

## 第4章 Linux文件及系统管理

### 4.1 文件和目录

文件和目录是操作系统中十分重要的概念，文件和目录实现了操作系统对系统和用户的数据的管理，了解文件和目录的有关概念是理解操作系统的一个重要部分。

#### 4.1.1 文件和目录概述

##### 1. 文件

绝大多数操作系统(从DOS到Windows，从UNIX到Linux)中都有文件的概念。简单的讲文件就是一组相关纪录的集合。Linux中文件是无结构的字符流，即文件中的任两个字节是完全独立的。具体的信息要放到相应的环境中才能理解。

文件通过文件名来标识引用，即文件是通过文件名来进行管理的。系统中每一个文件都有一个文件名。

##### 2. 目录

文件多了就会发生混乱，因此目录就出现了。目录就是存放一组文件的“夹子”，Windows中的“文件夹”就是这个概念。所以目录就是一组相关文件的集合，我们通常都通过目录来管理文件。目录和文件一样也有自己的名字。而一个目录下面除了可以放文件之外，还可以存放目录，称为这个目录的子目录。这个子目录之下还可以有它自己的子目录，依此类推从而形成一个树状目录结构。其中最上层的目录我们称为根目录，而系统内的所有目录都是根目录的子目录。图4-1是一个树状目录结构的范例。

##### 3. 路径

相信使用过其他操作系统的用户对这个概念不会陌生。打个比方，某个文件就好像是在一个写字楼中的一间办公室内的一把椅子，当然还有其他很多椅子也在这个办公室里。现在要把它找出来坐在上面办公，那么只知道这个椅子是什么样的还不行，还需要知道如何到达哪里。文件名就是这椅子，而路径就是用来说明如何找到该椅子的。

Linux文件系统是由目录和目录下的文件一起构成的，是一个树型的结构。在这个系统最顶层的是根目录“/”。下面是各级其他目录。Linux路径是由从根目录到文件所经过的所有目录加上文件名连在一起构成的，中间以“/”分隔。

同样的和DOS相同，在每个目录上面都有名为“.”和“..”的两个文件，前者代表的是当前目录，后者代表的是当前目录的父目录(上一级目录)。

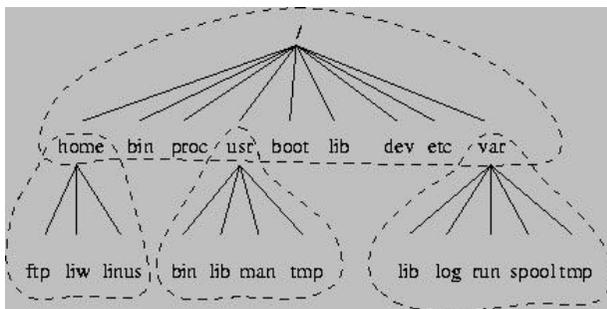
如果要存取不在工作目录中的文件，则除了文件名之外，也要指定目录的位置，而这就是路径。而路径由文件所在的目录加上文件名组成，中间再用“/”隔开。系统中每一个文件都有一个路径名，要想引用这个文件就要通过路径来引用。如图4-1所示，假设在图中的目录linus下有一个文件名为：linus-file，那么访问它的路径为：/home/linus/linus-file。

路径又分成以下两种：

绝对路径：路径名称以“/”开头，其中文件或目录的位置从根目录算起。

相对路径：路径名称不以“/”开头，其中文件或目录的位置从工作目录算起，工作目录可以理解为当前所在的目录。

标准文件系统布局(树型目录)



标准文件系统布局（树型目录）

图 4-1

#### 4. 文件目录命名规则

和DOS相比，L文件命名相当宽松。L文件名可以由字母、数字和一些标点符号组成。文件名中不能包含有空格和下列的字符：

! @ # \$ % ^ & \* ( ) " \ / | ; < >

但文件及目录名字的长度可以超过DOS的8.3的规定，只是通常要在256字之内。为了兼容UNIX文件系统，建议大家使用14个以下的字符作为文件名（一般的UNIX系统都只允许14个以下的字符作为文件名）。另外我们大家经常使用的Windows 95/98系统中是可以在文件名中使用空格的，Linux中实际上也有办法可以解决这个问题，那就是将文件名使用“ ”括起来，这样在双引号里面就可以随意使用任何字符了。也就是说没有任何限制了。但除非是真有必要，一般建议最好不要这么做。因为大多数程序不一定可以访问具有这种文件名的文件；而且使得文件系统的可移植性也变差。

#### 5. 处理文件的命令

见表4-1。

表4-1 文件操作常用命令

命 令	作 用
ls	列出目录的内容
ls -l	列出目录的详细内容
cd <dir>	切换工作目录为<dir>
Mkdir <dir>	建立名叫<dir>的目录
Rmdir <dir>	删除名叫<dir>的目录
cat <file>	看<file>的内容
More <file>	看<file>的内容，并且每印满一页后则暂停
rm <file>	删除<file>
cp <file-A> <file-B>	把<file-A>拷贝一份成<file-B>
ev <file-A> <file-B>	将<file-A>改名为<file-B>
Echo <string>	将<string>显示在屏幕上

### 4.1.2 文件类型

Linux文件系统中包括以下类型的文件：

**文本文件：**同其他操作系统的文本文件相同，是由一些字符在计算机内的 ASCII码组成的。通常这类文件在各个操作系统中是兼容的。

**二进制文件：**文件中保存的是数据的二进制表示，一般情况下用普通的文本浏览器所见到的是一堆乱码，需要专用的软件才能查看文件内容。例如：可执行文件、图像文件、声音文件等都是这类文件。

**目录文件：**Linux中对目录的管理也是通过文件进行的。顾名思义，目录文件就是一类特殊的文件，其中保存的信息不是普通的数据等，而是文件名和文件有关的信息，例如文件名、文件创建日期、文件类型、读取权限等。目录文件中还可以包含下一级的目录。

**连接文件：**普通的连接实际上不是文件，它们仅是指向同一索引节点的目录条目，是一个索引节点表。该表记录了一个文件有多少连接，这种连接不能跨越设备，因为不同设备上的i节点号不同，而另一种连接类型——符号链接——则没有这种限制。这将在后面详细介绍。

