任何文件系统对象将继承目录的组。当您需要创建一个属于同一组的一组人使用的目录树时，这种特殊的功能很管用。只需要这样做：

```
# mkdir /home/groupspace
# chgrp mygroup /home/groupspace
# chmod g+s /home/groupspace
```

现在，mygroup 组中的所有用户都可以在 /home/groupspace 内创建文件或目录，同样，他们也将自动地分配到 mygroup 的组所有权。根据用户的 umask 设置，新文件系统对象对于 mygroup 组的其他成员来说，可以或不可以是可读、可写或可执行的。

目录和删除

发布时间 :2007-01-26 23:29:51

缺省情况下，Linux 目录以一种不是在所有情况下都很理想的方式表现。一般来说，只要对一个目录有写访问权，任何人都可以重命名或删除该目录中的文件。对于个别用户使用的目录，这种行为是很合理的。

但是，对于很多用户使用的目录来说，尤其是 /tmp 和 /var/tmp，这种行为可能会产生麻烦。因为任何人都可以写这些目录，任何人都可以删除或重命名任何其他人的文件 — 即使是不属于他们的！显然，当任何其他用户在任何时候都可以输入“ rm -rf /tmp/* ”并损坏每个人的文件时，很难把 /tmp 用于任何有意义的文件。

所幸，Linux 有叫做“ 粘滞位 ”（sticky bit）的东西。当给 /tmp 设置了粘滞位（用 chmod +t），唯一能够删除或重命名 /tmp 中文件的是该目录的所有者（通常是 root 用户）、文件的所有者或 root 用户。事实上，所有 Linux 分发包都缺省地启用了 /tmp 的粘滞位，而您还可以发现粘滞位在其它情况下也很管用。

难以理解的第一位

发布时间 :2007-01-26 23:30:33

总结本章，我们最后来看一看数字模式的难以理解的第一位数。您可以看到，这个第一位数用来设置 sticky、suid 和 sgid 位：

suid sgid sticky 模式数字
on on on 7
on on off 6
on off on 5
on off off 4
off on on 3
off on off 2
off off on 1
off off off 0

这里有一个怎样用 4 位数字模式来设置一个目录的权限的示例，该目录将由一个工作组使用：

chmod 1775 /home/groupfiles

作为家庭作业，请想一想 1755 数字模式权限设置的含义。

Linux 帐户管理

Linux 帐户管理－介绍 /etc/passwd

发布时间 :2007-01-26 23:31:34

在这一章，我们来看一看 Linux 帐户管理机制。我将以介绍 /etc/passwd 文件开始，该文件定义了 Linux 系统上存在的所有用户。您可以通过输入 “ less /etc/passwd ” 来查看您自己的 /etc/passwd 文件。

/etc/passwd 中的每一行定义一个用户帐户。这里有一个来自于我的 /etc/passwd 文件的示例行：

drobbins:x:1000:1000:Daniel Robbins:/home/drobbins:/bin/bash

您可以看到，这一行中有相当多的信息。实际上，每个 /etc/passwd 行由多个字段组成，每个字段用 : 隔开。

第一个字段定义了用户名（drobbins），第二个字段包含一个 x。在旧式的 Linux 系统上，该字段将包含一个用来认证的加密密码，但是事实上现在所有的 Linux 系统将这个密码信息存储在另一个文件中。

第三个字段（1000）定义了与该特殊用户相关联的数字用户标识，第四个字段（1000）将用户与一个特殊组关联起来；在下面几屏中，我们将看到定义组 1000 的地方。

第五个字段包含该帐户的文本描述 — 在本例中，是用户的名称。第六个字段定义该用户的主目录，第七个字段指定用户缺省的 shell — 当用户登录时，将自动启动的 shell。

/etc/passwd 技巧和窍门

发布时间 :2007-01-26 23:32:10

您可能已经注意到，/etc/passwd 中定义的用户帐户比实际登录您系统的用户帐户多得多。这是因为不同的 Linux 组件使用用户帐户来加强安全性。通常，这些系统帐户有一个小于 100 的用户标识（“ uid ”），这其中的很多系统帐户将像 /bin/false 这样的程序列为缺省的 shell。因为 /bin/false 程序什么也不做，而是返回一个错误码退出，这有效地阻止这些帐户被用作登录帐户 — 他们只供内部使用。

