第 5 章

RHEL 5 系统管理

上一章我们学习了 RedHat Enterprise Linux 5 操作系统的安装以及基础配置知识，本章要正式介绍该系统各主要方面的配置与管理。其中主要包括文件管理、用户和组账户管理、用户和组的文件访问权限配置与管理。

在 Linux 系统的文件管理中，有一个非常重要的环节就是应用程序的安装，本章以示例的方式介绍 rpm 格式和 tar、tar.gz 格式程序包的安装方法。这些都是最重要，也是最基本的 Linux 系统管理技能。在文件管理、账户管理和权限管理中，Linux 系统都是采用命令行方式进行配置与管理的，这样的命令在 Linux 系统中有上千个，而且每个命令又有许多可选项和参数，这给我们的管理工作带来不小困难。在学习过程中，我们首先不必要全面掌握各命令的全部选项和参数的使用方法，只需要掌握在日常的系统管理中经常用到的一些命令选项和参数，然后在这个基础上一步步拓展，只有这样才不会觉得很困难，才会增加自己学习好 Linux 系统的兴趣、勇气和信心。

教学（自学）课时安排

课时安排	本章老师共需安排 5 个授课课时。	
授课课时	主要内容	重点
1	①利用 RPM 管理器安装 rpm 程序包示例 ②利用 RPM 管理器删除程序包示例 ③利用 RPM 管理器更新程序包示例 ④校验已安装的 rpm 程序包	①利用 RPM 管理器安装 rpm 程序包示例 ②利用 RPM 管理器删除程序包示例 ③校验已安装的 rpm 程序包
2	①tar 命令使用示例 ②.tar.gz 程序包的解压 ③编辑程序安装配置文件 ④tar 程序包的编译和安装	①tar 命令使用示例 ②.tar.gz 程序包的解压 ③编辑程序安装配置文件 ④tar 程序包的编译和安装
3	ls、find、cat、more 和 grep 命令应用示例	ls、find、cat、more 和 grep 命令应用示例
4	①cp、mv、rm、mkidr、rmdir 和 mount 命令及应用示例 ②Linux 系统目录基础及目录切换 ③Linux 系统用户、用户及其配置文件	①cp、mv、rm、mkidr、rmdir 和 mount 命令及应用示例 ②Linux 系统用户、用户及其配置文件
5	①useradd/adduser、passwd、usermod、groupadd 和 groupmod 命令应用示例 ②Linux 系统文件访问权限 ③chmod 命令及应用示例 ④用户访问权限的数字表示方式	①useradd/adduser、passwd、usermod、groupadd 和 groupmod 命令应用示例 ②Linux 系统文件访问权限 ③chmod 命令及应用示例 ④用户访问权限的数字表示方式

5.1 rpm 程序包的安装

安装软件可以说是使用操作的第一步，也是非常重要的一步。可是，对于 Linux 初学者来说，安装一个很小的软件（如输入法软件）恐怕都是一件很让人头疼的事，因为在 Linux 下安装软件远不像在 Windows 中那样简单。在 Linux 中大多数软件提供的是源代码，而不是现成的可执行文件，这就要求用户根据自己系统的实际情况和自身的需要来配置、编译源程序后才能使用。多数初学者往往不知道该如何进行配置和编译就盲目地运行一些有执行属性的文件或者机械地运行 make、make install 之类的命令。结果搞得晕头转向，连个小软件都安装不了，不要说全面使用 Linux 系统了。这也导致了相当一部分用户选择继续使用容易操作和使用的 Windows 系统。

在 Linux 系统中安装程序包有多种方式（根据不同程序包类型），如 rpm 程序包可直接利用 RPM 管理器进行安装，tar 之类的程序包则需要先用 tar 之类的命令进行解压，然后进行配置文件编辑，用 make 命令编译，用 make install 命令进行安装。还有一种程序包安装工具就是 yum 命令，它可以自动解决 rpm 程序包在安装过程中的依赖性问题。本节和下两节将分别介绍 rpm 和 tar 工具在程序包安装中的应用，yum 命令因为比较复杂，在此不作介绍。

5.1.1 了解 rpm 程序包

目前流行的软件包有两种比较常见的形式：一种是 rpm 包的形式，另一种是压缩成*.tar.gz 的形式。RPM（RedHat Package Manager，RedHat 包管理器）只能在安装了 RPM 软件的 Linux 系统中使用，如 RedHat Linux、Fedora Linux、Mandriva、SuSE 和 Turbo Linux 等都可以使用它。而且在 RedHat Enterprise Linux 5 系统中，最简单地运行 rpm 程序的方法可以像 Windows 系统那样通过双击方式运行。但这种运行方式不带任何参数，完全采用默认的参数设置，所以一般情况下还是难以满足实际的程序管理需求。

在一个操作系统下，需要安装实现各种功能的软件包。这些软件包一般都有各自的程序，但是同时也有错综复杂的依赖关系。同时还需要解决软件包的版本，以及安装、配置、卸载的自动化问题。为了解决这些问题，RedHat 针对自己的系统提出了一个较好的办法来管理成百上千的软件，这就是 RPM 管理系统。在系统中安装了 RPM 管理系统以后，只要是符合 rpm 文件标准的打包程序都可以方便地安装、升级、卸载。

rpm 包的文件名中包含了这个软件包的版本信息、操作系统信息、硬件要求等。比如 httpd-2.2.3-6.el5.i386.rpm，其中 httpd 是在系统中登记的软件包的名字，2.2.3 是软件的版本号，6 是发行号，el5 表示用于 RedHat Enterprise Linux 5 操作系统，i386 表示用于 Intel x86 平台。

rpm 包包括 6 个基本的操作模式（不包括包的编译）：安装（-i 或 Install）、卸载（-e 或-erase）、升级（-u 或--upgrade）、查询（-q 或--query）、校验（-v 或-y 或--verify）和检验（-k 或--checksig）。本节将对它们进行一一介绍。要了解完整的细节和选项，可以使用 rpm --help 命令。

5.1.2 利用 RPM 管理器安装 rpm 程序包示例

rpm 程序包（结尾都是“.rpm”）的安装基本命令格式如下（注意命令本身及关键字

都必须是小写字母）：

rpm -i (or --install) [*options*] [*file1.rpm ... fileN.rpm*]

其中 file1.rpm ... fileN.rpm 可选项是指将要安装的一个或多个 rpm 程序包文件名。如果是同时安装多个 rpm 包，则各包之间要用空格分隔。

在使用 rpm --help 命令时输出了太多的可选项，但在平常的应用中有些是极少使用的。为了简单起见，下面仅介绍一些在实际系统管理中比较常用的参数选项。

1．专用选项

安装模式中专用的主要选项（options）如下：

【经验之谈】注意下面各参数选项前面的“-”或“--”，这是不能随便用的，否则执行命令时会出错。如果选项是用一个字母来简写，则只用一个小横杠（-），如果输入了选项的完整单词，则要用两个小横杠。多数参数是不能简写的，所以一定要用两个小横杠（--）来进行命令关键字连接，而且命令本身和参数选项都只能输入小写字母。

另外，使用命令方式安装 rpm 程序包时，要安装的 rpm 源程序包只能在当前用户主目录下，否则安装会出错。

- -h,--hash：安装时输出一连串的 hash 符号（#）。
- --test：只对安装进行测试（告诉是否可以安装），并不进行实际的安装。
- --percent：以百分比的形式输出安装的进度，这是默认的选项，即使不选也是这样输出的。
- --excludedocs：不安装软件包中的文档文件，默认是随程序一起安装的。
- --replacepkgs：重新安装已经安装的程序，同时保存原来相冲突的程序。这在重新安装某个 rpm 软件又怕删除错了时特别有用。
- --replacefiles：替换有冲突的文件。
- --force：忽略软件包及文件的冲突，强制安装。
- --noscripts：不运行包脚本程序。
- --prefix=<*dir*>：将软件包安装到指定的路径下。
- --ignorearch：不校验软件包的结构。
- --ignoreos：不检查软件包运行的操作系统。最好不要使用这个选项，否则即使安装了程序包，也不能使用。
- --nodeps：不检查依赖性关系。最好不要使用这个选项，否则即使安装了程序包，也不能使用，因为有许多程序是需要按照顺序来安装的，就像在上一节介绍的 SCIM 输入法平台安装一样。

2．通用选项

在 rpm 程序包的安装模式中除了可以用以上的专用选项外，还可以使用下面几个在其他模式中都可使用的通用选项（这些选项在所有 RPM 管理器工作模式中都可以使用）。

- -v：显示详细的安装输出信息。
- --version：显示所使用的 rpm 程序包版本信息。
- -r,--root=<*root*>：使指定的路径作为“根目录”（默认的根目录都是“/”），这样预安装程序和后安装程序都会按这个根目录定位安装路径。
- --nodigest：不校验 rpm 包的摘要信息。
- --nosignature：不校验 rpm 包的签名信息。

在进行 rpm 程序包安装时，通常所使用的参数选项有 3 个：-i（这是安装程序时必须

要使用的）、-v（显示详细的安装输出信息）和-h（以“#”符号显示安装进程）。多个参数时一般不是分别以连接符给出，而是以一个连接符，后面直接输入所有的参数名即可。如同时用以上 3 个参数选项，则直接输入-ivh 即可，而不用输入-i -v -h。

3．rpm 程序包安装示例

为了方便大家理解前面介绍的各参数选项的使用，现列举几个具体示例。

- 单 rpm 程序包安装

图 5-1 显示了使用以下命令安装 control-center-devel-2.16.0-14.el5.i386.rpm 程序包的输出。

```
rpm -ivh control-center-devel-2.16.0-14.el5.i386.rpm
```

图 5-1　单 rpm 程序包安装示例

【说明】此处是为了方便才把这些要安装的程序包都复制到了 root 用户主目录下，所以在安装 rpm 程序包时没有输入完整的路径。如果这些要安装的程序包文件不是在当前用户根目录下，则一定要指出它的完整路径。下同，不再赘述。

- 多 rpm 程序包安装

大家在 Linux 下安装 rpm 包的时候可能会发现，安装一个服务需要安装很多 rpm 包，比如在 RedHat Enterprise Linux 5 系统中安装 httpd 服务，它有许多关联的程序包要安装，但其中有一个是主程序包，如 httpd-2.2.3-6.el5.i386.rpm，其他的都依赖这个主程序包的安装，而且相互之间还可能有相互依赖关系（有关 httpd 进程完整的程序包安装将在本章介绍）。如果一个个地安装，很可能会在安装某个程序包过程中出现错误提示：必须先安装某个程序包。

为了避免之间的依赖关系问题，可以采取同时安装所有相关 rpm 包的方式来解决。因为在一个 rpm 命令中同时安装多个 rpm 程序包过程中无须顾及各程序包间的先后次序，rpm 管理器会自动按所需的顺序安装所有相关的程序包。这就方便了大型软件的安装。在这里仅以安装与 httpd 服务相关的几个 rpm 程序包为例进行介绍。httpd 服务所对应的主要 rpm 软件包在第二张光盘或 ISO 映象名的 server 目录下。现在把其中主要的 3 个文件复制到当前的 root 根目录下（这是为了方便安装，不用输入各程序包的完整路径，如果不复制到根目录下，则在执行命令时一定要输入各程序包的完整路径），如图 5-2 所示。

打开终端窗口（打开终端窗口的方法在前面已经多次介绍），输入以下命令（各程序包间以空格分隔，另外文件名是区分大小写的，不能随便输入），输出结果如图 5-3 所示：

```
rpm –ivh --nodeps httpd-2.2.3-6.el5.i386.rpm httpd-manual-2.2.3-6.el5.i386.rpm system-config-httpd-1.3.3.1-1.el5.noarch.rpm
```

【说明】因为这里仅介绍的是多个 rpm 包同时安装的方法，为了能安装成功，在上面加了--nodeps 选项，即不进行依赖关系检查。也正如此，从如图 5-3 所示的输出可以看出，sytem-config-httpd 和 httpd 这两个程序包并没有完全安装，没有显示 100%，这是因为还有与它们相依赖的程序包没有安装。有关服务进程或者应用软件的程序相关性检查方法将在本章后面介绍，它可通过 grep 命令进行。

图 5-2　httpd 进程所包含的 3 个主要程序包

图 5-3　多 rpm 程序包同时安装输出示例

- 利用-- replacepkgs 参数覆盖安装 rpm 程序

有时觉得某个程序包有问题，就可以使用 RPM 管理器中的--replacefiles 选项来强制覆盖已安装的相同程序包文件。

如要覆盖安装 httpd-2.2.3-6.el5.i386.rpm 程序包，则可输入以下命令：

```
rpm -ivh --replacepkgs httpd-2.2.3-6.el5.i386.rpm
```

- 利用--test 参数测试程序包的依赖关系

有时在安装 rpm 程序包时出现安装失败，提示对应程序包与某些程序相依赖。其实我们在事先就可以通过 rpm 管理安装模式下的--test 选项来测试对应软件与其他程序包的依赖性。如现在要测试一下在安装 httpd-2.2.3-6.el5.i386.rpm 程序包前必须先要安装哪些程序包，则可输入以下命令，结果会显示出该程序包所依赖的程序包有两个：libapr-1.so.0 和 libaprutil-1.so.0，如图 5-4 所示。

```
rpm -ivh --test httpd-2.2.3-6.el5.i386.rpm
```

图 5-4　使用--test 选项测试依赖性示例

5.1.3 利用 RPM 管理器删除（卸载）程序包示例

rpm 命令不仅可以用来安装 rpm 程序包，还可以从系统中卸载指定的程序包，其基本命令格式为：

rpm -e (or --erase) [*options*] [*pkg1 ... pkgN*]

与 rpm 程序包的安装一样，也可以同时删除一个或多个程序包，可选项 *pkg1 ... pkgN* 用来指定要删除的程序包或软件名，各程序包或软件名间也是用空格分隔。但是这里的程序包不要带“.rpm”扩展名，软件名如 httpd、vsftpd、samba 等。

在删除程序包时专用的选项有（其实它们与程序包安装的对应选项一样）：

- --test：只执行删除的测试。
- --noscripts：不运行包脚本程序。
- --nodeps：不检查依赖性。

除此之外，还可以使用上节介绍的通用选项。

下面以删除上节我们安装的 httpd-manual-2.2.3-6.el5.i386.rpm 程序包为例进行介绍。为了避免程序包间的依赖性检查，导致程序包删除不成功，也加上--nodeps 选项，进行强制删除。在终端窗口中输入以下命令（注意不要包括“.rpm”扩展名）：

rpm -e --nodeps httpd-manual-2.2.3-6.el5.i386

删除成功后会返回到终端提示符下，如图 5-5 所示。

图 5-5 删除程序包的输出示例

5.1.4 利用 RPM 管理器更新程序包示例

有时虽然系统中已安装了相同的 rpm 程序包，但是现在有了新版本的，需要对原程序进行更新，或者还可以更新到更旧的版本。这时也可以使用 RPM 管理器进行。

更新 rpm 程序的基本命令格式如下（注意下面的第一个 U 是大写的）：

rpm -U (or --upgrade) [*options*] [*file1.rpm ... fileN.rpm*]