**设备文件：**设备文件是Linux系统中较特殊的文件。Linux中引入设备文件的目的是实现设备独立性。用户访问外部设备时就是通过设备文件进行的。操作系统对外设的支持程度直接影响到操作系统被使用的广泛程度，一个由于本身而限制了访问硬件的操作系统不能说是成功的操作系统。面对越来越多，越来越高级的外设，Linux将外设看作一个文件来管理，这样就避免了由于外设的增加而带来的问题。当需要增加新设备时，只要在操作系统内核中增加相应的设备文件即可。因此，设备文件在外设与操作系统之间提供了一个接口，用户使用外设就像使用普通文件一样。设备文件存放在 /dev目录下，它使用设备的主设备号和次设备号来区分指定的外设：主设备号说明设备类型，次设备号说明具体指哪一个设备。例如：/dev/fd0指系统中的一个软盘驱动器，主设备号 fd是软盘驱动器 (floppy disk)的缩写；次设备号 0是软盘驱动器编号，意思是指系统中的一软盘驱动器，也就是我们通常所说的“A:”盘。

**管道文件：**也是Linux中较特殊的文件类型，这类文件多用于进程间的通信方面。进程可通过管道文件实现互相的信息传输，一个进程将数据写在管道的一端，另一个进程从管道的另一端将数据读出，这样就能实现进程的相互协作。管道文件又可分为无名管道和有名管道两种。

## 4.2 文件权限

由于Linux是一个多用户的操作系统，这就要考虑安全性的问题。文件权限是操作系统安全性的一个重要因素。

### 4.2.1 文件权限的概念

与UNIX一样，Linux系统也是多用户系统。这样就产生了一个用户的文件会不会暴露在另一个用户面前的问题。为了保护用户的私人文件不被其他用户所侵犯，就出现了文件权限的概念。这种文件权限的概念允许文件和目录归一个特定的用户所拥有。用户权限除了

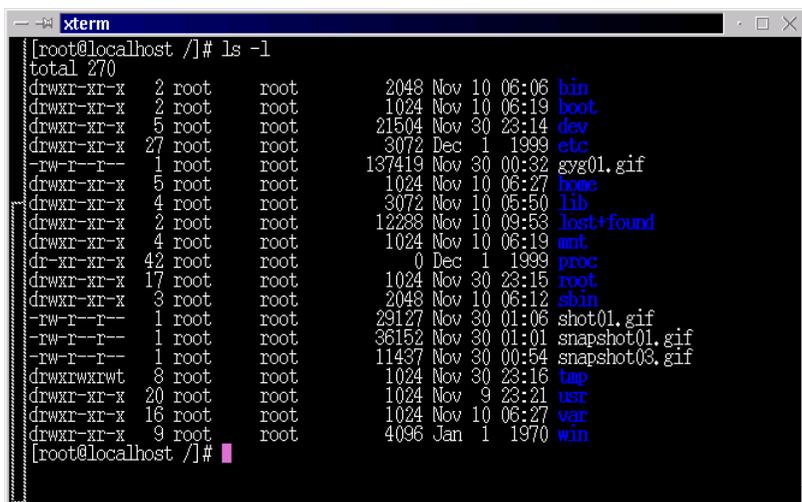
对他所拥有的文件或目录进行存取或其他操作，也包括限制其他用户在这些文件上的存取和操作。

例如，用户A在自己的主目录中建立了某些文件，那么用户A就成为这些文件的所有者，就有权对这些文件进行存取，也有权限制其他用户对这些文件的存取访问。Linux允许用户之间以及用户组之间共享文件。如果用户A愿意，他可以切断一切到他所拥有的文件的通路，即使得任何人都不能存取这些文件。当然在大多数情况下是只给其他用户分配只读权限（以只读方式访问自己的文件），而不给他们对文件的修改和删除的权限。

为了方便管理，每个文件都被一个特定用户所拥有的同时，还被系统指定的用户组所拥有。一个用户至少在一个用户组中，有时系统管理员还可以将一个用户编进多个用户组中。用户组通常是根据使用计算机的用户的种类来划分的。例如，在学校中根据用户的身份可以分为student、staff和guest等组。系统使用这些组来控制对资源的访问。

#### 4.2.2 文件权限的构成

文件的权限有三种：读、写和执行。读权限允许用户阅读文件的内容，对目录来说，允许用户列出目录中的内容。写权限允许用户写和修改文件，对目录来说，写权限允许用户在这个目录中建立新文件或删除文件。执行权限允许用户运行文件，对目录来说，执行权限允许用户进入和退出该目录。下面举例说明上面所说的文件权限。使用ls-l命令来显示文件列表，如图4-2所示。



```
[root@localhost ~]# ls -l
total 270
drwxr-xr-x  2 root    root      2048 Nov 10 06:06 bin
drwxr-xr-x  2 root    root      1024 Nov 10 06:19 boot
drwxr-xr-x  5 root    root     21504 Nov 30 23:14 dev
drwxr-xr-x 27 root    root     3072 Dec  1 1999 etc
-rw-r--r--  1 root    root    137419 Nov 30 00:32 gvg01.gif
drwxr-xr-x  5 root    root     1024 Nov 10 06:27 home
drwxr-xr-x  4 root    root     3072 Nov 10 05:50 lib
drwxr-xr-x  2 root    root    12288 Nov 10 09:53 lost+found
drwxr-xr-x  4 root    root     1024 Nov 10 06:19 mnt
dr-xr-xr-x 42 root    root         0 Dec  1 1999 proc
drwxr-xr-x 17 root    root     1024 Nov 30 23:15 root
drwxr-xr-x  3 root    root     2048 Nov 10 06:12 sbin
-rw-r--r--  1 root    root    29127 Nov 30 01:06 shot01.gif
-rw-r--r--  1 root    root    36152 Nov 30 01:01 snapshot01.gif
-rw-r--r--  1 root    root    11437 Nov 30 00:54 snapshot03.gif
drwxrwxrwt  8 root    root     1024 Nov 30 23:16 tmp
drwxr-xr-x 20 root    root     1024 Nov  9 23:21 usr
drwxr-xr-x 16 root    root     1024 Nov 10 06:27 var
drwxr-xr-x  9 root    root     4096 Jan  1 1970 win
[root@localhost ~]#
```