/etc/shadow

发布时间 :2007-01-26 23:32:42

这样，用户帐户本身在 /etc/passwd 中定义。Linux 系统包含一个 /etc/passwd 的同伴文件，叫做 /etc/shadow 。该文件不像 /etc/passwd，只有对于 root 用户来说是可读的，并且包含加密的密码信息。我们来看一看 /etc/shadow 的一个样本行：

drobbins:$1$12345678901234567890123456789O1:11664:0:-1:-1:-1:-1:0

每一行给一个特殊帐户定义密码信息，同样的，每个字段用 : 隔开。第一个字段定义与这个 shadow 条目相关联的特殊用户帐户。第二个字段包含一个加密的密码。其余的字段在下表中描述：

字段 3 自 1/1/1970 起，密码被修改的天数
字段 4 密码将被允许修改之前的天数（0 表示 “ 可在任何时间修改 ”）
字段 5 系统将强制用户修改为新密码之前的天数（1 表示 “ 永远都不能修改 ”）
字段 6 密码过期之前，用户将被警告过期的天数（-1 表示 “ 没有警告 ”）
字段 7 密码过期之后，系统自动禁用帐户的天数（-1 表示 “ 永远不会禁用 ”）
字段 8 该帐户被禁用的天数（-1 表示 “ 该帐户被启用 ”）
字段 9 保留供将来使用

/etc/group

发布时间 :2007-01-26 23:33:25

接下来，我们来看一看 /etc/group 文件，它定义了 Linux 系统上所有的组。这里有一个样本行：

drobbins:x:1000:

/etc/group 字段格式如下。第一个字段定义组名称，第二个字段是不再使用的密码字段（现在只是保留为 x），第三个字段定义了这个特殊组的数字组标识，第四个字段（上面的示例为空）定义是该组成员的所有用户。

您将回想起样本 /etc/passwd 行引用的组标识为 1000。即使 /etc/group 的第四个字段没有列出 drobbins 用户名，这将起到把 drobbins 用户放到 drobbins 组中的作用。

组提示

发布时间 :2007-01-26 23:33:56

关于用户和组相关联的一点提示 — 在一些系统上，您将发现每个新的登录帐户与同名组（通常是标识号一样）相关联。在其它系统上，所有登录帐户将属于单个用户组。在您管理的系统上，您使用的方法取决于您自己。为每个用户创建匹配组的好处是，通过将可信的朋友放在自己的个人组中，使用户能够更容易地控制对自己文件的访问权。

手工地添加用户和组

发布时间 :2007-01-26 23:34:55

现在，我将为您展示怎样创建您自己的用户和组帐户。学习怎样完成这些工作的最好方法是，手工地将用户新添加到系统中。为了开始学习，首先确保您的 EDITOR 环境变量设置为您喜欢的文本编辑器：

```
# echo $EDITOR
vim
```

如果不是，您可以通过输入这样的命令来设置 EDITOR：

```
# export EDITOR=/usr/bin/emacs
```

现在，输入：

```
# vipw
```

现在您应该发现自己在所喜欢的文本编辑器中，编辑器内 /etc/passwd 文件被装入，显示在屏幕上。当修改系统 passwd 和 group 文件时，使用 vipw 和 vigr 命令非常重要。它们采用了额外的预防措施来确保您关键的 passwd 和 group 文件被恰当地锁定，使它们不会破坏。

编辑 /etc/passwd

发布时间 :2007-01-26 23:35:24

既然您已经打开了 /etc/passwd 文件，则接着添加下面的代码行：

testuser:x:3000:3000:LPI tutorial test user:/home/testuser:/bin/false

我们刚刚添加了一个 UID 为 3000 的 “ testuser ” 用户。我们将他添加到 GID 为 3000 的组中，该组还未创建。另一种做法是，如果愿意，我们还可以给这个用户分配 users 组的 GID。这个新用户有一条注释为：LPI tutorial test user；该用户的主目录设置为 /home/testuser，出于安全的目的，该用户的 shell 设置为 /bin/false。如果我们正创建一个非测试帐户，那么我们可以将 shell 设置为 /bin/bash。OK，接着保存您所做的更改，然后退出。