可选项 *file1.rpm ... fileN.rpm* 也是表示可以一次性更新一个或多个 rpm 程序包，各程序间也是以空格进行分隔的。这时的程序包名必须带有.rpm 扩展名。升级 rpm 程序包的可用选项基本上与 rpm 程序包安装对应的选项是一样的（参见 5.1.2 节），但包括一个专用选项，即-oldpackage（强制更新到一个比当前版本更旧的程序包）。

5.1.5 利用 RPM 管理器查询程序包示例

RPM 管理器除了可以安装、更新和删除程序包，还可以查询已安装的 rpm 程序包信息，如了解系统中安装了哪些 rpm 程序包、每个已安装程序所对应的 rpm 程序包、查询具体程序包最终安装在系统中的哪个位置等。这就是 RPM 管理器的查询工作模式。

1．RPM 管理器查询模式命令格式

RPM 管理器查询模式的基本命令格式如下：

rpm -q (or --query) [*options*]　[*pkg1 ... pkgN*]

可选项 *pkg1 ... pkgN* 也是要查询的一个或多个程序包（可以是 rpm 程序包，也可以是不带.rpm 的程序文件，如 httpd 进程程序，要根据具体的应用而定），程序包或文件名间也是用空格分隔的。同样，如果要查询的程序包或文件不是在当前用户主目录下，则一定要指出对应程序包或文件的完整路径（路径都是“/”开始的）。

Rpm 程序查询专用的常用选项包括：

- -c, --configfiles：列出指定文件的配置文件。
- -d, --docfiles：列出指定文件的文档安装位置。
- --dump：清空基本文件信息。
- -l, --list：列出在包中所包含的文件及安装位置。
- -s, --state：显示列出文件的状态。
- -a, --all：查询所有程序包，可使用| more 参数分页显示，还可结合 grep 命令查询是否已安装了此程序包。
- -f, --file：查询指定的文件所属的程序包。
- -g, --group：查询组中所包含的程序包。
- -p, --package：查询一个包中所包含的文件。
- -i：显示程序包的概要信息。

同时也可使用 5.1.2 节介绍的通用可选项。

2．利用 RPM 管理器查询程序包示例

同样为了便于大家理解以上可用选项的具体用途，现列举几个示例进行介绍。

- 利用-qa | more 参数分页显示查询中已安装的程序包

如图 5-6 所示是使用 rpm -qa | more 命令查询系统中所安装的所有程序包，并且以分页显示方式输出。

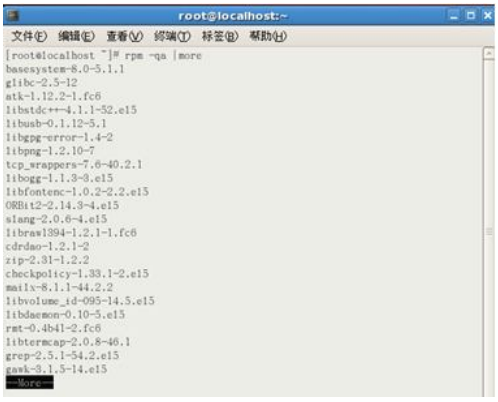

图 5-6　分页显示查询系统中所有已安装的程序包示例

- 利用-qa | grep 查询指定程序已安装的程序包

如图 5-7 所示是使用 rpm -qa | grep httpd 命令查询了 httpd 进程程序已安装的程序包仅 system-config-httpd-1.3.3.1-1.el5 这一个程序包。如果没有安装任何相关的程序包，则不会显示任何程序包。

图 5-7　查询指定程序已安装的程序包示例

- 使用-ql 参数列出指定包中所包含的所有文件

如图 5-8 所示是使用 rpm -ql system-config-httpd-1.3.3.1.el5 命令查询已安装的 system-config-httpd-1.3.3.1.el5 程序包中所包含的所有文件。在这里要注意的是，这里的程序包是指安装后的程序包，并不是指安装前的 rpm 包，所以只能取前面部分（不要包括.i386、.noarh 和.rpm 部分）。从中可以看出，通过这个程序包的安装，解压出许多文件，安装在不同的目录下。

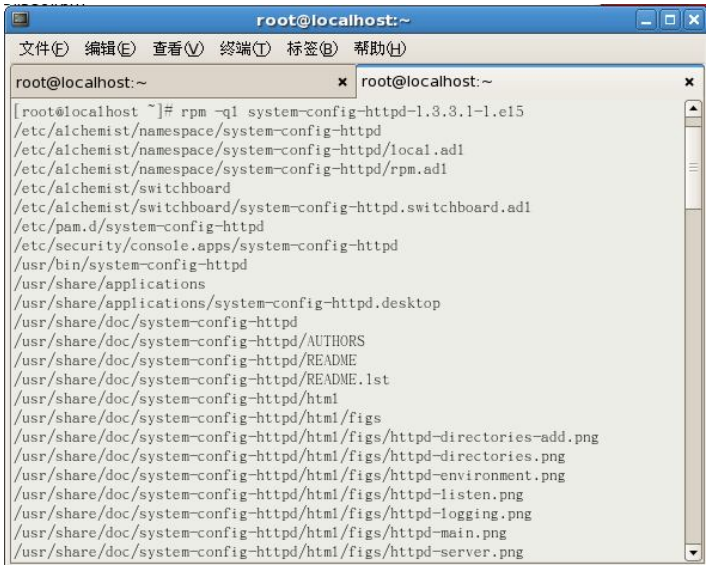

图 5-8　查询程序包中包括的文件及安装位置示例

- 利用-qf 参数查询指定文件所属的 rpm 程序包

如图 5-9 所示是使用 rpm -qf /usr/X11R6/bin/mkfontdir 命令查询/usr/X11R6/bin/mkfontdir 文件所属的程序包为 xorg-x11-font-untils-7.1-2 程序包。

图 5-9　查询文件所属程序包示例

- 利用-qpl 参数查询 rpm 包中包含的文件

如图 5-10 所示是使用 rpm -qpl httpd-2.2.3-6.el5.i386.rpm 查询已复制到当前用户根目录下（如果是在其他目录下则要输入完整的路径）的 httpd-2.2.3-6.el5.i386.rpm 程序中所包含的文件。从输出结果来看，它也包括了许多文件，并且给出了安装此程序包后各个文件的具体安装路径。

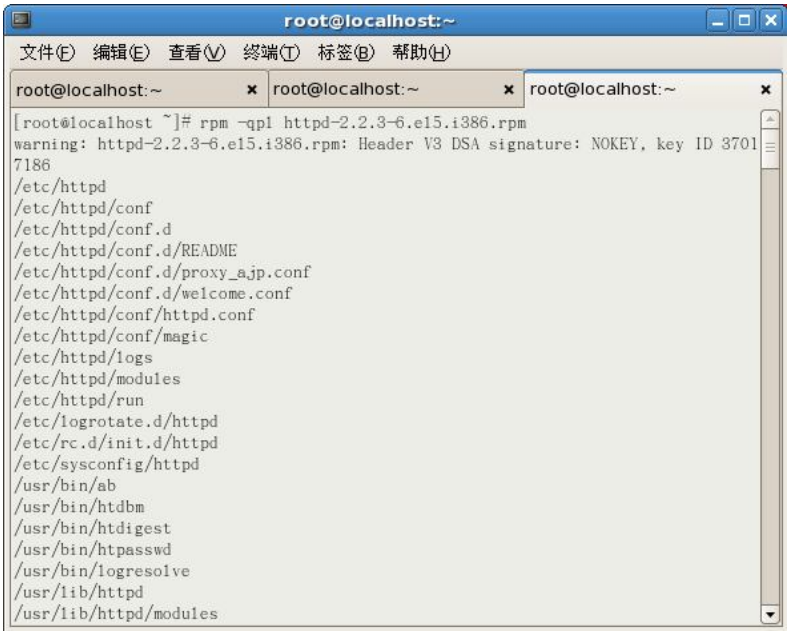

图 5-10　查询 rpm 包中包含的文件及安装后各文件的安装位置示例

5.1.6　校验已安装的 rpm 程序包示例

如果你担心 rpm 数据库已被破坏，则可以使用 RPM 管理器的校验工作模式。这种模式通常是用来检查某程序安装的完整性。如果一切均校验正常将不会产生任何输出。如果有不一致的地方，则会显示出来。这一点就远比 Windows 系统麻烦了。在 Windows 系统中运行一个安装程序，就会安装你所选的全部程序，而不必考虑这个应用程序到底要安装哪些程序文件。

1．RPM 管理器校验模式基本命令格式

利用 RPM 管理器检验已安装的程序包的基本命令格式如下（注意，第一个 V 是大写的）：

rpm -V (or --verify, or -y) [*options*] [*pkg1 ... pkgN*]

可选项 *pkg1 ... pkgN* 也可以是完整的 rpm 程序包或软件包名，要根据具体的应用而定。程序包或软件包名间以空格分隔。它可用的选项基本上与上节介绍的利用 RPM 管理器查询程序包一样，只是不能使用-c、-d、--dump、-l 这些可选项，另外还具有以下专用可选项：

- --noscripts：不运行校验脚本。
- --nodeps：不校验依赖性。
- --nofiles：不校验文件属性。
- --nomd5：不校验文件中的 MD5 摘要。

2．利用 RPM 管理器校验程序包示例

如图 5-11 所示是使用 rpm -Vp httpd-2.2.3-6.el5.i386.rpm 命令校验系统中 httpd-2.2.3-6.el5.i386.rpm 程序包安装的完整性。结果显示了许多文件没有安装。

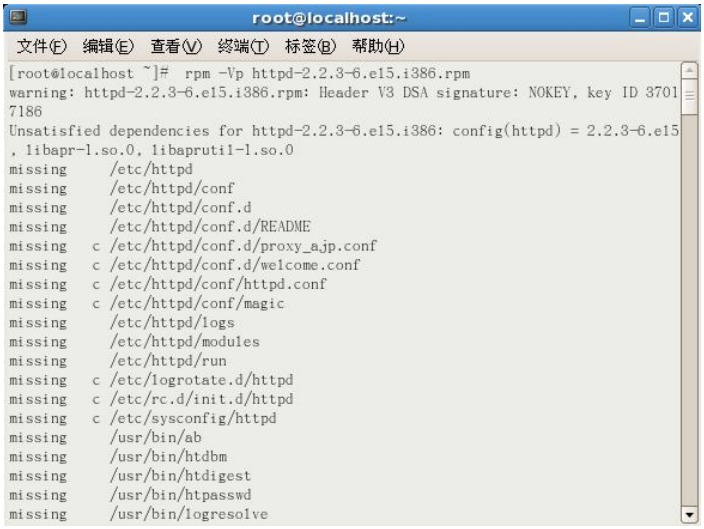

图 5-11　检查 rpm 程序包安装完整性的示例

5.1.7　其他 RPM 选项

在 RPM 管理器中，还有一些不是用于以上工作模式的，主要有以下选项：

- --rebuilddb：重建 RPM 数据库。一般在删除 RPM 程序包后，要使用这个选项来重建 RPM 数据库。
- --initdb：初始化 RPM 数据库。
- --help：显示 RPM 命令的帮助系统。
- --version：显示 RPM 管理器的当前版本。

5.2　编译、安装.tar.gz、.tar.bz2 程序包示例

在 Linux 系统中，由于发行版本非常多，所以不同发行版本所使用的程序包管理方式都不完全一样，这就造成了并不是所有 Linux 系统都使用上节介绍的 RPM 管理器，也不是所有程序包都是 rpm 格式。在 Linux 系统和 UNIX 系统中，普通使用的不是 rpm 包（特别是应用软件），而是像.tar.gz、.tar.bz2 这样的包格式。那么对于这类程序包，又如何安装呢？

5.2.1　tar 命令使用示例

tar.gz、tar.bz2 是源代码包（是使用 tar 命令打包，使用 gzip 或 bzip2 命令压缩的文件），需要编译之后才能安装，所以整个安装过程比较麻烦。在编译过程中可以指定各种参数以适应你的系统需求，如安装位置、优化参数、要哪些功能不要哪些功能等。

这类源代码类程序包的安装首先需要解压（tar.gz 类包用 tar zxvf 命令解压，tar.bz2 类包用 tar jxvf 命令解压，如果是 tar.tar 类包则解压时不要加 z 选项）。解压后一般都有一个 README、INSTALL 文件或 DOCS 目录，里面一般都是安装的详细说明，可以用 vi、nano、pico 或 Xwindows 下面的文本编辑器（如 gedit、gvim、kedit 等）打开查看。

这类程序包的安装一般是要经过 4 步：①用 tar zxvf 或 tar jxvf 命令解压程序包；②运行解压包中的 configure 脚本程序，配置安装前的参数，如程序安装位置和计算机环境（此

步可选，但若不配置，程序包会以默认方式安装，如程序包的执行文件会安装在/usr/local/bin 目录下，而资源文件会安装在/usr/local/share 目录下）；③运行 make 命令对程序包进行编译；④运行 make install 进行正式的程序安装。

1．tar 命令格式和可选项

下面先来了解 tar 命令的一些基本命令格式和可选项参数。

tar 命令在所有 Linux 系统中都是内置的，所以无须另外下载。在终端窗口中输入 tar--help 命令即可查看 tar 命令的语法格式和所有可用的参数，如图 5-12 所示。

图 5-12　tar 命令语法格式和参数

从中可以看出，它可用的命令参数相当多，但我们一般只需要记住一些主要的选项。以下是 tar 命令的几个操作模式命令，就像前面介绍的 rpm 命令有安装（-i）、删除（-e）、更新（-U）、校验（-V）等操作模式一样：

- -A, --catenate, --concatenate：把 tar 文件与另一个档案文件（可以是压缩的，也可以是不压缩的）连接。
- -c, --create：创建新的档案文件。
- -d, --diff, --compare：比较档案文件与文件系统中同名文件的不同。
- --delete：从档案文件中删除文件。
- -r, --append：把文件附加在指定档案文件的后面。
- -t, --list：列出档案文件所包含的文件内容。
- -u, --update：仅附加比档案文件中更新的副本。
- -x, --extract, --get：从档案文件中解压文件。