图 4-2

从图中所列出的文件的长格式信息中可以看出，文件信息共有9列。在第一列中由长度为十的字符组成一串字符的就是文件权限表示。每个文件的权限都是由一个十位的二进制数组成。

这十位二进制数分为四部分：第一部分一位，代表文件的类型（表4-2列出了文件类型的种类和标志符号）；第二部分是二到四位，代表文件所有者对文件的权限；第三部分是五到七位，代表该文件所有者所在组对该文件所拥有权限；第四部分是八到十位，代表其他用户对文件所拥有权限。第二列代表文件的链接数（我们将在下一节介绍文件的链接）；第三列是文件的所有者；第四列是文件所属的用户组；最后一列是文件名。

表4-2 文件类型和标志符号

文件类型	标志符号
普通文件	-
块设备文件	b
字符设备文件	C
连接文件	L
目录文件	D

表示文件权限的三个字符，依次代表读、写和执行权限。系统用“r”代表读权限、“w”代表写权限、“x”代表执行权限。当用户没有相应的权限时，该权限对应的位置用短线“-”表示。如图4-2中的第5行，文件名为gyg01.gif，第一个短线代表该文件是普通文件；后面的三个字符“rw\_”表示root用户对该文件有读、写权限，但没有执行权限。接下去的三个字符“r--”表示与用户root同属于root组的用户对该文件有读权限，但没有写权限和执行权限。也就是说，属于root组的其他用户只能读该文件。最后的三个字符“-r--”表示组外的其他用户也只有该文件的读权限，即只能阅读该文件，但不能对该文件进行写或执行操作。

### 4.2.3 文件权限的依赖性

要特别注意的是，用户对文件所拥有的权限依赖于他对文件所属的目录所拥有的权限。虽然一个文件的权限为“-rwxrwxrwx”，但其他用户也只有在对该文件所属目录拥有读写和执行权限时，才能够对该文件进行读、写和执行。

这就是说，如果用户A希望限制他人对其所拥有文件的访问，那么用户A只需将他的目录/home/A的权限设置成“-rwx-----”。这样，其他任何用户都不能访问他的目录，包括其中的所有文件和子目录。这样用户A就不必一个一个地修改他的文件的权限了。如果用户为了访问某个文件，这个用户必须具有对该文件路径名中所有目录的执行权限以及对该文件自身的相应的访问权限。

通常的用户都是非常开放的，文件的权限一般都设置成“-rw-r--r--”，这种权限的设置允许其他用户读文件，但不能写和执行。而目录的权限一般都设置成“-rwxr-xr-x”，这种权限的设置允许其他用户查看目录，但不能在其中建立或删除文件。

### 4.2.4 改变权限

在系统应用中有时需要让其他用户使用某个他本来不能使用的文件的时候，就需要改变文件的权限。在Linux中，使用chmod命令来进行有关权限的设置。但只有当文件的拥有者对某个文件有写的权限时，他才能够改变该文件的权限（超级用户对所有文件进行权限设置）。

chmod的命令格式主要有两种，下面是直接设置文件权限的第一种格式：

```
chmod [-R] mode filename1 filename2 ...
```

其中-R是指递归地改变所指定的目录参数及其子目录和它们所组成的文件的访问权限。mode是一个八进制的数，直接描述一个文件的权限。文件权限是一个十位的二进制数，Linux中规定：十位中每个位上如果出现了字母，则该位记为1，如果是“-”，则记为0，这样将十位都设置完后就将得到一个十位的二进制数。由于UNIX系统最开始是采用八进制数系统，所以

沿袭至今，Linux上仍然需要把这十位二进制数转化为八进制数（转化方法在许多计算机入门的书本中都有介绍，这里不再讲了），这样就得到一个四位的八进制数，也就是 mode参数了。实际上对于普通文件第一位始终为0，输入mode参数的时候写三位就可以了。

例如：将某文件权限设置为对文件所有者可读、可写、可执行，同组用户可读，其他用户可读。首先写出文件权限字段：-rwxr--r--，然后转化为二进制：0111100100，接着转化为八进制：0744，最后用chmod命令进行设置即可。chmod的第二种命令格式相对来说比较直观，更容易掌握。命令格式如下：

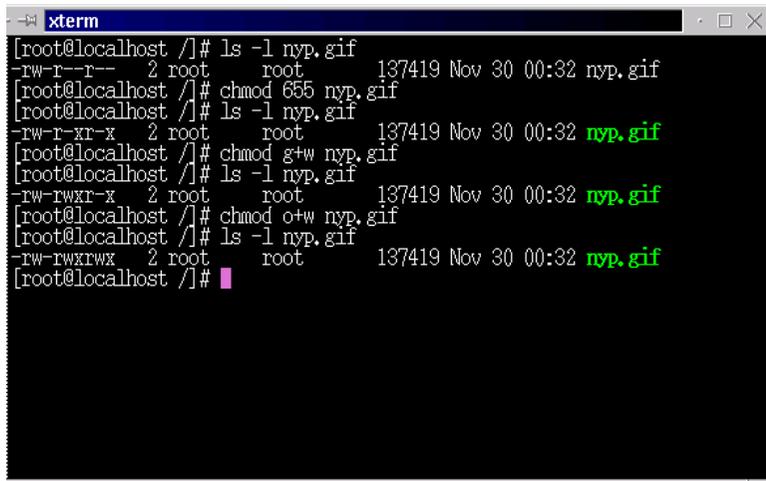
```
chmod {a u, g, o} {+-} {rwx} filename1 filename2 ...
```

该命令首先要指明权限所要赋予的用户类，可以是 a、u、g、o中的一个或多个，其中a代表所有的用户，u代表文件的所有者自身，g代表同组的用户，o代表组外的其他用户；接着，要指明是增加权限还是删除权限，用“+”选项表示增加权限，用“-”选项表示删除权限；最后，要指明所要增加或删除的权限，这些权限可以是 r、w和x中的一个或多个。