编辑 /etc/shadow

发布时间 :2007-01-26 23:35:55

现在，我们需要在 /etc/shadow 中给这个特殊用户添加一个条目。要这样做，输入 vipw -s。您将会看到您喜欢的编辑器，它现在包含 /etc/shadow 文件。现在，接着复制一个现有用户帐户行（也就是有一个密码，并且长于标准系统帐户条目）：

drobbins:$1$12345678901234567890123456789O1:11664:0:-1:-1:-1:-1:0

现在，将所复制的代码行中的用户名改为您的新用户的名称，确保所有的字段（特别是密码的期限设置）设置为您喜欢的模式：

testuser:$1$12345678901234567890123456789O1:11664:0:-1:-1:-1:-1:0

保存，然后退出。

设置密码

发布时间 :2007-01-26 23:36:24

您将回到提示符。现在，是给您的新用户设置密码的时候了：

passwd testuser
Enter new UNIX password: (enter a password for testuser)
Retype new UNIX password: (enter testuser's new password again)

编辑 /etc/group

发布时间 :2007-01-26 23:36:56

既然 /etc/passwd 和 /etc/shadow 设置好了，现在该恰当配置 /etc/group 了。要这么做，输入：

vigr

您的 /etc/group 文件将出现在您面前，准备好进行编辑。现在，如果您选择给您的特殊的测试用户分配 users 缺省组，那么您不需要将任何组添加到 /etc/groups 中。但是，如果您选择给该用户创建新的组，接着添加下面的行：

testuser:x:3000:

保存，然后退出。

创建主目录

发布时间 :2007-01-26 23:37:30

我们基本上已经完工。输入下面的命令来创建 testuser 的主目录。

```
# cd /home
# mkdir testuser
# chown testuser.testuser testuser
# chmod o-rwx testuser
```

我们用户的主目录现在已经到位，并且帐户已准备好可用。好的，基本就绪。如果您想使用该帐户，您将需要使用 vipw 来将 testuser 的缺省 shell 改为 /bin/bash，使用户可以登录。

帐户管理实用程序

发布时间 :2007-01-26 23:38:04

既然您知道怎样手工添加新帐户和组，我将要评论一下 Linux 下可用的各种省时的帐户管理实用程序。由于版面的限制，我将不深究描述这些命令的众多细节。请记住，通过查看命令的手册页，您总能够获得关于命令的更多信息。如果您计划参加 LPIC 101 考试，我建议您花些时间来让您自己熟悉一下下面每一条命令。

newgrp

缺省情况下，用户创建的任何文件都被分配到 /etc/passwd 中所指定的用户的组。如果用户属于其他组，他或她可以输入 newgrp thisgroup 来将当前缺省组的成员资格设置为组 thisgroup。然后，所创建的任何新文件将继承该组的成员资格。

chage

chage 命令用来查看和改变存储在 /etc/shadow 中的密码期限设置。

gpasswd

一个一般目的的组管理工具

groupadd/groupdel/groupmod

用来在 /etc/group 中添加／删除／修改组

更多的命令

发布时间 :2007-01-26 23:38:32

useradd/userdel/usermod

用来在 /etc/passwd 中添加／删除／修改用户。这些命令还完成其它各种便利功能。要获取更多的信息，请参阅手册页。

pwconv/grpconv

用来将 passwd 和 group 文件转换为 “ 新式 ” 的 shadow 密码。事实上，所有 Linux 系统已经使用 shadow 密码，因此您应该不会需要使用这些命令。

pwunconv/grpunconv

用来将 passwd、shadow 和 group 文件转换成 “ 旧式 ” 的非 shadow 密码。您应该不会需要使用这些命令。

调节用户环境

调节用户环境－介绍“ fortune ”

发布时间 :2007-01-26 23:39:13

您的 shell 有很多可设置为适合您的个人爱好的有用的选项。但是，到目前为止，除了每次重新输入以外，我们还没有讨论到每次您登录时，自动设置这些设置的任何方法。在本章中，我们将看一看通过修改启动文件来调节您的登录环境。

首先，当您初次登录时，我们来添加一条友好的消息。要看示例消息，运行 fortune：

```
$ fortune
No amount of careful planning will ever replace dumb luck.
```

.bash_profile

发布时间 :2007-01-27 11:05:17

现在，我们来设置 fortune，使每次您登录时，它能运行。使用您喜欢的文本编辑器来编辑您的主目录中名为 .bash_profile 的文件。如果该文件还不存在，则接着创建它。在顶部插入一行：

fortune

试着注销，然后再回来。除非您正在运行一个像 xdm、gdm 或 kdm 这样的显示管理器，否则当您登录时，您应该会很愉快地看到：

mycroft.flatmonk.org login: chouser
Password:
Freedom from incrustations of grime is contiguous to rectitude.
$