在解压应用中使用的可选项就是-x。

下面再介绍一些与档案文件解压应用相关的主要可选项：

- -f：使用档案文件或设备，这个选项通常是必选的。
- -k：在解压时不替换已经存在的文件。
- -j, --bzip2：使用 bzip2 压缩/解压文件。
- -v：详细报告 tar 处理的文件信息。如无此选项，tar 不报告文件信息。
- -z,--gzip, --gunzip, --ungzip：用 gzip 来压缩/解压缩文件。

2．tar 命令应用示例

- 使用-tvf 选项列出 tar 文件中的内容

如图 5-13 所示是使用 tar -tvf Mplayer-1.0rc2.tar.bz2 命令列出 Mplayer-1.0rc2.tar.bz2 文件中的内容。

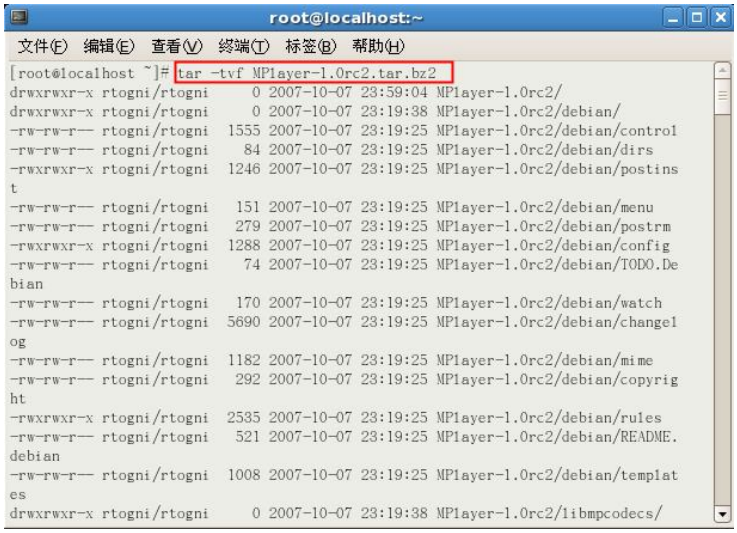

图 5-13　列出 tar 程序包中文件内容的示例

- 使用-cvf 选项备份指定目录

图 5-14 所示是使用 tar -cvf usr.tar /home 命令把/home 目录下的文件归档为 usr.tar 文件（但不压缩），最终在当前用户主目录下生成这个 usr.tar 文件，如图 5-15 所示。也可以自由指定新的档案文件保存的位置。但如果不是在当前用户主目录下，一定要包含完整路径，最前面是以根目录（/）开始的。

图 5-14　生成新 tar 文件示例

【说明】如果既要执行文件归档，又要压缩归档中的文件，则要使用-czvf 选项，并且文件名一定是 tar.gz 或 tar.bz2 格式，而不能仅是.tar，因为这里生成的归档文件是已压缩的。

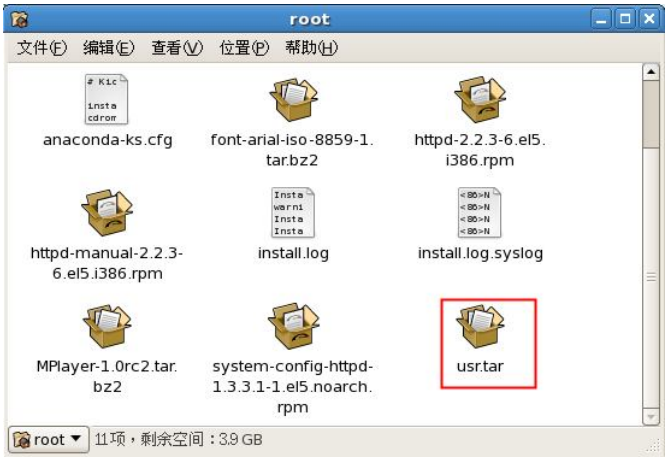

图 5-15　生成的新 tar 文件

- 使用-zxvf 选项解压压缩文件

如图 5-16 所示是使用 tar -zxvf usr.tar.gz 命令解压 usr.tar.gz 这个压缩文件。解压后就会在当前用户主目录下生成相应的目录，如图 5-17 所示。

图 5-16　解压 tar 程序包示例

图 5-17　解压后生成的新目录

5.2.2 .tar.gz 程序包的解压示例

下面以在 RedHat Enterprise Linux 5 系统中安装 Mplayer-1.0.rc2 媒体播放器为例介绍.tar.bz2 类程序包的安装，.tar.gz 程序包的安装基本一样。本节先介绍.tar.bz2 类程序包的解压方法。

先在网上下载安装所需软件（在此只取最基本的 3 个文件，分别是主程序软件 MPlayer-1.0rc2.tar.bz2、中文字体支持软件 font-arial-iso-8859-1.tar.bz2 和支持图形界面的皮肤文件 plastic-1.2.tar.bz2，至于其他像解码器之类的软件在此就不作介绍了）：

- 主程序 MPlayer-1.0rc2.tar.bz2 下载地址：http://www1.mplayerhq.hu/MPlayer/releases/。
- 中文字体支持程序 font-arial-iso-8859-1.tar.bz2 下载地址：http://www1.mplayerhq.hu/MPlayer/releases/fonts/。
- 皮肤程序 plastic-1.2.tar.bz2 下载地址：http://www.mplayerhq.hu/design7/dload.html。

然后把它们复制到 root 主目录中，按照前面的介绍在终端窗口中依次输入以下命令解压以上 3 个程序包：

```
tar -jxvf Mplayer-1.0rc2.tar.bz2
tar -jxvf font-arial-iso-8859-1.tar.bz2
tar -jxvf plastic-1.2.tar.bz2
```

结果会生成 3 个对应的目录，如图 5-18 所示。

图 5-18 解压后生成的 3 个新目录

5.2.3 编辑程序安装配置文件示例

通常在正式进行程序包安装前都是需要对安装参数进行配置的，配置时会自动生成一个 makefile 工具软件，但必须要安装了 2.96 版本以上的 gcc 程序包（在系统源程序包的第二张光盘或 ISO 映象包中有）。因为主 tar 程序包在解压后通常会包括一个名为 configue 的 shell 类脚本程序，通过它可以配置一些安装参数。而且这是个可执行程序，可以通过使用--help 选项列出它的帮助内容。

如本示例中的 Mplayer-1.0rc2 目录下就有这个 configure 可执行文件，如图 5-19 所示。打开终端窗口，通过 cd 命令进入到 Mplayer-1.0rc2 目录，然后键入./configure --help（相对

路径格式）或/root/configure --help（绝对路径格式）命令查看帮助内容（其实也可以通过文本编辑工具，如 vi、gedit，--help 选项也可以直接用-h 选项代替）。输出如图 5-20 所示。有关 Linux 系统中的相对路径与绝对路径将在本章后面介绍。

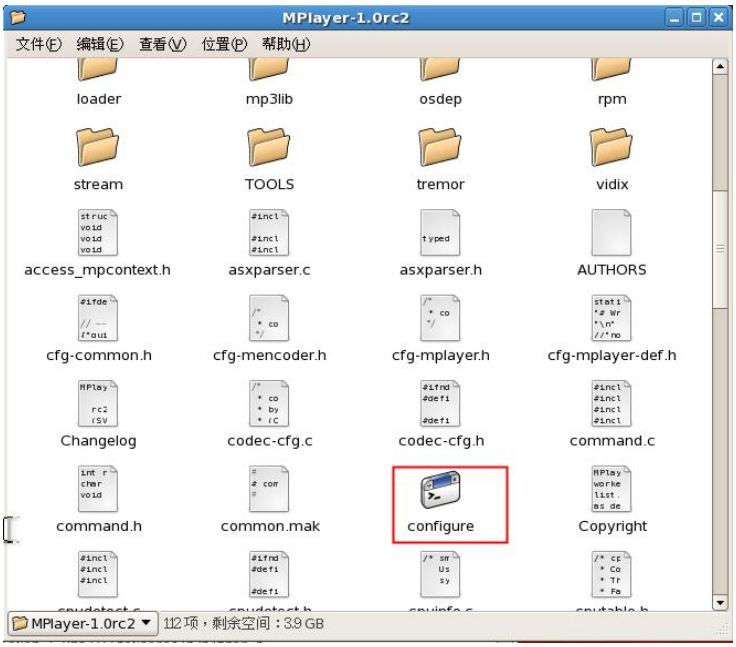

图 5-19　解压后生成的 configure 脚本程序文件

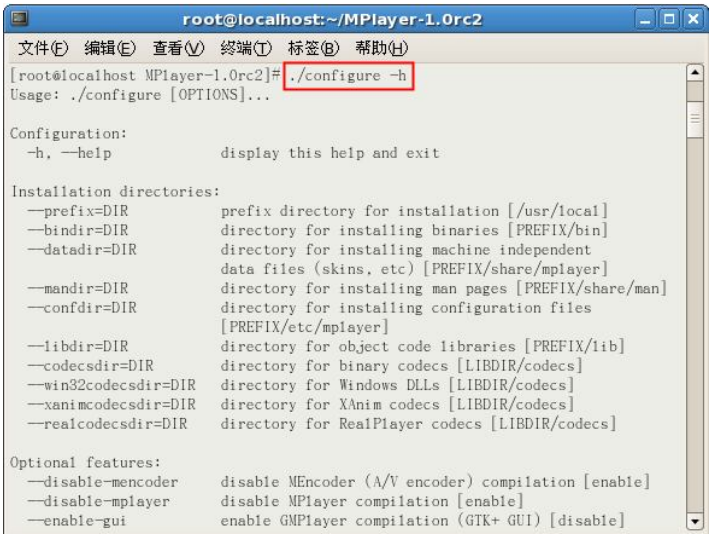

图 5-20　configure 程序的帮助内容

在这里我们要关注的是 Installion Directories（安装目录）这个项目下的选项。其中的--prefix=DIR 选项用来指定主程序的路径，默认是/usr/local 下；--bindir=DIR 用来指定程序安装过程中二进制文件的安装路径，默认是在主程序安装路径下的 bin 目录下安装；--datadir=DIR 用来指定独立程序文件（如皮肤程序）的安装路径，默认为主程序安装路径下 share 目录下的 mplayer 子目录下；--mandir=DIR 用来指定手册文件存放的路径，默认为主程序安装路径下 share 目录下的 man 子目录下；--confdir=DIR 用来指定安装配置文件的安装路径，默认为主程序安装路径下 etc 目录下的 mplayer 子目录下。下面那些是用来

指定各种解码器程序的安装路径的。

从以上路径可以看出，实际上我们只需要指定主程序的安装路径（也就是配置--refix=DIR 选项）就基本上可以确定其他程序的安装路径了，这与 Windows 程序的安装是一样的，而不必为下面的其他程序一一指定安装路径。

在这里最好还要配置一个所支持语言，它所使用的选项是--language=list，可以用空格或逗号分隔多个语言配置项。国内用户最常用的就两个 en（英语）和 zh_CN（中文简体），如果还要支持中文繁体，则还要添加 zh_TW 语言项。另外，为了顺利完成编辑配置，使用,--disable-gcc-checking 选项禁止 gcc（GNU Compiler Collection，GNU 编译器套装）版本检查（是一套由 GNU 开发的编程语言编译器。）

重新打开一个终端窗口，按顺序输入以下命令，配置 Mplayer-1.0rc2 主程序的安装路径为/usr，配置所支持语言包括英语、中文简体和中文繁体：

```
[root@localhost root]# cd MPlayer-1.0rc2
[root@localhost MPlayer-1.0rc2]# ./configure --enable-gui –prefix=/usr/local/mplayer --language=en --language=zh_CN --language=zh_TW
```

--enable-gui 是用于启用编辑配置文件的结果显示，如图 5-21 所示。

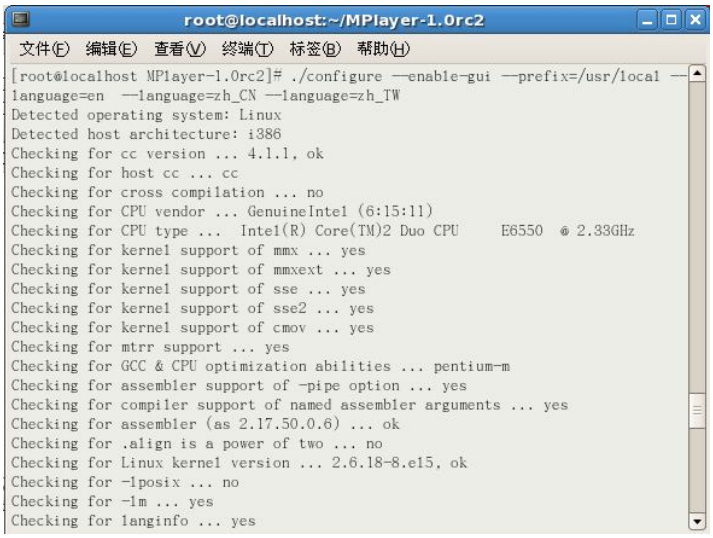

图 5-21　编辑配置文件的输出显示

如果系统中所使用的 gcc 版本不符合要求，会提示你要下载或更新 C 语言编辑器到 2.95 版本以上。如果 gcc 版本不符合，可在这里下载 2.96 版本以上的 gcc 程序包：http://download.chinaunix.net/download/0001000/69.shtml（其实在 RedHat Enterprise Linux 5 的第二张光盘或 ISO 文件中都有相应 rpm 格式的 gcc 程序包），再按前面的方法进行解压，再按照下节介绍的方法进行编译和安装，在此就不作具体介绍了。

5.2.4　tar 程序包的编译和安装示例

tar 程序在解压和配置好安装前参数后就可以进行最后的程序编译和安装了。编译的命令是 make，安装命令是 make install。configure 的目的是为了生成 makefile 文件，make 的作用是根据这个文件提供的系统参数编译包中的其他文件。make install 是把编译好的文件做成能用的二进制文件。

使用 make 命令对 tar 程序包进行编译是大多数源代码包都必须经过的一步。使用 make insatll 命令进行的 tar 程序包安装必须具有 root 权限（因为要向系统写入文件），前面的包

解压、安装参数配置和编译过程都可以是普通用户。

在本示例中的 Mplayer-1.0rc2 目录下依次输入 mak 和 make install 命令即可完成程序包的编译和安装。执行 make 命令时需要较长的时间。如图 5-22 所示就是执行最后的 make install 命令安装 Mplayer-1.0rc2 程序的进程。最后的主程序 mplayer 安装在/usr/bin 目录下，如图 5-23 所示。安装完毕后你就可以删除解压目录了。