下面举个例子来进一步说明如何更改文件权限：假设在根目录下有一个名为 nyp.gif的文件，先用

```
ls -l
```

命令来查看其权限为 -rw-r--r--，将其转化为二进制表示“0110100100”，相应八进制表示为644，然后依次用第一种命令方式改变其权限为 -rw-r-xr-x，其八进制表示为655，所以用chmod 655 nyp.gif即可，第二种方式改变其权限为 -rw-rwxr-x和-rw-rwxrwx，如图4-3所示。



```
xterm
[root@localhost ~]# ls -l nyp.gif
-rw-r--r-- 2 root root 137419 Nov 30 00:32 nyp.gif
[root@localhost ~]# chmod 655 nyp.gif
[root@localhost ~]# ls -l nyp.gif
-rw-r-xr-x 2 root root 137419 Nov 30 00:32 nyp.gif
[root@localhost ~]# chmod g+w nyp.gif
[root@localhost ~]# ls -l nyp.gif
-rw-rwxr-x 2 root root 137419 Nov 30 00:32 nyp.gif
[root@localhost ~]# chmod o+w nyp.gif
[root@localhost ~]# ls -l nyp.gif
-rw-rwxrwx 2 root root 137419 Nov 30 00:32 nyp.gif
[root@localhost ~]#
```

图 4-3

## 4.3 文件链接

文件的链接可以使Linux系统中的一个文件有多个名字，即可以用多个不同的名字来访问同一个文件。这种方式是为了节省空间和方便多个用户共享而产生的。

### 4.3.1 硬链接

Linux系统是通过使用文件的 i 节点号来标识文件的，i 节点号是文件系统中文件的惟一标

识符。目录实际上是i节点号和其相应文件名的一个列表。目录中的每一个文件都是一个与特定的i节点的链接。硬链接就是为文件另取一个文件名并将它与原来的i节点链接。可以使用ln命令来为一个文件建立多个链接。例如，假设用户在当前的目录中有gyg01.gif。使用命令

```
ls-l
```

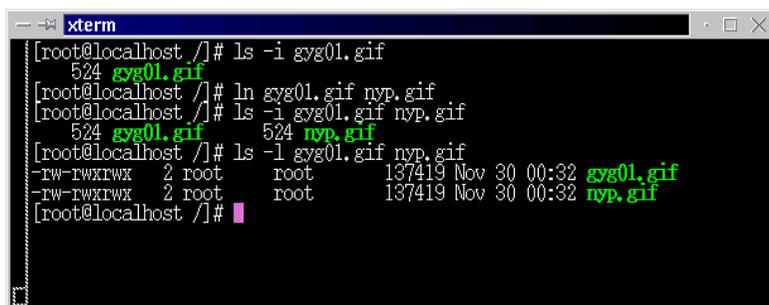
我们可以看到这个文件的i节点号。可见，文件gyg01.gif在文件系统中的i节点号是524。用户可以用以下命令建立一个称做nyp.gif的到文件gyg01.gif的链接：ln fabc fxyz。使用

```
ls-i
```

我们可以看到这两个文件有相同的i节点号，然后像上一节介绍的那样使用

```
ls-l
```

命令可以列出文件的链接数以及其他信息，如图4-4所示。



```
xterm
[root@localhost ~]# ls -i gyg01.gif
524 gyg01.gif
[root@localhost ~]# ln gyg01.gif nyp.gif
[root@localhost ~]# ls -l gyg01.gif nyp.gif
-rw-rw-rw- 2 root root 137419 Nov 30 00:32 gyg01.gif
-rw-rw-rw- 2 root root 137419 Nov 30 00:32 nyp.gif
[root@localhost ~]# ls -l gyg01.gif nyp.gif
-rw-rw-rw- 2 root root 137419 Nov 30 00:32 gyg01.gif
-rw-rw-rw- 2 root root 137419 Nov 30 00:32 nyp.gif
[root@localhost ~]#
```

图 4-4

执行这些操作以后，不论访问gyg01.gif还是nyp.gif，实际上都是访问同一个文件。如果对nyp.gif做了修改，实际上也是对gyg01.gif作修改。也就是说，gyg01.gif和nyp.gif其实是同一个文件。

上面这种链接过程的链接称为硬链接，因为它们直接链接到i节点上。注意，建立硬链接只能在同一个文件系统中进行，而符号链接（下面介绍）就没有这个限制。当用户使用rm命令删除文件时，实际上是删除了这个文件的一个链接。只有当文件没有其他链接时，才是真正地从文件系统中删除了。所以为了删除文件，用户必须首先删除该文件的所有链接。

上面信息中的第二列代表了文件的链接数。从图中可以发现两个文件链接数都为2。目录实际上只是一个包含了从链接到i节点号的转换信息的文件。每个目录中至少有两个硬链接——指向目录自身的链接“.”和指向其父目录的链接“..”，根目录中的“..”链接也指向其自身。

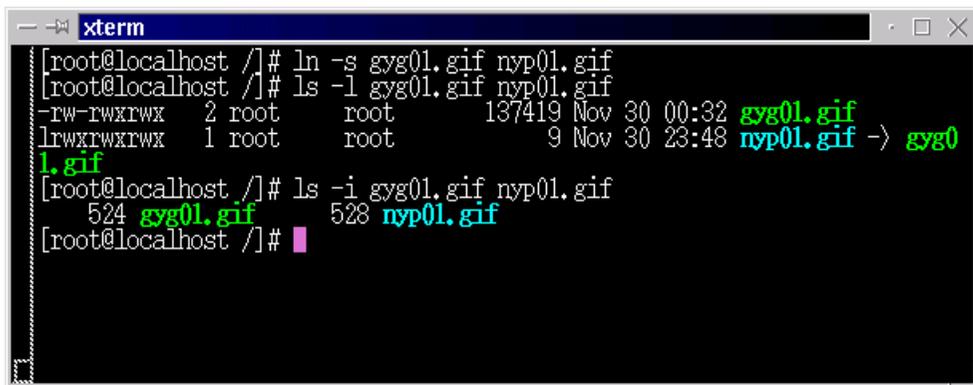
### 4.3.2 符号链接

符号链接与硬链接不同，它是另一种类型的链接。符号链接允许用户为文件另取一个名字，但却不将其与i节点相链接。

可以用ln-s命令建立一个文件的符号链接。使用命令ln-s gyg01.gif nyp01.gif就会建立一个指向文件gyg01.gif的符号链接nyp01.gif。如果这时我们使用ls-i命令，便会看到这两个文件实际上有不同的i节点号，一个是524，另一个是528。这时使用

```
ls-l
```