登录 shell

发布时间 :2007-01-27 11:05:54

当 bash 启动，它将遍历您主目录中的 .bash_profile 文件，就象在 bash 提示符下输入命令一样运行每一行。这叫做 “ source ” 文件。

根据 bash 启动的方式，bash 的动作有些不同。如果它作为 “ 登录 ” shell 被启动，它将像上面那样动作 — 首先 source 系统范围的 /etc/profile，然后是您个人的 ~/.bash_profile。

告诉 bash 作为登录 shell 运行有两种方式。一种方式是当您初次登录时使用 — bash 由一个名为 -bash 的进程启动。您可以在您的进程清单中看到这些：

```
：$ ps u
USER       PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
chouser    404  0.0  0.0  2508  156 tty2     S     2001   0:00 -bash
```

您很可能看到长得多的清单，但是在您的 shell 名之前，至少有一个带有短划线的 COMMAND，如上面示例中的 -bash。shell 用这个短划线来确定它是否正作为 “ 登录 ” shell 运行。

理解 --login

发布时间 :2007-01-27 11:06:29

告诉 bash 作为“ 登录 ”shell 运行的第二种方法是用 --login 命令行选项。终端仿真器（如 xterm）有时使用这个选项来使它们的 bash 会话表现得像初始登录会话。

当您登录以后，将运行 shell 的更多副本。除非它们以 --login 启动或进程名中有短划线，否则这些会话将不是“ 登录 ”shell。但是，如果它们给出提示符，那么它们叫“ 交互式 ”shell。如果 bash 作为“ 交互式 ”shell 启动，而不是作为“ 登录 ”shell 启动，它将忽略 /etc/profile 和 ~/.bash_profile，而将 source ~/.bashrc。

interactive login profile rc
yes yes source ignore
yes no ignore source
no yes source ignore
no no ignore ignore

交互性测试

发布时间 :2007-01-27 11:07:03

有时 bash source 您的 ~/.bashrc，即使它不是真正的交互式 shell，比如当使用像 rsh 和 scp 这样的命令时。将这些牢记在心很重要，因为像前面我们用 fortune 命令所做的一样，打印出文本可能真的会打乱这些非交互式的 bash 会话。在从启动文件打印出文本之前，使用 PS1 变量来检测当前的 shell 是否是交互式 shell 是一个好办法：

```
if [ -n "$PS1" ]; then
 fortune
fi
```

/etc/profile 和 /etc/skel

发布时间 :2007-01-27 11:07:33

作为系统管理员，您掌管着 /etc/profile。因为当初次登录时，每个人都 source 它，所以使它保持工作状态很重要。它也是提供给新用户的强大工具，该工具使新用户一登录进他们的新帐户，一切就正确运行。

但是，有一些您可能希望新用户作为缺省值的设置，而且允许容易地修改它们。这是 /etc/skel 目录的用途所在。当您用 useradd 命令来创建一个新用户帐户时，它将所有的文件从 /etc/skel 复制到用户的新的主目录中。这意味着您可以将有帮助的 .bash_profile 和 .bashrc 文件放在 /etc/skel 中，使新用户有一个好的开始。

export

发布时间 :2007-01-27 11:08:10

可以给 bash 中的变量作上标记，使它们在任何 bash 启动的新的 shell 中设置相同；这被称为做上标记以便导出。在您的 shell 会话中，您可以列出 bash 所有的当前标记为要导出的变量：

```
$ export
declare -x EDITOR="vim"
declare -x HOME="/home/chouser"
declare -x MAIL="/var/spool/mail/chouser"
declare -x PAGER="/usr/bin/less"
declare -x PATH="/bin:/usr/bin:/usr/local/bin:/home/chouser/bin"
declare -x PWD="/home/chouser"
declare -x TERM="xterm"
declare -x USER="chouser"
```

标记要导出的变量

发布时间 :2007-01-27 11:08:46

如果变量没有标记为导出，任何它启动的新的 shell 将不会设置该变量。但是，您可以通过将变量传给内置的 export 来将其标记为导出：

```
$ FOO=foo
$ BAR=bar
$ export BAR
$ echo $FOO $BAR
foo bar
$ bash
$ echo $FOO $BAR
bar
```

在本示例中，一起设置了变量 FOO 和 BAR，但是只有 BAR 被标记为导出。当启动了新的 bash，它丢掉 FOO 的值。如果您退出这个新的 bash，您可以看到最初的 bash 仍然有 FOO 和 BAR 的值：

```
$ exit
$ echo $FOO $BAR
foo bar
```