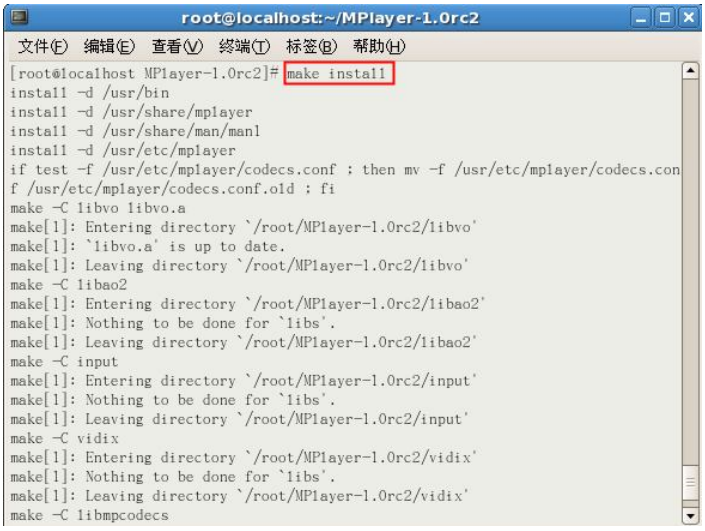

图 5-22　执行 make install 命令的进程

图 5-23　mplayer 播放器主程序

5.3　Linux 文件查看命令及应用示例

在 Linux 系统中进行文件的操作通常也都是在终端窗口下使用命令方式进行的，了解一些基本的文件操作命令的使用方法是非常必要的。因为 Linux 系统中的命令非常多，在此不可能一一介绍，仅介绍一些常用的文件操作命令。注意所有命令都可以通过--help 选项查看详细的参数选项。

在 Linux 系统中，用于文件查看的命令比较多，如 ls、find、file 和 cat 命令。当然它们的主要用途是不一样的。

【说明】在下面介绍的命令中有些选项既可以用一个横杠（-）连接（称之为“短选项”），又可以用两个横杠（--）连接（称之为“长选项”）。这时长选项中的参数要使用短选

项时也必须使用这些参数而不能仅使用短选项。以下适用于所有 Linux 命令，不再另行说明。

5.3.1 ls 命令及应用示例

ls 命令的作用是显示文件和目录列表，类似 DOS 下的 dir 命令，它的使用权限是所有用户。

ls 命令的基于本格式为：ls [选项][文件或目录名]

常用选项如下：

- -a, --all：不隐藏任何以“.”字符开始的项目，也就是会显示所有目录下的根目录名。
- -A,--almost-all：列出除了“.”及“..”以外的任何项目，不显示所有目录下的根目录名。
- --author：列出所列文件的所有者。
- -B,--ignore-backups：不列出任何以“~”字符结束的项目。
- -g：与-l 选项类似，但不列出文件的所有者。
- -G,--no-group：与-l 选项类似，但不列出组信息。
- -i,--inode：列出每个文件的索引号。
- -l：使用较长格式列出信息，也就是列出文件和目录的详细信息。
- -L,--dereference：当显示符号链接的文件信息时，显示符号链接所指示的对象，而并非符号链接本身的信息。
- -m：所有项目以逗号分隔，并填满整行行宽。
- -n,--numeric-uid-gid：类似-l 选项，但列出 UID 及 GID 号。
- -r, --reverse：逆序排序。
- -R,--recursive：同时列出所有子目录层。
- -s,--size：和-l 选项同时使用时以块为单位列出每个文件的大小。
- -S：根据文件大小排序。

如果直接在终端窗口提示符下输入 ls 命令，结果显示的是当前目录下所有的文件和目录列表，如图 5-24 所示。

图 5-24 直接使用无选项的 ls 命令输出示例

【经验之谈】在图 5-24 的列表中可以见到，不同的部分有不同的颜色显示，那么这是如何规定的呢？原来在 Linux 系统中规定，在输出列表中“蓝色”字代表的是目录，“绿色”字代表的是可执行文件，“红色”字代表的是压缩文件，“浅蓝色”字代表的是链接文件，“灰色”字代表的是其他文件。

如图 5-25 所示是使用-S 选项的 ls -S 命令按照由大到小顺序排列当前目录下的文件和目录的输出。与图 5-24 中没有使用-S 选项时的输出进行对比可以发现，两种列表方式的输出顺序是不一样的。

图 5-25　使用-S 选项时 ls 命令输出示例

如图 5-26 所示是使用-l 选项的 ls -l 命令列出文件和目录详细信息的 ls 命令输出示例。在这个输出中，列出了每个文件和目录中各用户和组的访问权限（有关文件和目录的访问权限将在本章后面具体介绍）。

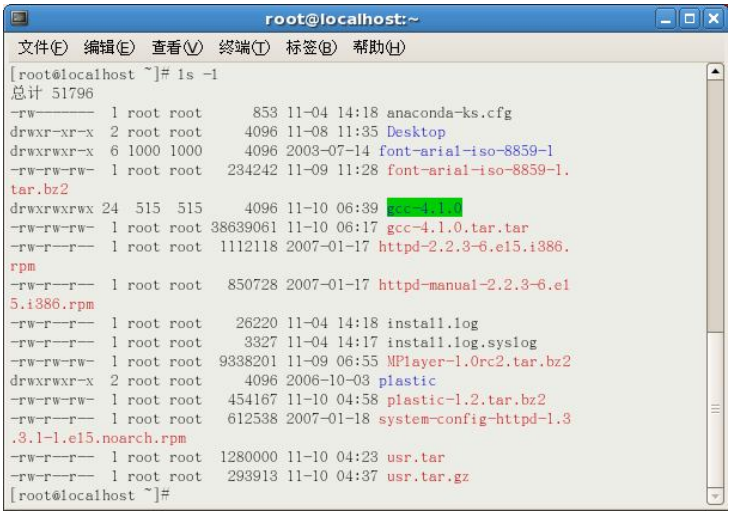

图 5-26　使用-l 选项详细列出文件和目录信息的示例

【经验之谈】在图 5-26 所示的输出列表的列中共分为了 7 个大的部分，前面那 10 个类似于 drwxr-xr-x 的小列项为对应文件目录的用户和组访问权限。后面那个数字，如 1、24 之类的代表所包括的文件数，如果是文件则为 1，如果是目录或压缩文件，则为对应目录或压缩文件中所包含的文件数。再后面那个账户，如 root（有的还是数字，如图 5-26 中的 gcc-4.1.0 压缩行）表示对应文件/目录或压缩文件的所有者。再后面那个账户就表示对应文件/目录或压缩文件所属的主要群。后面那个比较大的数字大家都可能知道是代表文件或目录的数量。后面那个日期代表的是文件或目录的最后修改日期和时间，最后一个就是对应的文件或目录名。

下面抽取图 5-26 中的第一行来说明，其结构如图 5-27 所示。

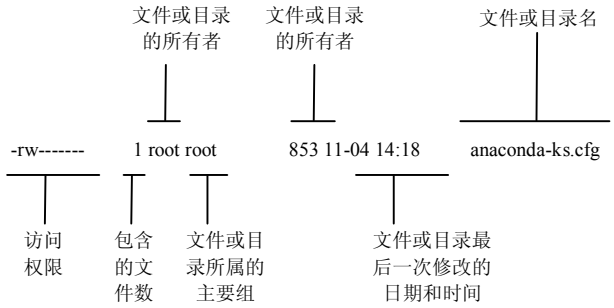

图 5-27　文件列表结构

如图 5-28 所示是使用-R 选项的 ls -R /home 命令列出/home 目录下各级子目录下的文件列表示例。在输出中依次列出了/home 目录及以下各级子目录的所有文件。

图 5-28　使用-R 选项列出各级目录下文件的示例

5.3.2　find 命令及应用示例

find 命令的作用是在目录中搜索文件，它的使用权限是所有用户，基本命令格式为：find [路径][选项][表达式]。

路径是指定查找的开始路径，系统从这里开始沿着目录树向下查找文件。它是一个路径列表，相互间用空格分离，如果不指定路径，则默认从当前目录（.）开始向下查找。在一个命令中可以指定多个查找路径，不同路径间用空格分隔，如 find /usr /home /tmp -name "*.jar"。

主要选项如下（注意，尽管下列选项都是缩写，但前面全只用一个小横杠连接）：

- -depth：使用深度级别的查找过程方式，在某层指定目录中优先查找文件内容。
- -maxdepth levels：表示至多查找到开始目录的第 level 层子目录。level 是一个非负数，如果 level 是 0 则表示仅在当前目录中查找。
- -mindepth *levels*：表示至少查找到开始目录的第 level 层子目录。
- -mount：不在其他文件系统（如 Msdos、Vfat 等）的目录和文件中查找。

表达式可以看做是一个查找条件，是 find 命令接受的表达式，find 命令的所有操作都是针对表达式的。它的参数非常多，这里只介绍以下一些常用的参数：

- -and：连接多个查找表达式，以实现混合查找。
- -name：查找指定文件名的文件（区分大小写），支持通配符*和?。如果包括通配

符，则文件名要用英文双引号括住。

- -iname：与-name 一样，只是不区分大小写。
- -atime *n*：搜索在过去 *n* 天读取过的文件。
- -ctime *n*：搜索在过去 *n* 天修改过的文件。
- -group *groupname*：搜索所有组为 *groupname* 的文件。
- -user *username*：搜索所有文件所有者为指定用户的文件。

如要从根目录开始查找一个文件名为 lilo.conf 的文件，可以使用如下命令：

```
find  / －name lilo.conf
```

如果只知道某个文件包含有 lio 这 3 个字母，想要在当前系统中查找所有包含有这 3 个字符的文件，则可以输入以下命令（输出如图 5-29 所示）：

```
find / -name "*lio*"
```

图 5-29　使用通配符查找文件的示例

如果要在/usr、/home 和/tmp 三个路径下（不仅这三个目录）查找最后为“.jar”的所有文件，则可以输入以下命令：

```
find /usr /home   /tmp -name "*.jar"
```

以下命令是在/usr/lib 目录下搜索在过去 2 天内读取过的文件：

```
find /usr/lib -atime 2
```

5.3.3　cat 命令及应用示例

cat 是一个简单而通用的命令，可以用它来显示文件内容（仅限于查看可编辑文档文件，但执行用户必须具有相应的文件访问权限）、创建文件，还可以用它来显示控制字符。cat 命令是将文件或屏幕输入组合输出到屏幕输出。如果“文件”选项缺省或者“文件”选项为“-”，则只读取屏幕输入；如果有“文件”选项，则可通过“>*文件名*”把当前屏幕上输入的内容输出到指定的文件中。要查看文件内容时，cat 命令是一次性把文件内容全部显示出来，所以最好把终端窗口设置得大些（设置方法参见上章的终端设置部分）。

其基本命令格式为：cat [选项] [文件]

cat 命令的主要选项有：

- -A,--show-all：等价于-vET 选项组合。
- -b,--number-nonblank：对非空输出行编号，空行不编号。
- -e：等价于-vE 选项组合。

- -E,--show-ends：在每行结束处显示$符号。
- -n,--number：对输出的所有行进行编号。
- -s,--squeeze-blank：不输出多行空行。
- -t：与-vT 选项组合等价。
- -T,--show-tabs：将跳格字符显示为^I。
- -n,--number：由 1 开始对所有输出的行数编号。
- -b,--number-nonblank：与-n 选项相似，只不过对于空白行不编号。
- -s,--squeeze-blank：遇到连续两行以上空白行时，就替换为一行的空白行。
- -v,--show-nonprinting：显示所有字符，包括非输入的控制字符（^）。

如图 5-30 所示是使用 cat /etc/passwd 命令查看 passwd 文件内容的输出。输入该命令后即相当于打开了 passwd 这个文件。如果要同时显示多个文件的内容，则各文件间用空格分隔即可，如 cat myfile1 myfile2 myfile3。

如果想把一个或多个文件中的内容合并，然后输出到一个新的文件（也可以是已存在的文件）中，可以使用以下命令（“>”符号与后面的文件名间可以有空格，也可以没有空格）：

```
cat myfile1 myfile2 myfile3 > myfile4
```

【经验之谈】在使用“>”单管道符时，如果输出的目标文件是已经存在的，则输出后目标文件中原来的内容将全部删除。这一点要特别注意。下面的示例同样要注意这个问题。

如果想不删除原文件中的内容，只是把新内容附加在原文件的后面，则要使用“>>”双管道符了。如上面的示例中，如果 myfile4 是已有内容的现有文件，现要把 myfile1、myfile2 和 myfile3 三个文件中的内容附加在这个 myfile4 文件现有内容的后面，则只需把上面示例中的“>”单管道符换成“>>”双管道符即可。

如果想用 cat 命令创建一个名为 myfile 的文本文件，其中的内容就是屏幕中所输入的内容，则可先在提示符下输入如下命令：

```
cat >myfile
```

然后在下面输入要在文件中输出的内容，完成时按 Ctrl+D 组合键结束输入，这时就会把所输入的内容输出到指定的新文件中，如图 5-30 所示。此时打开新创建的文件，即可发现，其内容与上面在屏幕中输入的内容完全一致，如图 5-31 所示。

图 5-30　使用 cat 命令输出当前屏幕输入的内容来创建新文件的示例

下面的 cat -n /etc/passwd >myfile 命令是把/etc/passwd 文件中的内容输出到新的或者已有的当前用户主目录下的 myfile 文件中（如果是目标文件已存在，则会清除原来的所有内容），并且给添加的内容每行从 1 开始加上行号。打开这个新创建的文件，即可见到每行都添加了行号，如图 5-32 所示。

cat 命令也可以用来制作映象文件，如制作启动软盘。例如要把软盘中的文件制作成映象文件，则可输入“cat /dev/fd0 > *映象文件*”命令。相反地，如果想把某映象文件写到软盘中，则要输入“cat *映象文件* > /dev/fd0”命令。

图 5-31　用 cat 命令查看文件内容的示例

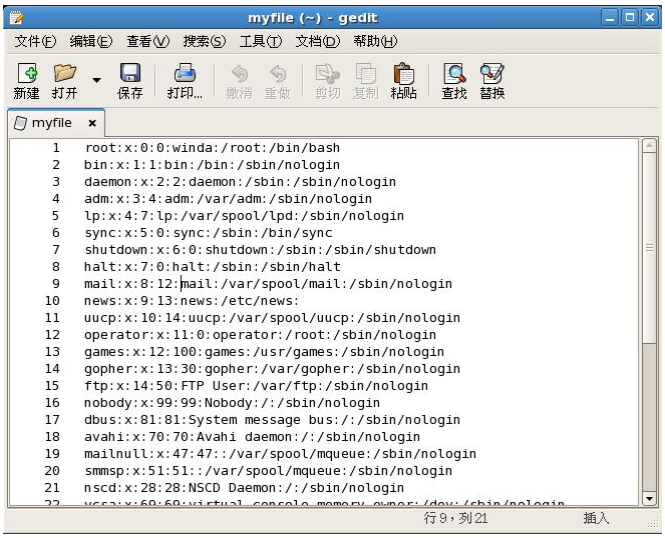

图 5-32　利用-n 选项给输出内容加上行号的 cat 命令示例

5.3.4　more 命令及应用示例

more 命令是一个配置显示方式或者查看文件内容（这方面与 cat 命令功能类似）的命令。其实 more 命令通常不是单独使用的命令，而是与 ls 之类的文件查看命令（但不能与 cat 结合使用）一起使用的，使超过一屏输出的内容按窗口大小或者按指定行数分页显示。在用来查看文件内容时，要查看的文件可以是一个或多个。在与 ls 等命令结构使用时要使用管道符“|”连接两个命令。