命令，我们就会看到文件 nyp01.gif 是指向 gyg01.gif 的一个符号链接，如图 4-5 所示。



```
xterm
[root@localhost ~]# ln -s gyg01.gif nyp01.gif
[root@localhost ~]# ls -l gyg01.gif nyp01.gif
-rw-rwxrwx  2 root    root    137419 Nov 30 00:32 gyg01.gif
lrwxrwxrwx  1 root    root      9 Nov 30 23:48 nyp01.gif -> gyg01.gif
[root@localhost ~]# ls -li gyg01.gif nyp01.gif
 524 gyg01.gif  528 nyp01.gif
[root@localhost ~]#
```

图 4-5

可以看到符号链接的权限位是 `rw-rwxrwx`，其实它本没有实际意义。它们总是为 `rw-rwxrwx`。符号链接的权限是由符号链接的目标 (这里是 `gyg01.gif`) 的权限决定的。

从功能上讲，硬链接和符号链接是相似的，但是它们也有一些区别。例如，用户可以对一个不存在的文件建立符号链接；但硬链接却不能。系统核心对符号链接和硬链接的操作是不同的，虽然这只是技术上的不同，但这一点有时却很重要。符号链接只是指明它们所指向的文件，这对用户是很有帮助的，而硬链接的问题是，没有一个简单的办法用来确定有哪些文件是链接到相同的 `i` 节点上。

Linux 系统中很多地方都用到了链接。对 `/lib` 中的共享程序库映像来说，符号链接显得更为重要。

## 4.4 文件系统和标准文件系统布局

操作系统通过文件系统管理系统中的数据，Linux 与 UNIX 的文件系统布局基本相同，都是由根目录和子目录构成的。

### 4.4.1 文件系统

文件系统是操作系统中非常重要的概念，文件系统管理机器上的文件目录，使之有效地被存取访问。文件系统是操作系统用于明确磁盘或分区上的文件的方法和数据结构，即在磁盘上组织文件的方法。也可以指用于存储文件的磁盘或分区，或文件系统种类。简单地讲，文件系统是指按照一定规则组织的文件结构。Linux 中文件系统是按照树型结构组织的，所有 Linux 的数据 (包括程序、库、用户文件等等) 都是由文件系统按照树型目录结构管理的。这些概念和 DOS/Windows 是一样的。

Linux 文件系统的树型结构如图 4-1 所示，其中每台机器都有根文件系统 (一般在本地盘中，当然也可以在 RAM 盘或网络盘中)。它包含系统引导和使其他文件系统得以 `mount` 所必要的文件，根文件系统应该有单用户状态所必需的足够的內容。还应该包括修复损坏系统、恢复备份等的工具。

除了根文件系统外，用户可以根据自己的需要创建一个或多个其他的文件系统。

#### 4.4.2 标准文件系统布局

UNIX经过长期的发展，其文件系统的命名规则已经趋于规范化，大多数用户都遵循这样的规定。Linux则安全继承了UNIX的文件系统布局。以下的介绍基于Linux文件系统标准FSSTND 1.2版本。这一版本是非强制性的，就是说，如果你愿意，完全可以采用其他的文件系统布局。但由于这一版本已经得到了很多Linux程序的支持，遵循它将会使管理更加容易进行。并且这样一个标准具有易于写或移植Linux软件、管理Linux系统的优点，所以，如果没有什么特殊的理由，最好还是遵从FSSTND标准。FSSTND意图遵从UNIX传统和当前趋势，使熟悉其他UNIX系统的人对Linux系统更容易接受(反之亦然)。

标准文件系统布局将整个目录树分为小的部分，每个部分可以在自己的磁盘或分区上，只要能为磁盘容量所容纳，并易于备份及其他系统管理即可。如图中被虚线所画出的范围所示，文件系统主要部分是根、/usr、/var和/home文件系统。每个部分各有其不同的目的。目录树已被设计成能在Linux机器的网络中很好地工作，可以通过只读设备(如CD-ROM)或NFS网络共享文件系统的一些部分。

下面对几个重要的目录做一下简要介绍：

##### 1. 根文件系统

根文件系统是每台计算机都必须具有的部分。它包含系统引导和使其他文件系统得以mount所必要的文件，根文件系统应该有单用户状态所必需的足够的内容。还应该包括修复损坏系统、恢复备份等的工具。根文件系统一般应该比较小，因为包括严格的文件和一个小的不经常改变的文件系统通常都不容易损坏。如果根文件系统发生损坏，则意味着除非用特定的方法(例如从软盘)引导系统，否则系统将无法引导，所以要格外小心。

##### 2. /usr文件系统

/usr文件系统包含所有命令、库、帮助(man)页和其他一般操作中所需的不改变的文件。/usr不应该包括对特定机器使用的特定的文件，也不应该有一般使用中要修改的文件。这样允许此文件系统中的文件通过网络共享，才可以使之更有效，因为这样节省了磁盘空间(/usr很容易是数百兆)，且易于管理(当升级应用时，只有主/usr需要改变，而无须改变每台机器)。即使此文件系统在本地盘上，也可以将其设为只读属性，以减少系统崩溃时对文件系统的损坏。

##### 3. /var文件系统

/var文件系统包含会改变的文件，比如，spool目录(mail、news、打印机等用的)，log(日志)文件、formatted manual pages和暂存文件。传统上/var的所有东西曾在/usr下的某个地方，但这样/usr就不可能只读安装了。

##### 4. /home文件系统

/home文件系统包含用户本地目录，即系统上的所有实际数据。最好将/home目录安装到专门的目录树或文件系统(分区)中，因为这样易于备份，而不必备份其他文件系统(至少不必经常备份，因为它们很少改变)。一个大的/home可能要分为若干文件系统，需要在/home下加一级名字，如/home/students、/home/staff等。