导出和设置 -x

发布时间 :2007-01-27 11:09:19

由于这种行为，可以在 ~/.bash_profile 或 /etc/profile 中设置变量和标记为导出，然后再也不需要重新设置。但是，有一些不能导出的选项，因此为了设置得一致，必须将它们放在您的 ~/.bashrc 和 环境配置文件中。这些选项用内置的 set 来调整：

$ set -x

-x 选项使 bash 打印出它要运行的每个命令：

$ echo $FOO
+ echo foo
foo

这对于理解没有预料到的引用行为或类似的莫名其妙的现象非常有用。要关闭 -x 选项，设置 set +x。请参阅 bash 手册页来获取内置的 set 的所有选项的信息。

用“ set ”设置变量

发布时间 :2007-01-27 11:09:50

内置的 set 还可以用来设置变量，但这样使用时，它是可选的。bash 命令 set FOO=foo 表示的意思正好和 FOO=foo 相同。取消设置变量用内置的 unset 来完成：

```
$ FOO=bar
$ echo $FOO
bar
$ unset FOO
$ echo $FOO
```

取消设置与 FOO= 的比较

发布时间 :2007-01-27 11:10:26

这与将变量设置为什么也不设不相同，虽然有时很难区别。一种区别的方法是使用不带参数的内置的 set 来列出所有当前变量：

```
$ FOO=bar
$ set | grep ^FOO
FOO=bar
$ FOO=
$ set | grep ^FOO
FOO=
$ unset FOO
$ set | grep ^FOO
```

除了 set 列出所有变量而不仅仅是那些标记为要导出的变量外，像这样不带参数使用 set 与使用内置的 export 类似。

导出以改变命令行为

发布时间 :2007-01-27 11:10:57

通常，可以通过设置环境变量来改变命令的行为。正和新的 bash 会话一样，从您的 bash 提示符启动的其它程序将只能看见标记为导出的变量。例如，命令 man 检查变量 PAGER，看一看用什么程序来每次一页地遍历文本。

```
$ PAGER=less
$ export PAGER
$ man man
```

将 PAGER 设置为 less，您将每次看到一页，按空格键移到下一页。如果您将 PAGER 改为 cat，将立刻显示所有的文本，没有停顿。

```
$ PAGER=cat
$ man man
```

使用“ env ”

发布时间 :2007-01-27 11:11:30

不幸的是，如果您忘记将 PAGER 设置回 less，man（像其它命令一样）将继续没有停顿地显示所有的文本。如果您仅一次想将 PAGER 设为 cat，您可以使用 env 命令：

```
$ PAGER=less
$ env PAGER=cat man man
$ echo $PAGER
less
```

这一次，PAGER 值为 cat，被导出到 man，但在 bash 会话中，PAGER 变量本身仍然未改变。

汇集海量Linux技术文章

海量Linux技术文章

发布时间 :2006-11-24 16:50:29

下面是linux技术文章快速入口。需要联网：
Linux 技术交流

http://www.linuxdiyf.com/bbs/forum-3-1.html

Linux 应用

http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=1

Linux 安装及学习指导

http://www.linuxdiyf.com/bbs/forum-45-1.html

Linux 系统安装

http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=45&filter=type&typeid=11

Linux 学习指导

http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=45&filter=type&typeid=12

Linux 软件安装

http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=45&filter=type&typeid=13

shell

http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=3

Linux 壁纸

http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=4

红旗

http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=5

Redhat

http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=6

SuSE

http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=7

Linux 认证

http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=9

Linux下载分享{酷件、书籍、视频分享 }

http://www.linuxdiyf.com/bbs/forum-6-1.html

服务器应用

http://www.linuxdiyf.com/bbs/forum-7-1.html

数据库应用

http://www.linuxdiyf.com/bbs/forum-8-1.html

Linux 编程与内核

http://www.linuxdiyf.com/bbs/forum-9-1.html

UniX 技术文章

http://www.linuxdiyf.com/bbs/forum-32-1.html

Linux 业界声音、新闻

http://www.linuxdiyf.com/bbs/forum-11-1.html

Linux 人才招聘信息

http://www.linuxdiyf.com/bbs/forum-46-1.html

网络转载，感谢原创作者！
制作：红联Linux论坛
祝您阅读愉快！