基本命令格式为：more [选项] [参数] 文件

常用的选项和参数如下：

- -d：在显示完一屏或一页时，会在下方显示---More--提示，按空格键继续下一屏或下一页的显示。
- -l：不处理控制符（如换页符）。
- -f：计算行数时，以实际的行数而非自动换行过后的行数（因为有些一行的内容在

一行中显示不完，会转到下行的，加了这个选项后仍以一行计算）。

- -p：不以卷动的方式显示每一页，而是先清除屏幕后再显示内容。
- -c：跟-p 选项相似，不同的是先显示内容再清除原有屏幕下面的内容。
- -s：当遇到有连续两行以上的空白行时，就替换为一行的空白行。
- -u：不显示下引号（根据环境变量 TERM 指定的 terminal 而有所不同）
- +/：在各文件显示前搜寻该字串，然后从该字串之后开始显示。
- +*num*：从指定的行处开始显示。
- -*num*：指定一次可以显示的行数。

如图 5-33 所示是使用 ls -R /root/gcc-4.1.0 | more 命令分页显示/root/gcc-4.1.0 目录下各层子目录中的文件列表。每一页的结束都会显示“--More--”提示，按空格键继续查看下一页内容。

图 5-33　more 命令与 ls 命令结合使用分页显示的示例

如果要分页查看某文件的内容，并且把所有连续两行或者以上的空行都换成一行空行，则可以使用 more 命令的-S 命令：

```
more -s /root/myfile
```

如果想指定一屏中只显示指定行数的内容，则可以使用-*num* 参数。如下所示命令就是使一屏只显示 10 行，按空格键继续后 10 行内容的显示，如图 5-34 所示。

```
more -10 /root/myfile
```

图 5-34　指定显示行数的 more 命令示例

如果上例要求是从第 22 行开始显示/root/myfile 文件的内容，则要输入以下命令（输出结果显示如图 5-35 所示）：

```
more +22 /root/myfile
```

图 5-35　从指定行开始显示的 more 命令示例

5.3.5　grep 命令及应用示例

grep 命令用于查找内容中包含指定的参数（又称正则表达式）样式的文件。如果发现某文件的内容符合所指定的参数样式，grep 命令会把含有参数样式的那一列显示出来。若不指定任何文件名称或是所给的文件名为“-”，则 grep 命令会从标准输入设备（如键盘、手写笔）读取数据。当表达式中含有特殊含义的字符（如$、*、[、|、^、(、)、\）时，必须带双引号。如果参数不是简单字符串，通常必须用单引号将整个模式括起来。如果只是普通字符，有没有引号（包括双引号和单引号）都可以

grep 命令的基本语法格式为：grep [选项]...参数 [文件] ...

常用的选项如下：

- -c, --count：只输出匹配行的计数。
- -e, --regexp=PATTERN：使用指定的式样作为查询规则。
- -i, --ignore-case：不区分大小写（只适用于单字符）。
- -h, --no-filename：查询多文件时不显示文件名。
- -l,--files-with-matches：查询多文件时只输出包含匹配字符的文件名。
- -n, --line-number：显示匹配的行和行号。
- -w, --word-regexp：强制只显示与整个单词匹配的行，也就是只搜索整个单词。
- -s, --no-messages：不显示不存在或无匹配文本的错误信息。
- -v, --invert-match：显示不包含匹配文本的所有行。

常用的输出控制选项及正则表达式参数如下：

- -m, --max-count=NUM：当搜索到指定的匹配数后停止搜索。
- -H, --with-filename：显示每个匹配的文件名。
- -o, --only-matching：仅显示行中匹配参数的部分。
- \：忽略正则表达式中特殊字符的原有含义。
- ^：匹配正则表达式的开始行。
- $: 匹配正则表达式的结束行。
- \<：从匹配正则表达式的行开始。
- \>：到匹配正则表达式的行结束。

- [ ]：单个字符，如[A]即 A 符合要求。
- [-]：范围，如[A～Z]，即 A、B、C 一直到 Z 都符合要求。
- .：可以包括所有单个字符，就像一个我们在 Windows 系统中代表单个字符的“?”通配符一样。
- *：所有字符，长度可以为 0。

如图 5-36 所示是使用 grep -n NSCD /root/myfile 或 grep -n "NSCD" /root/myfile 命令（如果要查找的只是普通字符，可以用引号也可以不用引号括住）查找/root/myfile 文件中包括 NSCD 字符串的行及行号。

图 5-36　使用 grep 命令查找/root/myfile 文件中包括 NSCD 字符串的行及行号示例

grep -n vi.e /root/config 命令可以用来查询前两个字符为“vi”，中间可以是任何字符，最后一个是“e”字母的字符串（.可代表任意一个字符）。查询输出如图 5-37 所示。

图 5-37　使用“.”替代单一字符的应用示例

使用 grep –n [lr]ive /root/config 命令查询 config 文件中凡是包含 live 或 rive 的行（[ ]可用来指定字符搜索的范围），输出结果如图 5-38 所示。

图 5-38　使用“[ ]”指定查询字符范围的应用示例

5.4 文件复制、移动和删除命令及应用示例

在操作系统中，复制、移动和删除文件（或目录）是经常要进行的一项基本文件管理工作，也是作为系统管理员所必须掌握的一项基本技能。本节介绍几个 Linux 系统中的基本文件复制（cp 命令）、移动（mv 命令）和删除（rm 命令）命令。

5.4.1 cp 命令及应用示例

该命令的功能是将指定的文件或目录拷贝到另一个文件或目录中，与 DOS 下的 copy 命令一样，功能十分强大。如果是目录，目标目录必须已经存在，否则复制不会成功。

基本语法格式：cp [选项] *源文件或目录 目标文件或目录*

该命令的主要选项如下：

- -a：用于保留文件或目录的链接、文档属性，并递归地拷贝目录，其作用等于后面将要介绍的-dpR 选项的组合。
- -d：复制时保留文件或目录的链接。
- -f：删除已经存在的目标文件而不提示。
- -i：与-f 选项相反，在覆盖目标文件之前将给出提示，要求用户确认。回答 y 时目标文件将被覆盖，是交互式复制模式。

【说明】为防止用户在不小心时覆盖了原来的同名文件，建议用户在使用 cp 命令拷贝文件时最好使用 i 选项。

- -p：加了此选项后，除复制源文件的内容外，还将把其修改时间和访问权限也复制到新文件中。
- -r,-R：加了此选项后，如源文件是一个目录文件，此时 cp 将递归复制该目录下所有的子目录和文件。此时目标文件必须为一个目录名。
- -l：不复制，只链接文件。
- -t,--target-directory=DIRECTORY：把所有源文件或目录复制到目标目录中。
- -T, --no-target-directory：视目标为普通文件。

cp 命令有以下几种用法：

cp [选项]... [-T] 源 目的
cp [选项]... 源... 目录
cp [选项]... -t 目录 源...

该命令将文件拷贝到/usr/wang 目录下，并改名为 shiyan1.c。若不希望重新命名，可以使用命令：$ cp exam1.c /usr/ wang/ $ cp - r /usr/xu/ /usr/liu/ 将/usr/xu 目录中的所有文件及其子目录拷贝到目录/usr/liu 中。

下面的命令是把当前用户主目录下的 exam1.c 整个目录复制到/usr/wang 目录下，并改名为 shiyan1.c，不提示覆盖：

```
cp - i exam1.c /usr/wang/shiyan1.c
```

如果复制后的目标文件名要保持与源文件名一样，则上述示例所对应的命令为（注意目录的最后必须是“/”，否则被视为文件）：

```
cp - i exam1.c /usr/wang/
```

下面是将/usr/wang 目录中的所有文件及其子目录拷贝到目录/usr/li 中：

```
cp - r /usr/wang/ /usr/li/
```

5.4.2 mv 命令及应用示例

可以使用 mv 命令来将<源>名称重命名为<目的地>名称，或将<源>文件移动至<目录>。该命令与 DOS 系统下的 ren 和 move 的组合功能类似。

基本语法格式：mv [选项] *源文件或目录 目标文件或目录*

如果目标是目录，mv 命令将文件重命名或将其移至一个新的目录中；当目标是文件时，mv 命令完成文件重命名。

其用法与 cp 命令类似，同样有 3 种：

mv [选项]... [-T] 源 目的

mv [选项]... 源... 目录

mv [选项]... -t 目录 源...

常用的选项如下：

- -i：以交互方式操作，如果移动操作时遇到同名的目标文件或目录，会提示用户回答 y 或 n，这样可以避免误覆盖文件。
- -f：禁止交互操作，覆盖不提示。
- -t,--target-directory=DIRECTORY：把所有源文件或目录移到目标目录中。
- -T, --no-target-directory：视目标为普通文件。
- -v, --verbose：详细显示进行的步骤。

下面的命令是将/usr/wang 中的所有文件（可以用通配符*表示，与 Windows 系统中一样）移到当前目录（用“.”表示）中：

```
mv /usr/wang/ *  .
```

下面的命令是将文件 test.txt 重命名为 text.txt：

```
mv test.txt    text.txt
```

5.4.3 rm 命令及应用示例

用户可以用 rm 命令删除不需要的文件。该命令的功能为删除一个目录中的一个或多个文件或目录，它也可以将某个目录及其下的所有文件及子目录均删除。对于链接文件，只是断开了链接，原文件保持不变。

rm 命令的一般形式为：

rm [选项] 文件…

常用的选项如下：

- -f：忽略不存在的文件，不给出提示。
- -i, --interactive：进行任何删除操作前必须先确认。
- -r, -R, --recursive：递归删除目录及其内容。如果没有使用-r 选项，则 rm 不会删除目录。
- -v, --verbose：详细显示进行的步骤。

【注意】使用 rm 命令要小心。因为一旦文件被删除，它是不能被恢复的。为了防止这种情况的发生，可以使用 i 选项来逐个确认要删除的文件。如果用户输入 y，文件将被删除；如果输入任何其他东西，文件则不会删除。

下面的命令是以互动方式删除 piglet.txt 文件：

```
rm -i piglet.txt
```

执行以上命令后，会询问是否删除文件 piglet.txt，只有按下 y 键后才执行删除操作。

还可以使用通配符“*”来删除文件，不过，必须谨慎而为，因为它很容易删除并不想删除的文件。如要删除当前目录下所有以 pig 开头的文件，则可以输入以下命令：

```
rm pig*
```

还可以使用 rm 命令来删除多个文件，多个文件间以空格分隔。例如可以使用以下命令同时删除当前目录下的 piglet.txt 和 readme.txt 两个文件：

```
rm piglet.txt readme.txt
```

还可以使用 rmdir 来删除目录，但是目录必须为空。可以通过-r 选项来强制删除目录。例如，如果想递归地删除当前目录 tigger，可以输入：

```
rm -r tigger
```

如果想组合选项，例如强制一种递归的删除（忽略不存在的文件），可以输入：

```
rm -rf tigger
```

5.5 目录创建、删除、切换和挂载命令及应用示例

在文件操作中，目录（或者称“文件夹”）的操作是必需的，如目录创建、删除和当前目录位置切换。在 Windows 系统中，目录的创建和删除操作很容易，直接在资源管理器中右击，就会有相应的快捷菜单。在 RedHat Enterprise Linux 5 系统中，其实也有这两种操作的快捷菜单，即“创建文件夹”和“移动到回收站”（创建空文件也有快捷菜单，即“创建文档”→“空文件”）。但是在 Linux 系统中，进行这类操作，远没有在 shell 提示符下操作功能强大，所以在此还是以在终端窗口下如何进行这些操作为例进行介绍。

5.5.1 Linux 系统目录基础及目录切换

在 Linux 系统中，像 Windows 系统一样也有表示根目录、当前目录和上级目录的符号，根目录为/，当前目录为.，上层目录为..，当前用户主目录为~。

切换目录的命令与 DOS 和 Windows 系统中一样，也是 cd 命令。如要进入到根目录，可以输入 cd /命令；如要退出上一层目录，可以输入 cd ..命令；如要退回到当前用户主目录，可以输入 cd ~命令。

在 Linux 系统中也有相对路径和绝对路径的概念，相对路径是从当前位置开始来描述路径的。如要进入到当前路径下的 progd 目录，则可键入 cd ./progd 命令。绝对路径是从根目录开始描述路径的，如/usr/share/mplayer。如果要用绝对路径进入到这个目录，则要输入 cd /usr/share/mplayer 命令。

5.5.2 mkidr 命令及应用示例

在 Linux 系统中，创建目录的命令是 mkdir。要求创建目录的用户在当前目录中具有写权限，并且目录名不能是当前目录中已有的目录或文件名称。mkdir 的基本命令格式如下：

mkdir [选项] 目录

常用的选项如下：

- -m, -m, --mode=模式：对新建目录设置存取权限，也可以用 chmod 命令（本章后面将介绍）设置。
- -p：新目录存放的路径名称。此时若路径中的某些目录尚不存在，加上此选项后，系统将自动建立好那些尚不存在的目录，即一次可以建立多个目录。

如要在当前目录下新建一个 program 的目录，并且只有所有者具有完全控制权限，则要输入以下命令（有关用户访问权限的配置将在本章后面介绍）：

```
mkdir -m 700 ./program/
```

如果新建的目录不是直接在当前目录下，而且在这个路径中还可能存在有不存在的目录，则可用-p 选项一同创建整个路径下的所有目录。如要在/usr/winda 路径下创建 doc 目录，而且 winda 目录也不存在，则可输入以下命令：

```
mkdir -p /usr/winda/doc
```

5.5.3 rmdir 命令及应用示例

rmdir 命令是用来删除空目录的，也就是所删除的目录必须是没有任何文件和目录的，并且你必须要有它的父目录的写权限。在本章前面介绍的 rm 命令，在使用-r 选项后可以删除非空目录，所以在实际的目录删除中使用更多的还是 rm 命令，而不是这里介绍的 rmdir 命令。

rmdir 命令的基本命令格式为：rmdir [选项]... 目录...