虽然上面将不同的部分称为文件系统，但并不需要将它们安装在不同的磁盘上，它们也不必是真的分离的文件系统。如果用户系统是小的单用户系统，并且用户希望使它简单化，那么它们可以很容易地放在一个文件系统中。根据磁盘容量和不同目的所需分配的空间，目录树也可以分到不同的文件系统中。重要的是使用标准的名字，即使/var和/usr在同一分区上，

名字/usr/lib/libc.a 和/var/adm/messages 必须能工作,例如将/var 下的文件移动到/usr/var ,并将/var 作为/usr/var 的符号连接,这样使文件组织有序,并且更加容易查找。UNIX文件结构根据文件使用目的来分组文件,即所有的命令在一个地方,所有的数据在另一个地方,所有的文档又在一个地方,等等。另一个方法是根据属于的程序分组文件,即所有 Emacs文件在一个目录中,所有txt文件在另一个中,等等。后一种方法的问题是文件难于共享(程序目录经常同时包含静态可共享的和动态不可共享的文件),有时难于查找(例如man页在极大数量的地方,使man程序查找它们极其困难)。

## 4.5 基本目录介绍

### 4.5.1 根文件系统

通常情况下,根文件系统所占空间一般应该比较小,因为其中的绝大部分文件都不需要经常改动,而且包括严格的文件和一个小的不经常改变的文件系统不容易损坏。

除了可能的一个叫 /vmlinuz标准的系统引导映像之外,根目录一般不含任何文件。所有其他文件在根文件系统的子目录中。

#### 1. /bin目录

/bin目录包含了引导启动所需的命令或普通用户可能用的命令(可能在引导启动后)。这些命令都是二进制文件的可执行程序(bin是binary--二进制的简称),多是系统中重要的系统文件。例如,4.1节中介绍的文件操作常用命令ls,cp,mkdir等都在其中。

#### 2. /sbin目录

/sbin目录类似/bin ,也用于存储二进制文件。因为其中的大部分文件多是系统管理员使用的基本的系统程序,所以虽然普通用户必要且允许时可以使用,但一般不给普通用户使用。

#### 3. /etc目录

/etc目录存放着各种系统配置文件,其中包括了用户信息文件 /etc/passwd,系统初始化文件/etc/rc等。Linux正是靠这些文件才得以正常地运行。

#### 4. /root目录

/root 目录是超级用户的目录。

#### 5. /lib目录

/lib目录是根文件系统上的程序所需的共享库,存放了根文件系统程序运行所需的共享文件。这些文件包含了可被许多程序共享的代码,以避免每个程序都包含有相同的子程序的副本,故可以使得可执行文件变得更小,节省空间。

#### 6. /lib/modules 目录

/lib/modules 目录包含系统核心可加载各种模块,尤其是那些在恢复损坏的系统时重新引导系统所需的模块(例如网络和文件系统驱动)。

#### 7. /dev目录

/dev目录存放了设备文件,即设备驱动程序,用户通过这些文件访问外部设备。比如,用户可以通过访问/dev/mouse来访问鼠标的输入,就像访问其他文件一样。

#### 8. /tmp目录

/tmp 目录存放程序在运行时产生的信息和数据。但在引导启动后,运行的程序最好使用

/var/tmp来代替/tmp，因为前者可能拥有一个更大的磁盘空间。

#### 9. /boot目录

/boot目录存放引导加载器(bootstrap loader)使用的文件，如LILO，核心映像也经常放在这里，而不是放在根目录中。但是如果有许多核心映像，这个目录就可能变得很大，这时使用单独的文件系统会更好一些。还有一点要注意的是，要确保核心映像必须在IDE硬盘的前1024柱面内。

#### 10. /mnt目录

/mnt目录是系统管理员临时安装 (mount)文件系统的安装点。程序并不自动支持安装到/mnt。/mnt下面可以分为许多子目录，例如/mnt/dosa可能是使用MSDOS文件系统的软驱，而/mnt/exta可能是使用ext2文件系统的软驱，/mnt/cdrom光驱等等。

#### 11. /proc, /usr, /var, /home目录

其他文件系统的安装点。

### 4.5.2 /etc文件系统

/etc目录包含各种系统配置文件，下面说明其中的一些。其他的你应该知道它们属于哪个程序，并阅读该程序的man页。许多网络配置文件也在/etc中。

#### 1. /etc/rc或/etc/rc.d或/etc/rc?.d

启动、或改变运行级时运行的脚本或脚本的目录。

#### 2. /etc/passwd

用户数据库，其中的域给出了用户名、真实姓名、用户起始目录、加密口令和用户的其他信息。

#### 3. /etc/fdprm

软盘参数表，用以说明不同的软盘格式。可用setfdprm进行设置。更多的信息见setfdprm的帮助页。

#### 4. /etc/fstab

指定启动时需要自动安装的文件系统列表。也包括用swapon -a启用的swap区的信息。

#### 5. /etc/group

类似/etc/passwd，但说明的不是用户信息而是组的信息。包括组的各种数据。

#### 6. /etc/inittab

init的配置文件。

#### 7. /etc/issue

包括用户在登录提示符前的输出信息。通常包括系统的一段短说明或欢迎信息。具体内容 by 系统管理员确定。

#### 8. /etc/magic

“file”的配置文件。包含不同文件格式的说明，“file”基于它猜测文件类型。

#### 9. /etc/motd

motd是Message Of The Day的缩写，用户成功登录后自动输出。内容由系统管理员确定。常用于通告信息，如计划关机时间的警告等。

#### 10. /etc/mstab

当前安装的文件系统列表。由脚本 (scrip)初始化，并由mount命令自动更新。当需要一

个当前安装的文件系统的列表时使用 (例如df 命令)。

#### 11. /etc/shadow

在安装了影子 (shadow) 口令软件的系统上的影子口令文件。影子口令文件将 /etc/passwd 文件中的加密口令移动到 /etc/shadow 中, 而后者只对超级用户 (root) 可读。这使破译口令更困难, 以此增加系统的安全性。

#### 12. /etc/login.defs

login 命令的配置文件。

#### 13. /etc/printcap

类似 /etc/termcap, 但针对打印机。语法不同。

#### 14. /etc/profile、/etc/csh.login、/etc/csh.cshrc

登录或启动时 Bourne 或 C shells 执行的文件。这允许系统管理员为所有用户建立全局缺省环境。

#### 15. /etc/securetty

确认安全终端, 即哪个终端允许超级用户 (root) 登录。一般只列出虚拟控制台, 这样就不可能 (至少很困难) 通过调制解调器 (modem) 或网络闯入系统并得到超级用户特权。

#### 16. /etc/shells

列出可以使用的 shell。chsh 命令允许用户在本文件指定范围内改变登录的 shell。提供一台机器 FTP 服务的服务进程 ftpd 检查用户 shell 是否列在 /etc/shells 文件中, 如果不是, 将不允许该用户登录。

#### 17. /etc/termcap

终端性能数据库。说明不同的终端用什么 “转义序列” 控制。写程序时不直接输出转义序列 (这样只能工作于特定品牌的终端), 而是从 /etc/termcap 中查找要做的工作的正确序列。这样, 多数的程序可以在多数终端上运行。

### 4.5.3 /dev 文件系统

/dev 目录包括所有设备的设备文件。设备文件用特定的约定命名, 这在设备列表中说明。设备文件在安装时由系统产生, 以后可以用 /dev/MAKEDEV 描述。/dev/MAKEDEV.local 是系统管理员为本地设备文件 (或连接) 写的描述文稿 (即如一些非标准设备驱动不是标准 MAKEDEV 的一部分)。下面简要介绍 /dev 下一些常用文件。

#### 1. /dev/console

系统控制台, 也就是直接和系统连接的监视器。

#### 2. /dev/hd

IDE 硬盘驱动程序接口。如: /dev/hda 指的是第一个硬盘, had1 则是指 /dev/hda 的第一个分区。如系统中有其他的硬盘, 则依次为 /dev/hdb、/dev/hdc、.....; 如有多个分区则依次为 hda1、hda2.....

#### 3. /dev/sd

SCSI 磁盘驱动程序接口。如有系统有 SCSI 硬盘, 就不会访问 /dev/had, 而会访问 /dev/sda。

#### 4. /dev/fd

软驱设备驱动程序。如: /dev/fd0 指系统的第一个软盘, 也就是通常所说的 A: 盘, /dev/fd1 指第二个软盘, .....而 /dev/fd1H1440 则表示访问驱动器 1 中的 4.5 高密盘。

#### 5. /dev/st

SCSI磁带驱动器驱动程序。

#### 6. /dev/tty

提供虚拟控制台支持。如：/dev/tty1指的是系统的第一个虚拟控制台，/dev/tty2则是系统的第二个虚拟控制台。

#### 7. /dev/pty

提供远程登陆伪终端支持。在进行Telnet登录时就要用到/dev/pty设备。

#### 8. /dev/ttys

计算机串行接口，对于DOS来说就是“COM1”口。

#### 9. /dev/cua

计算机串行接口，与调制解调器一起使用的设备。

#### 10. /dev/null

“黑洞”，所有写入该设备的信息都将消失。例如：当想要将屏幕上的输出信息隐藏起来时，只要将输出信息输入到/dev/null中即可。

### 4.5.4 /usr文件系统

/usr是个很重要的目录，通常这一文件系统很大，因为所有程序安装在这里。/usr里的所有文件一般来自Linux发行版(distribution)；本地安装的程序和其他东西在/usr/local下，因为这样可以在升级新版系统或新发行版时无须重新安装全部程序。/usr目录下的许多内容是可选的，但这些功能会使用户使用系统更加有效。/usr可容纳许多大型的软件包和它们的配置文件。下面列出一些重要的目录(一些不太重要的目录被省略了)。

#### 1. /usr/X11R6

包含X Window系统的所有可执行程序、配置文件和支持文件。为简化X的开发和安装，X的文件没有集成到系统中。X Window系统是一个功能强大的图形环境，提供了大量的图形工具程序。用户如果对Microsoft Windows或Machintosh比较熟悉的话，就不会对X Window系统感到束手无策了。

#### 2. /usr/X386

类似/usr/X11R6，但是是专门给X11 Release 5的。

#### 3. /usr/bin

集中了几乎所有用户命令，是系统的软件库。另有些命令在/bin或/usr/local/bin中。

#### 4. /usr/sbin

包括了根文件系统不必要的系统管理命令，例如多数服务程序。

#### 5. /usr/man、/usr/info、/usr/doc

这些目录包含所有手册页、GNU信息文档和各种其他文档文件。每个联机手册的“节”都有两个子目录。例如：/usr/man/man1中包含联机手册第一节的源码(没有格式化的原始文件)，/usr/man/cat1包含第一节已格式化的内容。L联机手册分为以下九节：内部命令、系统调用、库函数、设备、文件格式、游戏、宏软件包、系统管理和核心程序。

#### 6. /usr/include

包含了C语言的头文件，这些文件多以.h结尾，用来描述C语言程序中用到的数据结构、

子过程和常量。为了保持一致性，这实际上应该放在 `/usr/lib` 下，但习惯上一直沿用了这个名字。

#### 7. `/usr/lib`

包含了程序或子系统的不变的数据文件，包括一些 `site-wide` 配置文件。名字 `lib` 来源于库 (library)；编程的原始库也存在 `/usr/lib` 里。当编译程序时，程序便会和其中的库进行连接。也有许多程序把配置文件存入其中。

#### 8. `/usr/local`

本地安装的软件和其他文件放在这里。这与 `/usr` 很相似。用户可能会在这发现一些比较大的软件包，如 `TEX`、`Emacs` 等。

### 4.5.5 `/var` 文件系统

`/var` 包含系统一般运行时要改变的数据。通常这些数据所在的目录的大小是要经常变化或扩充的。原来 `/var` 目录中有些内容是在 `/usr` 中的，但为了保持 `/usr` 目录的相对稳定，就把那些需要经常改变的目录放到 `/var` 中了。每个系统是特定的，即不通过网络与其他计算机共享。下面列出一些重要的目录 (一些不太重要的目录省略了)。