选项只有一个，那就是-p，它的作用是当要删除的目录下的子目录为空时，将连同这些空子目录也一起删除。如果不加这个选项，当下面还有空子目录时都不能删除所指定的目录。

如现在要删除/usr/winda/doc 路径中的 winda 目录，doc 子目录已为空，并且 winda 目录中没有其他文件或目录了，这时就可以输入以下命令连同 doc 子目录一起删除 winda 目录：

```
rmdir –p  /usr/winda
```

5.5.4 mount 命令及应用示例

在 Linux 系统中，有一个可以挂载目录或设备的命令，即 mount 命令。它可以将一个 Windows 分区（可以是 FAT、FAT32 和 NTFS 格式）挂载在一个已存在的目录上，这个目录可以不为空，但挂载后这个目录下以前的内容将不可用，从而将 Windows 的分区和/mnt 这个目录联系起来，因此我们只要访问这个文件夹，就相当于访问该分区了。从而实现了 Linux 系统与 Windows 系统之间的互访。它还可以将软驱、光盘和 U 盘这类设备作为 Linux 的一个“文件”挂接到 Linux 的一个空文件夹下，以实现对这些设备的访问。当然，事实上，现在的 Linux 系统自己已可以识别软驱和光盘了，一般不需要使用 mount 命令来挂载。

1．mount 命令格式和参数选项

mount 命令的基本格式为：mount [选项] [-t 文件系统类型] [-o 选项] [设备名称] [挂载点]

常用的参数和选项如下：

- -a：加载文件/etc/fstab 中设置的所有设备。除非标记为 noauto 或作了排除在外的。

- -h：显示在线帮助信息。
- -t <*文件系统类型*>：指定挂载设备的文件系统类型。目前常用设备的文件类型如下：
  - ext2/ext3：Linux 目前的常用文件系统。
  - msdos：MS-DOS 的 FAT 格式。
  - vfat：Windows 系统的 FAT32 格式。
  - nfs：网络文件系统。
  - ntfs：Windows NTFS 文件系统。
  - iso9660：CD-ROM 光盘的标准文件系统。
  - smbfs：Windows 网络共享文件。
  - auto：自动检测文件系统。
- -o <选项>：指定加载文件系统时的选项（有些选项也可以在/etc/fstab 中使用），主要包括：
  - loop：用来把一个文件当成硬盘分区挂接上系统。
  - ro：采用只读方式挂接设备。
  - rw：采用读写方式挂接设备。
  - iocharset：指定访问文件系统所用的字符集。
  - remount：重新加载设备，通常用于改变设备的设置状态。

可以用 umount 命令卸载所挂载的目录或设备。如用 umount /mnt/cdrom 命令可以卸载在/mnt/cdrom 上挂载的目录或设备。

2．mount 命令应用示例

下面是 mount 命令的一些具体使用示例。

- 挂载光驱到/mnt/cdrom 目录上

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```

- 挂载 Windows FAT 分区 hda3 到/mnt/vfat 目录上

```
mount -t vfat /dev/hda3 /mnt/fat
```

- 挂载 Windows NTFS 分区 hda6 到/mnt/ntfs 目录上

```
mount -t ntfs /dev/hda6 /mnt/ntfs
```

- 挂载软驱 fd0 到/mnt/floppy 目录上

```
mount /dev/fd0 /mnt/floppy
```

- 挂载 FAT32 格式 U 盘 sda1 到/mnt/cdrom 目录上

```
mount /dev/sda1 /mnt/cdrom
```

- 以只读方式挂载 Windows 系统计算机 lycb-w020 上的 share 共享文件夹到/mnt/winshare 目录下

```
mount -t smbfs -o ro //lycb-w020/share   /mnt/winshare
```

5.6 Linux 系统用户和组基础

RedHat Enterprise Linux 5 的用户和组群管理方法与它的前一版本 RedHat Enterprise Linux 4，甚至 RedHat Linux 9.0 几乎是完全一样的。所以，如果以前有这方面基础的话，本部分的学习就非常轻松了。下面先来总结一下 Linux 系统（适用于所有 Linux 系统）中

的用户和组群基本管理方法。

5.6.1 Linux 系统用户和组群管理概述

在 Linux 系统中，每个用户对应一个账号。而且在系统安装完成后，系统本身已创建了一些特殊用户（在 Windows 系统中我们称之为“内置用户”），它们具有特殊的意义，其中最重要的是超级用户，即 root 账户。

超级用户承担了系统管理的一切任务，可以不受限制地进行任何操作，因此建议只有在完全必要的情况下才以 root 身份进行操作。由超级用户来创建允许登录系统的普通用户（一般出于账户安全考虑，超级用户也需要为自己建立一个用来处理一般事务的普通账户）。

1．与 Linux 系统用户和组群管理有关的基本术语

在 Linux 系统中，下面是一些基本的用户和组群管理概念（其实与 Windows 系统的基本上是一样的）：

- 用户名

系统中用来标识用户的名称，可以是字母、数字组成的字符串，但区分大小写（与 Windows 系统中的用户一样，但 Windows 系统中的用户名不区分大小写）。

- 用户标识 UID

系统中用来标识用户的数字，相当于 Windows 系统的 SID。有关 Windows 系统的 SID 参见本系列教材中的《金牌网管师（初级）中小型企业网络组建、配置与管理》一书。

- 用户主目录

系统为每个用户配置的单独使用环境，即用户登录系统后最初所在的目录，这就是每个用户的主目录。用户文件默认都是存放在这个目录下。在 Windows 2000 Server 和 Windows Server 2003 等服务器系统中也有用途一样的“用户主文件夹”的概念，具体也请参见本系列教材中的《金牌网管师（初级）中小型企业网络组建、配置与管理》一书。

- 登录 shell

“登录 shell”简单地说就是指用户登录时所使用的一套系统外壳程序（其实不同 shell 是由不同人或组织开发的，相当于一个批处理程序），采用不同登录 shell 的用户在启动时所使用的 Startup（启动）程序是不一样的，如/bin/bash、/bin/csh、/bin/ksh、/bin/sh、/bin/tcsh、/bin/zsh、/sbin/nologin，默认是/bin/bash。其实登录 shell 与 Windows 系统中的配置文件功能类似。

- 用户组/组群

具有相似属性的多个用户被分配到一个组中，也与 Windows 系统中的组一样，但 Windows 系统中的组还有作用域和类型区分。在 Linux 系统中创建用户的同时可以创建与用户名一样的组，也可以不创建组，把它隶属于其他组中。这与 Windows 系统是一样的。

- 组标识 GID

用来表示用户组的数字标识。超级用户在系统中的 UID 和 GID 都是 0。普通用户的 UID 从 500 开始编号，并且默认属于与用户名同名的组。GID 也是从 500 开始编号。

2．用 su 命令改变身份

用户在系统使用过程中可以随时使用 su 命令来改变身份。例如，系统管理员在平时工

作时可以用普通账号登录，在需要进行系统维护时用 su 命令获得 root 权限，之后再用 su 回到原账号。

su 的基本语法为：su *username*

username 是要切换到的用户名（如果不指定用户名，则默认将用户身份切换为 root），切换时系统会要求给出相应用户的正确口令。

【经验之谈】默认情况下，只要知道 root 账户口令，任何用户都可以通过 su 命令切换到 root 身份。尽管 Windows 系统中的系统管理员账户的默认名都知道，但它可以改名，而 Linux 系统中的 root 账户不能改名，所以只需要知道密码即可拥有超级用户和管理权限。因此在 Linux 系统中增加了一个安全设置：可以通过编辑/etc/pam.d/su 文件使只有 wheel 组（是一个特殊组）成员（默认只有 root 账户是其成员，可以加入其他用户或组账户）才可以通过 su 命令转换为 root。如果换成一个不属于 wheel 组的用户时，执行了 su 命令后，即使输入了正确的 root 密码，也无法登录为 root 用户。实现的办法是修改，取消图 5-39 中如下语句的注释符（也就是去掉前面的“#”号）：

auth required pam_wheel.so use_uid

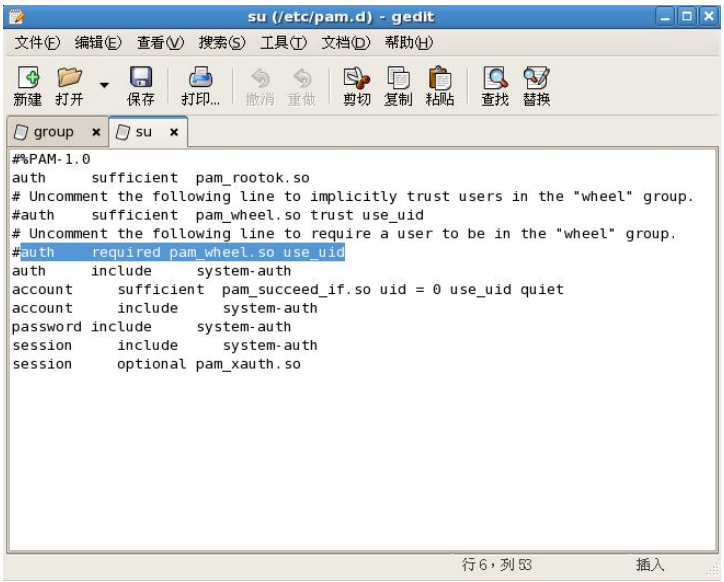

图 5-39　su 文件的内容

5.6.2　Linux 用户和用户配置文件

除了像 Windows 系统一样新建用户账户外，在 Linux 系统中同样有一些用户账户是在系统安装后就有的，就像 Windows 系统中的内置账户一样。它们是用来完成特定任务的，如 nobody 和 ftp 等，访问 LinuxSir.Org 的网页程序，就是 nobody 用户（相当于 Windows 系统中的匿名账户）；匿名访问 ftp 时，会用到用户 ftp 或 nobody。如果想了解 Linux 系统有哪些用户账号，可以通过/etc/passwd 文件来查看。可通过双击 gedit 文本工具软件打开，也可通过本章前面介绍的 cat 或 more 命令在终端窗口中查看。内置账户的 UID 都小于 500（其中 1～99 号为系统保留，分配给系统预定义账号），新建用户的 UID 都等于或大于 500，所以内置账户都在前面，很容易看出系统自动创建了哪些内置账户（在新启用服务进程后会创建一些新的内置账户，这与 Windows 服务器系统是一样的），如下所示（每行的开始处就是用户账户名）：

```
root:x:0:0:winda:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
avahi:x:70:70:Avahi daemon:/:/sbin/nologin
mailnull:x:47:47::/var/spool/mqueue:/sbin/nologin
smmsp:x:51:51::/var/spool/mqueue:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
haldaemon:x:68:68:HAL daemon:/:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
pcap:x:77:77::/var/arpwatch:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
rpm:x:37:37::/var/lib/rpm:/sbin/nologin
gdm:x:42:42::/var/gdm:/sbin/nologin
xfs:x:43:43:X Font Server:/etc/X11/fs:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
winda:x:500:500::/home/winda:/bin/bash
lycb:x:501:501::/home/lycb:/bin/bash
sally:x:590:590::/home/sally:/bin/bash
```

从中可以看出各用户对应的用途、所属组、登录 shell 等。

在 passwd 文件中包含了系统中所有用户的用户名和它们的相关信息，所有用户都可以查看，但只有 root 账户才有权限修改。该文件中每个用户账号在文件中对应一行，并且用冒号（:）分为 7 个部分（Linux 系统中称为“域”）。这 7 个部分的具体格式如下：

```
账户名:是否设置了口令:UID:所属组 GID:账户全名或描述:用户主目录:登录 shell
```

如上面的 root 用户账户行为：

```
root:x:0:0:winda:/root:/bin/bash
```

它表示 root 账户是有密码的（以 x 表示设置了密码，没有 x 的表示没有设置密码），用户 ID 和组 ID 号均为 0（内置账户的用户 ID 和组 ID 均小于 500，而新建的账户用户 ID 和组 ID 均等于或大于 500），账户全名为 winda（账户全名可以与账户名不一样），所用的登录 shell 为/bin/bash。

因为 passwd 文件对系统的所有用户都是可读的，这样的好处是每个用户都可以知道系统上有哪些用户，但缺点是其他用户的口令容易受到攻击（尤其当口令较简单时），所以在像红帽子和红旗等品牌的 Linux 中均使用影子口令格式，将用户的口令存储在另一个文件——/etc/shadow 中，该文件只有根用户 root 可读，因而大大提高了安全性。shadow 文件是前面介绍的 passwd 文件的补充，包括用户、被加密的密码和其他 passwd 不能包括的信息，如用户的有效期限等。在 shadow 文件中，每个用户对应一行，并且用冒号（:）分成 9 个部分，如下所示：

```
root:$1$xBZe1ZjG$zTc0Xtwb6.YR.d.2KCerT1:14552:0:99999:7:::
bin:*:14552:0:99999:7:::
daemon:*:14552:0:99999:7:::
adm:*:14552:0:99999:7:::
lp:*:14552:0:99999:7:::
sync:*:14552:0:99999:7:::
shutdown:*:14552:0:99999:7:::
halt:*:14552:0:99999:7:::
mail:*:14552:0:99999:7:::
news:*:14552:0:99999:7:::
uucp:*:14552:0:99999:7:::
operator:*:14552:0:99999:7:::
games:*:14552:0:99999:7:::
gopher:*:14552:0:99999:7:::
ftp:*:14552:0:99999:7:::
nobody:*:14552:0:99999:7:::
dbus:!!:14552:0:99999:7:::
avahi:!!:14552:0:99999:7:::
mailnull:!!:14552:0:99999:7:::
smmsp:!!:14552:0:99999:7:::
nscd:!!:14552:0:99999:7:::
vcsa:!!:14552:0:99999:7:::
haldaemon:!!:14552:0:99999:7:::
rpc:!!:14552:0:99999:7:::
rpcuser:!!:14552:0:99999:7:::
nfsnobody:!!:14552:0:99999:7:::
sshd:!!:14552:0:99999:7:::
pcap:!!:14552:0:99999:7:::
ntp:!!:14552:0:99999:7:::
rpm:!!:14552:0:99999:7:::
gdm:!!:14552:0:99999:7:::
xfs:!!:14552:0:99999:7:::
winda:CBQYuxVV/.Vqk:14552:0:99999:7:::
lycb:$1$8WXJXTkE$SDJCRFmLZd6iII8T7gUg81:14555:0:99999:7:::
sally:$1$saYZia0j$miwac/f4m99A1DZsfoz.s.:14555:0:99999:7:::
apache:!!:14556::::::
```

上面每一行的格式（从左到右）如下：

- 用户登录名。
- 用户加密后的口令：若为空，表示该用户不需要口令即可登录；若为*号，表示该账号被禁用，不能用来登录；如果为两个感叹号（!!），则表示该用户账户暂时被冻结了。
- 从 1970 年 1 月 1 日至口令最近一次被修改的天数。
- 口令在多少天内不能被用户修改，如果为 0，表示可以随时修改。
- 口令在多少天后必须被修改（也就是密码更改的最长有效期），默认为 99999 天。
- 口令在到期多少天内给用户发出警告。账号的密码失效期限快到时，系统依据这个字段的设定发出警告，提醒用户再过多少天密码将过期，请尽快重新设定密码。默认的是 7 天。
- 口令过期多少天后用户账号被禁止。默认为空，也就是没有设置。
- 用户过期日期，也就是指出该账号在此字段规定的日期之后将无法再使用。默认为空，也就是没有设置。
- 保留域。

同样以 root 账户为例，它在上面的行为：

root:1xBZe1ZjG$zTc0Xtwb6.YR.d.2KCerT1:14552:0:99999:7:::

对照上面的格式可以得出，它的用户登录名为 root，加密口令为 1xBZe1ZjG$zTc0Xtwb6.YR.d.2KCerT1（因为是加密的，所以显示的并不是直接的口

令），从 1970 年 1 月 1 日至口令最近一次被修改的天数为 14552 天，口令可随时修改，口令在 99999 天后必须被修改，口令过期 7 天后用户账号被禁止，后面的 3 个域没有配置。

5.6.3 Linux 组和组配置文件

用户组（group）就是具有相同特征的用户（user）的集合体，这很好理解，因为它与 Windwos 系统中的组是一样的。比如有时我们要让多个用户具有相同的权限，如查看、修改某一文件或执行某个命令，这时如果借助用户组把这些需要相同权限的用户都定义到同一用户组，通过修改文件或目录的权限，让用户组具有一定的操作权限，这样该用户组下的所有用户对该文件或目录都具有相同的权限。

管理用户组的基本文件是/etc/group，其中包含了系统中当前所有用户组（当然，在以后还可以新添加组）的相关信息，如下所示：

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon,winda
tty:x:5:
disk:x:6:root
lp:x:7:daemon,lp
mem:x:8:
kmem:x:9:
wheel:x:10:root
mail:x:12:mail
news:x:13:news
uucp:x:14:uucp
man:x:15:
games:x:20:
gopher:x:30:
dip:x:40:
ftp:x:50:
lock:x:54:
nobody:x:99:
users:x:100:
dbus:x:81:
utmp:x:22:
utempter:x:35:
avahi:x:70:
mailnull:x:47:
smmsp:x:51:
nscd:x:28:
floppy:x:19:
vcsa:x:69:
haldaemon:x:68:
rpc:x:32:
rpcuser:x:29:
nfsnobody:x:65534:
sshd:x:74:
pcap:x:77:
ntp:x:38:
slocate:x:21:
rpm:x:37:
gdm:x:42:
xfs:x:43:
winda:x:500:
lycb:x:501:
sally:x:590:
produ:x:591:
```

```
apache:x:48:
```

在这个文件中，同样每个用户组对应文件中的一行，并用冒号（:）分成 4 个域。其中每一行的格式为（在“组成员”域中各成员是以逗号分隔的）：

```
用户组名:是否设置了口令:GID:组成员
```

如上面 adm 组账户所在的行为：

```
adm:x:4:root,adm,daemon,winda
```

通过对照上面的基本格式可以得出，adm 组是设置了口令的，GID 为 4（内置组账户的 ID 都小于 500，新建组的组 ID 均等于或大于 500，如上面的 sally 组），这个组中包括 4 个用户成员，分别是 root、adm（管理员组，类似于 Windows 系统中的 administrators 组）、daemon 和 winda。

同样，负责保管组口令的也有一个影子文件，即/etc/gshadow，它与前面介绍的用户口令影子文件类似，在此不再介绍。

5.7 Linux 系统的主要用户管理命令

在服务器系统中，用户的管理是最重要的管理工作。在 Linux 系统中，用户的管理主要还是通过命令方式进行的，因为这样更灵活、功能更强大。本节要向大家介绍 RedHat Enterprise Linux 5 系统中的一些主要用户管理命令。

在 Linux 下登录用户账户的管理是通过 utmp 和 wtmp 这两个工具来实现的。wtmp 还记录系统重启和系统状态变化的有关信息。所有与 utmp 和 wtmp 相关的数据都分别被保存在/var/run/utmp 和/var/log/wtmp 这两个文件中。这两个文件均归属于 root 用户所有，并且访问权限被设置为 644（具体权限设置在本章后面介绍），这些文件中的数据是加密过的。

5.7.1 useradd/adduser 命令及应用示例

useradd 命令是 Linux 系统中最经常用来创建用户账户的命令。在 RedHat Enterprise Linux5 中，还有与它具有相同功能的命令——adduser。两者的功能是完全一样的。这两个命令只有 root 账户才可以使用。

1．命令格式和选项

useradd 命令的基本命令格式为：

useradd [选项] *账户名*

useradd 命令的主要可用选项如下：

- -c,-- comment *COMMENT*：为新用户账号添加说明。
- -d,--home-dir *HOME_DIR*：新用户账号创建指定的主目录。
- -e,--expiredate *EXPIRE_DATE*：设置新用户账户有效终止日期。日期的指定格式为 MM/DD/YY。
- -f,--inactive *INACTIVE*：设置新用户账号过期多久后便失效。当设置为 0 时账号则立刻被禁用，而当值为-1 时则关闭此功能，预设值为-1。
- -g,--gid *GROUP*：指定新用户账户所属的主要组。
- -G--groups *GROUPS*：设置新用户账户所隶属的组（不是组 ID）列表，每个组间用逗号（,）分隔。

- -m,--create-home：选择在创建新用户账户时自动创建同名的组账户。
- -k, --skel *SKEL_DIR*：加了这个选项后，系统会自动把/etc/skel 目录下的文件复制到新建用户的主目录中。

【说明】skel 目录非常有用，如果你是一个多用户系统的管理员，则可以在 skel 目录下写一个 readMe.txt 之类的使用说明文件，这样每个新建的用户都会在自己的目录下看到这个说明文件。另外，可以在/etc/skel 目录下自动创建一个.xinitrc 类用户配置文件，让所有用户都复制并使用这个配置文件。

- -M：在创建用户账户时不创建用户主目录。
- -r：此参数用来建立系统账号，当成系统服务账户。
- -s, --shell SHELL：指定新用户账户的登录 shell，默认登录 shell 为/bin/bash。
- -u, --uid *UID*：强制为新用户账户设置一个唯一的 UID（新用户账户 ID 在 500 以上）。
- -o, --non-unique：允许使用已使用的 UID。

2．应用示例

下面是 useradd 命令的一些具体应用示例。

- 不带选项的 useradd 命令

不带任何选项，直接输入 useradd 命令和用户账户名，即可创建一个具有默认属性设置（创建同名的用户组、使用/etc/bash 作为登录脚本创建用户主目录）的用户账户。如下命令创建的是一个具有默认属性设置的 winda 账户，这是最简单的创建用户账户的方法：

```
useradd winda
```

- 使用-d 和-G 参数分别为新创建用户账户指定主目录和隶属组

下面的命令是创建一个名为 heney 的用户账户，并指定它的主目录为/home/lycb，隶属于 adm 和 ftp 组：

```
useradd –d /home/lycb –G adm,ftp henny
```

- 使用-M 和-e 选项分别为新建用户账户指定不创建主目录和设置有效日期

下面的命令是创建一个名为 ruthy 的用户账户，并指定在创建账户的同时不创建主目录，账户有效期设为 2010 年 10 月 30 号。

```
useradd –M –e 10/30/2010 ruthy
```

- 使用-u 和-s 选项分别为新创建用户指定 UID 和登录脚本

下面的命令是创建一个名为 tina 的用户账户，并指定在创建账户的同时指定其 UID 为 800，登录脚本为/etc/sh。

```
useradd –u 800 –s /etc/sh tina
```

5.7.2 passwd 命令及应用示例

passwd 命令是 Linux 系统用来设置或修改用户账户密码的。所有用户都可以使用，但只能设置或修改自己账户的密码，并且多数选项只有 root 账户才有权使用。

passwd 命令的基本格式为：

passwd [选项] 用户账户

1．命令格式和选项

passwd 命令的主要选项如下：

- -l, --lock：锁定已经命名的账户名称，但只有 root 用户才能使用这个选项。
- -d ,--delete：删除账户的密码，也只有 root 用户才能使用此选项。
- -k, --keep-tokens：保持账户密码继续有效，所有用户都可以针对自己的账户密码进行该项设置。
- -i, --inactive=*DAYS*：设置在账户禁用后多少天密码失效，也只有 root 用户才能使用此选项。
- -n,--minimum=*DAYS*：设置账户的最短密码使用天数，也只有 root 用户才能使用此选项。
- -S,--status：检查指定使用者的密码状态，也只有 root 用户才能使用此选项。
- -w, --warning=*DAYS*：设置账户密码到期前发出警告的天数，也只有 root 用户才能使用此选项。
- -u, --unlock：解除账户的锁定状态，也只有 root 用户才能使用此选项。
- -x,--maximum=*DAYS*：设置账户密码的最长使用天数，也只有 root 用户才能使用此选项。
- 不带选项，只有 passwd 命令和用户账户，则可为该账户重新设置新密码，当然只有 root 账户才可以为其他账户设置密码。如果用户要为自己当前使用的账户设置新密码，可以直接输入 passwd 命令，按提示输入原密码、新密码，并重复输入一次新密码。

2．应用示例

下面是 passwd 命令的一些具体应用示例。

- 直接使用 passwd 命令修改当前用户密码

图 5-40 所示是 lycb 用户自己修改自己密码的过程，它分为 3 步：

（1）在终端窗口提示符下输入 passwd 命令，系统会提示输入当前密码（也就是原密码）。

（2）输入原密码，按 Enter 键，系统又提示输入新的密码。

（3）输入新密码后按 Enter 键，系统又提示重复输入一次新密码，按 Enter 键系统会提示密码更新成功。但要注意，新密码不能太简单，否则会出错的。

图 5-40　用户自己修改当前账户密码的示例

- root 账户修改其他用户账户密码

只有 root 账户才可以修改其他用户账户的密码。这时除了要使用 passwd 命令外，还需要指定要修改密码的用户账户名称。在这个修改过程中，root 用户不需要知道要修改账户的当前密码，也就是只需要两步，即输入新密码并重复一遍新密码。也就是 root 用户可以强制对对应用户账户的密码进行更新，使对方无法登录系统。修改成功时也会有提示的。

下面的命令是 root 用户修改 winda 账户的密码，其修改过程如图 5-41 所示。

```
passwd winda
```

图 5-41　root 用户修改其他用户账户密码的示例

- 使用-l 选项锁定指定账户（锁住后暂时该账户不能用）

下面的命令是使用-l 选项锁住 winda 账户，使这个账户暂时不能使用。锁定成功后会有提示的。

```
passwd –l winda
```

- 使用-u 选项解除指定账户的锁定状态（使原来被锁定的用户可用）

下面的命令是使用-u 选项解除 winda 账户的锁定状态，使这个账户可以使用。解除锁定成功后也会有提示的。

以上两个使用 passwd 命令锁定和解除锁定用户账户示例的过程如图 5-42 所示。

图 5-42　使用 passwd 命令锁定和解除锁定用户账户的示例

- 使用-n、-x 和-i 选项为指定用户设置密码最短使用时间、最长使用时间和账户禁用后密码失效的时间

下面的命令是设置 winda 用户账户的密码最短使用时间为 10 天，最长使用时间为 30 天，账户禁用 3 天后密码失效，配置过程如图 5-43 所示。

```
passwd –n 1 –x 30 –i 3 winda
```

图 5-43　使用 passwd 命令修改密码使用时间和失效期限的示例

5.7.3　usermod 命令及应用示例

usermod 可用来修改用户账号的各项设定，是一个功能很全面的用户账户设置命令。本命令只允许 root 账户使用。

1．基本格式和选项

usermod 命令的基本格式为：

usermod [选项] 用户账户

主要的选项如下：

- -c,--comment *COMMENT*：修改用户账户的说明。
- -d,--home *HOME_DIR*：修改用户主目录为指定目录。
- -e,--expiredate *EXPIRE_DATE*：设置账户的过期日期。
- -f,--inactive *INACTIVE*：设置账户过期后密码失效的天数。
- -g,--gid *GROUP*：强制账户使用指定组为主要组。
- -G,--groups *GROUPS*：指定账户隶属的组列表，组间用逗号分隔。
- -l,--login *NEW_LOGIN*：设置账户的新登录名。
- -L,--lock：锁定指定账户。
- -m,--move-home：移动主目录内容到新的位置（仅可与-d 选项一起使用）。
- -o,--non-unique：允许使用其他用户已使用的 UID。
- -p,--password *PASSWORD*：设置新的加密密码。
- -s,--shell *SHELL*：设置新的登录 shell。
- -u,--uid *UID*：设置新的 UID。
- -U,--unlock：解除账户被锁定状态。

2．应用示例

下面是 usermod 命令的一些应用示例。

- 使用-G 选项把用户加入到指定组中

以下命令可以把 lycb 用户加入到 winda 和 adm 组中：

```
usermod –G winda,adm lycb
```

- 使用-l 选项修改用户账户的登录名

以下命令可以把 lycb 账户的登录名改为 lycb-gz：

```
usermod –l lycb-gz lycb
```

- 使用-p 和-s 选项修改用户账户密码和登录 shell

以下命令是把 winda 账户的密码修改为 xyZJs，并把其登录 shell 修改为/etc/csh：

```
usermod –p xyZJs –s /etc/csh winda
```

5.7.4 其他用户管理命令

在 Linux 系统中，除了以上介绍的这些用户管理命令外，常用的还有以下几个：

- pwcov：从/etc/passwd 文件同步账户信息到/etc/shadow 文件。
- pwck：校验用户配置文件/etc/passwd 和/etc/shadow 的文件内容是否合法或完整。
- pwunconv：是 pwcov 的逆向操作，是从/etc/shadow 和/etc/passwd 创建/etc/passwd，然后会删除/etc/shadow 文件。这主要是在 paawd 文件损坏时使用。
- finger：查看用户信息工具。
- id：查看用户的 UID、GID 及所归属的用户组。
- su：用户切换工具，参见 5.6.1 节。
- sudo：通过另一个用户来执行命令。

5.8 Linux 系统主要的组管理命令

在 Linux 系统中，也有专门针对组的管理命令，主要有以下几个：

- groupadd：添加用户组。
- groupdel：删除用户组。
- groupmod：修改用户组信息。
- groups：显示用户所属的用户组。
- grpck：验证口令文件和组文件的完整性。
- grpconv：通过/etc/group 和/etc/gshadow 的文件内容来同步或创建/etc/gshadow，如果/etc/gshadow 不存在则创建。
- grpunconv：通过/etc/group 和/etc/gshadow 文件内容来同步或创建/etc/group，然后删除 gshadow 文件。

下面具体介绍 groupadd 和 groupmod 这两个相对比较复杂的组管理命令。

5.8.1 groupadd 命令及应用示例

groupadd 命令是用来在系统中添加组账户的。与前面介绍的 useradd 命令相对应（useradd 命令是添加用户账户的）。

1．基本命令格式和选项

groupadd 命令的基本命令格式为：
groupadd [选项] 组账户
主要的选项如下：

- -f,--force：若组群已存在，退出并显示错误（组群不会被改变）。如果指定了-g 和-f 选项，而组群已存在，-g 选项就会被忽略。
- -r：创建 GID 小于 500 的系统组账户。
- -g *gid*：使用指定的 GID 创建新的组账户。
- -K,--key KEY=*VALUE*：使用指定值覆盖/etc/login.defs 中的默认值。
- -o,--non-unique：允许使用已使用的 GID 创建组账户。

2．应用示例

- 使用不带选项的 groupadd 命令创建组账户

以下命令可以创建 sales 组：

```
groupadd sales
```

- 使用-g 选项创建指定 GID 的组

以下命令可以创建 GID 为 510 的 produ 组：

```
Groupadd –g 510 produ
```

5.8.2 groupmod 命令及应用示例

groupmod 命令用于修改组的 GID 或组名。

1．基本命令格式和选项

groupmod 命令的基本命令格式为：
groupmod [选项] 组账户
主要的选项如下：

- -g *GID*：修改组的 GID。修改 GID 后，可能会造成原本属于该组的文件现在却不属于该组了，所以需要你手动修改这些文件的所属组。
- -n *new_group_name*：修改组账户名称。
- -o *GID*：强制使用已使用的 GID。

2．应用示例

以下命令可以修改 winda 组账户名称为 lycb（注意新组账户名称在前，原组账户名称在后）。

```
groupmod –n lycb winda
```

5.9 Linux 系统用户和组对文件访问权限配置命令

在 Linux 系统中，用户和组对目录访问权限的配置都是通过命令方式进行的，而不会像 Windows 系统那样配置方便。用于配置用户或组账户对文件和目录访问权限的命令主要有以下几个：

- chmod：设置用户的文件操作权限。
- chown：改变文件所有者。
- umask：设置权限掩码（决定新建文件的权限）。

5.9.1 Linux 系统文件访问权限查看示例

在 Linux 系统中，用户/组对文件和目录的访问权限规定与 Windows 系统中完全不同，它是由一系列的符号和控制符来表示的，显得更加抽象。本节先来了解在 Linux 系统中文件和目录访问权限的描述。

在 Linux 系统中，在文件和目录访问权限中，首先它把访问的用户分成了以下 4 类，并各用一个字符表示：

- u：表示“用户（user）”，即文件或目录的所有者。
- g：表示“同组（group）用户”，即与文件属主有相同组 ID 的所有用户。
- o：表示“其他（others）用户”。
- a：表示“所有（all）用户”，它是系统默认值。

在 Linux 系统中，规定的文件和目录访问权限的分类也比较粗，仅分为以下 3 类：

- r：读取权。
- w：写入权。
- x：执行权（只有可执行文件和目录才能为用户或组分配可执行权）。

【经验之谈】无论从配置访问权限的用户类型（仅 4 类，无法对具体的用户和组进行配置），还是从权限的分类（仅 3 类）都可以看出，Linux 系统下，远没有 Windows 系统下用户对文件和目录的访问权限规定得那么细，所以配置起来比较难以满足实际权限管理的需求。

在前面介绍的 3 类基本的访问权限中，用户和组可以根据需要任意组合它们得出各种最终的访问权限。其中 r--表示只读，rw-表示可读、可写，rwx 表示可读、可写和可执行，-w-表示可写，-wx 表示可写和可执行，--x 表示可执行，r-x 表示可读和可执行，---表示无访问权限。

在使用 ls -l 命令时就可以见到各文件的用户和组对该文件的访问权限，如图 5-44

所示（在第一列中显示，有关文件列表结构在本章前面介绍 ls 命令时就已有介绍，参见图 5-27）。

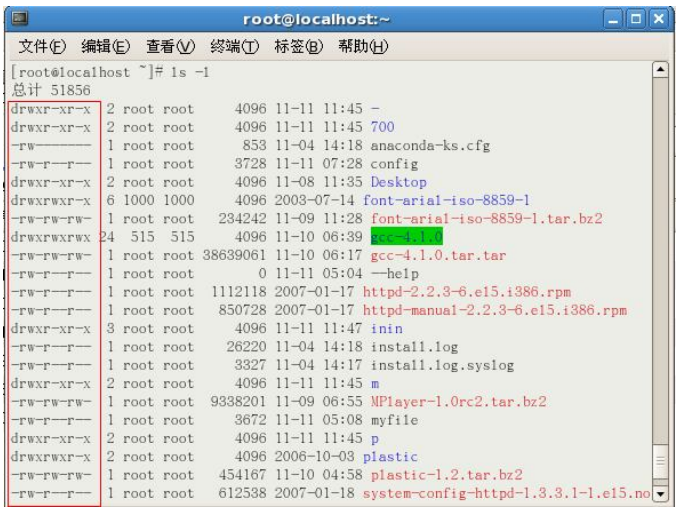

图 5-44　使用 ls-l 命令查看文件和目录列表时显示的访问权限

从上面的第一例可以看出，文件和目录的访问权限部分一共是有 10 项。Linux 系统规定：从左边算起，第一位用来表示当前是文件还是目录，如果是文件，则以“-”表示，如果是目录，则以“-d”表示，所以这一位其实不能算是权限列。第 2～4 位是文件或目录的所有者对它的访问权限，第 5～7 位是与文件或目录所有者属于同一组群中的其他用户对它们的访问权限，最后三位，也就是第 8～10 位是其他用户对它们的访问权限。

如下面是图 5-44 中的一行：

drwxr-xr-x　3 root root　　4096 11-11 11:47 inin

从这条的最前面那个 d 字符可以看出，它是一个目录，从它后面的两个 root 可以知道，它的所有者用户账户和主要群账户都是 root。在权限列中 2～4 位表示 inin 目录的所有者 root 账户的访问权为 rwx，即是可读、可写和可执行（相当于完全控制权限）；在权限列的第 5～7 位表示 inin 目录主要组的 root 组对它的访问权为 r-x，表示可读和可执行，但不可写；在权限列中的最后 3 位表示的是其他用户和组对 inin 目录的访问权为 r-x，也是可读和可执行，但不可写。

下面再看一个例子。假设下面是 readme.txt 文件的初始权限设置：

-rw-rw-r--　　1 winda winda　　398 11 12:04 readme.txt

根据上面的介绍可以得出，这个是一个文件，因为权限列的第一位为“-”，readme.txt 文件的所有者（winda）和主要组（winda）对 readme.txt 文件的访问权限都是可读和可写（rw-），但不可执行（因为这是个文本文件，不能执行），其他用户和组对 readme.txt 文件的访问权限都是只读权限（r--）。

5.9.2　chmod 命令及应用示例

使用 chmod 命令可以将每个命令中指定的文件的模式更改为设定的模式，通俗地讲就是改变用户对指定文件或目录的访问权限。下面这个例子显示了如何使用 chmod 命令来改变 readme.txt 文件的权限。

1．基本命令格式

chmod 命令的基本命令格式为：

chmod [操作对象][+|-|=][权限]文件名

上面的“操作对象”就是上节介绍的 Linux 用户类型，[+|-|=]选项的各操作符所代表的意义如下：

- +：添加某个权限。
- -：取消某个权限。
- =：赋予给定权限并取消其他所有权限（如果有的话）。

“权限”部分也是上节介绍的 3 种基本的访问权限类型。

2．应用示例

下面以一些具体的示例介绍 chmod 命令的应用方法。

- 添加权限

同样借用上面已介绍的 readme.txt 文件访问权限配置的示例，如下：

```
-rw-rw-r--      1 winda winda       398 11 12:04 readme.txt
```

从前面的权限列中可以知道，该文件的所有者 winda 和主要组 winda 对它的访问权限都是可读可写的，但其他人只是可读，不具有写入权限。如果想给每个人以写入 readme.txt 文件的权限，这时其实只需要改变其他人对该文件的访问权限部分。根据上节的介绍可以知道，其他人是用“o”表示的，而根据前面的介绍可知，添加权限的操作符是“+”，写权限为“w”。此时使用以下 chmod 命令就可以实现以上目的：

```
chmod o+w readme.txt
```

o+w 命令参数就可以为其他人分配具有写入 readme.txt 文件的权限。因为所有者和主要组账户原来就已具有写入权限了，所以现在所有人都对 readme.txt 文件具有写入权限了。此时再用 ls -l 命令列表时可以看到这个文件的用户访问权限就如下所示了（在第 3 列中多了一个 w）：

```
-rw-rw-rw-      1 winda winda       398 11 12:56 readme.txt
```

- 删除权限

如果要从 readme.txt 中删除主要组和其他人的读写权限，可以使用 chmod 命令来实现，具体命令如下（组账户用“g”表示，其他用户用“o”表示，取消权限用“-”表示，读写权限表示为 rw）：

```
chmod go -rw readme.txt
```

通过输入 go-rw 参数就可以使系统删除主要组群和其他人对 readme.txt 文件的读、写权限。再次通过 ls -l 命令列出的结果如下：

```
-rw-------      1 winda winda       398 11 13:011 readme.txt
```

- 删除所有人的访问权限

删除所有人的访问权限就要用到“a”这个用户类型了，它代表的是所有人。删除权限操作符仍是“-”，所有权限为 rwx。现假设从文件 readme.txt 中删除所有用户对它的权限，则可使用以下命令：

```
chmod a -rwx readme.txt
```

删除所有人对它的访问权限后，就没有任何人能够访问它了，包括文件或目录的所有者。如果使用 cat 命令来读取这个文件，它会返回一个权限被禁用的错误消息——

Permission denied。

但是仍可以由所有者使用 chmod 命令更改用户对它的访问权限，这时就要用到添加权限的操作符“+”了。

以下是几个可以用在 chmod 命令设置上的常用用户类型、操作符和权限组合的例子：

- g+w：为组群添加写入权限。
- o-rwx：删除其他人的所有权限。
- u+x：允许文件所有者执行这个文件。
- a+rw：允许每个人读取并写入文件。
- ug+r：允许所有者和组群读取文件。
- g=rx：只允许组群读取和执行（不能写入）。

5.9.3 用数字表示用户访问权限配置示例

上面介绍的是使用符号来表示权限的文件或目录访问权限的修改方法。在 Linux 系统中还有一种用数字表示权限的方法。它把 4 种基本权限用不同的数字来表示：读取权限（r）用数字 4 表示，写入权限（w）用数字 2 表示，执行权限（x）用数字 1 表示，无权限（-）用数字 0 表示。每一类用户的最终访问权限是由这 4 种基本权限的组合数值和表示的。如 r--（只读）可用 4 表示，rw-（可读、可写）可用 6 表示，rwx（可读、可写和可执行）可用 7 表示，-w-（可写）可用 2 表示，-wx（可写和可执行）可用 3 表示，--x（可执行）可用 1 表示，r-x（可读和可执行）可用 5 表示，---（无访问权限）可用 0 表示。

回到前面介绍的那个 readme.txt 文件的原始权限：

```
-rw-rw-r-- 1 winda winda 39 8 月  11 12:04 readme.txt
```

如果用数字来表示权限，则上面的权限列可以表示如下：

```
-       (rw-)          (rw-)          (r--)
          |              |              |
       4+2+0=6        4+2+0=6        4+0+0=4
```

所有者的总和为 6，组群的总和为 6，其他人的总和为 4，则这个权限设置读作 664。

如果想改变 readme.txt 文件的权限，使组群中的人没有写入权，但是仍旧能够读取文件，则这个文件的数字权限用的表示形式就是 644。只需要直接输入以下命令即可实现：

```
chmod 644 readme.txt
```

现在，如果再要使其他人不具有对 readme.txt 文件的访问权限，则只需要把其他用户的访问权限项 4 变为 0 即可，也就是最终的数字权限表示为 640。输入以下命令即可实现：

```
chmod 640 readme.txt
```

【说明】把权限设为 666 会允许每个人对文件或目录都有读取和写入的权限。把权限设为 777 允许每个人都有读取、写入和执行的权利。这些权限可能会允许对机密文件的篡改，因此，一般来说，使用这类设置是不明智的。

以下是某些常用命令的设置、数值以及它们的含义：

- -rw-------（600）：只有所有者才有读取和写入的权限。
- -rw-r--r--（644）：只有所有者才有读取和写入的权限，主要组和其他人只有读取的权限。
- -rwx------（700）：只有所有者才有读取、写入和执行的权限。
- -rwxr-xr-x（755）：所有者有读取、写入和执行的权限，主要组和其他人只有读取和执行的权限。

- -rwx--x--x（711）：所有者有读取、写入和执行的权限，主要组和其他人只有执行权限。
- -rw-rw-rw-（666）：每个人都有读取和写入文件。
- -rwxrwxrwx（777）：每个人都有读取、写入和执行的权限。

【说明】在 RedHat Enterprise Linux 5 系统中，创建用户和组也可以通过图形界面的方式进行。具体操作方法是，root 用户在桌面上执行“系统”→“管理”→“用户和组群”命令，打开如图 5-45 所示的窗口。在其中的“用户”选项卡中会显示当前已有的新建用户账户（不包括内置账户），在“组群”选项卡中会包括当前已有的新建组，也不包括内置组账户。单击工具栏中的“添加用户”按钮，打开如图 5-46 所示的界面。在这里可以配置新用户账户的基本属性、创建新的用户账户。在图 5-45 所示窗口的工具栏中单击“添加组群”按钮，会打开如图 5-47 所示的界面。在这里可以配置新组账户的基本属性、创建新的组账户。

图 5-45　RedHat 用户管理器

图 5-46　“创建新用户”界面

图 5-47　“创建新组群”界面

新用户和组账户创建好后，选择相应的账户并右击，在弹出的快捷菜单中选择“属性”选项，与 Windows 系统一样也可配置对应账户的详细属性选项。

这些操作都非常简单，因篇幅原因在此就不作具体介绍了，但也是考试的范围。