#### 1. `/var/catman`

包括了格式化过的帮助 (man) 页。帮助页的源文件一般存在 `/usr/man/man` 中；有些 man 页可能有预格式化的版本，存在 `/usr/man/cat` 中。而其他的 man 页在第一次看时都需要格式化，格式化完的版本存在 `/var/man` 中，这样其他人再看相同的页时就无须等待格式化了。

(`/var/catman` 经常被清除，就像清除临时目录一样。)

#### 2. `/var/lib`

存放系统正常运行时要改变的文件。

#### 3. `/var/local`

存放 `/usr/local` 中安装的程序的可变数据 (即系统管理员安装的程序)。注意，如果必要，即使本地安装的程序也会使用其他 `/var` 目录，例如 `/var/lock`。

#### 4. `/var/lock`

锁定文件。许多程序遵循在 `/var/lock` 中产生一个锁定文件的约定，以用来支持他们正在使用某个特定的设备或文件。其他程序注意到这个锁定文件时，就不会再使用这个设备或文件。

#### 5. `/var/log`

各种程序的日志 (Log) 文件，尤其是 `login` (`/var/log/wtmp` log 纪录所有到系统的登录和注销) 和 `syslog` (`/var/log/messages` 纪录存储所有核心和系统程序信息)。`/var/log` 里的文件经常不确定地增长，应该定期清除。

#### 6. `/var/run`

保存在下一次系统引导前有效的关于系统的信息文件。例如，`/var/run/utmp` 包含当前登录的用户的信息。

#### 7. `/var/spool`

放置“假脱机 (spool)”程序的目录，如 `mail`、`news`、打印队列和其他队列工作的目录。每个不同的 `spool` 在 `/var/spool` 下有自己的子目录，例如，用户的邮箱就存放在 `/var/spool/mail` 中。

#### 8. /var/tmp

比/tmp 允许更大的或需要存在较长时间的临时文件。

注意 系统管理员可能不允许/var/tmp 有很旧的文件。

### 4.5.6 /proc文件系统

/proc 文件系统是一个伪的文件系统，就是说它是一个实际上不存在的目录，因而这是一个非常特殊的目录。它并不存在于某个磁盘上，而是由核心在内存中产生。这个目录用于提供关于系统的信息。下面说明一些最重要的文件和目录 (/proc 文件系统在proc man页中有更详细的说明)。

#### 1. /proc/X

关于进程X的信息目录，这一X是这一进程的标识号。每个进程在 /proc 下有一个名为自己进程号的目录。

#### 2. /proc/cpuinfo

存放处理器(CPU)的信息，如CPU的类型、制造商、型号和性能等。

#### 3. /proc/devices

当前运行的核心配置的设备驱动力的列表。

#### 4. /proc/dma

显示当前使用的DMA通道。

#### 5. /proc/filesystems

核心配置的文件系统信息。

#### 6. /proc/interrupts

显示被占用的中断信息和占用者的信息，以及被占用的数量。

#### 7. /proc/ioports

当前使用的I/O端口。

#### 8. /proc/kcore

系统物理内存映像。与物理内存大小完全一样，然而实际上没有占用这么多内存；它仅仅是在程序访问它时才被创建。(注意：除非你把它拷贝到什么地方，否则 /proc 下没有任何东西占用任何磁盘空间。)

#### 9. /proc/kmsg

核心输出的消息。也会被送到 syslog。

#### 10. /proc/ksyms

核心符号表。

#### 11. /proc/loadavg

系统“平均负载”；3个没有意义的指示器指出系统当前的工作量。

#### 12. /proc/meminfo

各种存储器使用信息，包括物理内存和交换分区 (swap)。

#### 13. /proc/modules

存放当前加载了哪些核心模块信息。

14. /proc/net

网络协议状态信息。

15. /proc/self

存放到查看/proc 的程序的进程目录的符号连接。当 2个进程查看/proc 时，这将会是不同的连接。这主要便于程序得到它自己的进程目录。

16. /proc/stat

系统的不同状态，例如，系统启动后页面发生错误的次数。

17. /proc/uptime

系统启动的时间长度。

18. /proc/version

核心版本。

## 4.6 启动和关机

### 4.6.1 启动和关机概述

开启计算机并导致其操作系统被加载的过程叫引导。启动过程中，计算机首先加载了一小段叫 bootstrap loader的程序，它依次加载和启动操作系统，bootstrap loader通常存储在硬盘或软盘的固定的位置。这 2步过程的理由是操作系统大而复杂，而计算机加载的第一段代码很小(几百字节)，以免使固件不必要地复杂化。

不同的计算机的bootstrap不同。对于PC，计算机(它的BIOS)读软盘或硬盘的第一个扇区(叫引导扇)。bootstrap loader包含在这个扇区中。它加载位于磁盘(和其他)的其他地方的操作系统。

Linux加载后，它初始化硬件和设备驱动，然后运行 init。init启动其他进程以允许用户登录和做其他事情。这部分的细节在下面讨论。

为了关闭一个Linux系统，首先所有进程被告知结束(这使它们关闭所有文件，完成必要的其他事情，使之整齐地结束)，然后unmount文件系统和对换区，最后打印可以关掉电源的信息到控制台。如果没有遵循正确的过程，可怕的事情可能发生。最重要的，文件系统缓冲 cache可能没有回写，这意味着其中的所有数据将丢失，磁盘上的文件系统不完整，并可能不可用。

### 4.6.2 近观启动过程

可以从软盘或硬盘引导Linux。

当PC引导后，BIOS做一些测试保证一切正常，然后开始真正的引导。它选择一个磁盘(通常是第一个软驱，如果有软盘的话，否则就是第一个硬盘，顺序是可设置的)。然后读第一个扇区，这叫引导扇；对于硬盘，也叫主引导记录，因为硬盘可以包含多个分区，每个分区都有自己的引导扇。

引导扇包含一个小程序(小到可以存入一个扇区)，它的责任是从磁盘读入真正的操作系统并启动之。从软盘启动Linux时，引导扇包含的代码只读前数百个数据块(当然，依赖于核心的大小)到预定的内存位置。Linux引导软盘上，没有文件系统，核心存在连续的扇区中，因为这样简化了引导过程。当然，使用LILO(Linux LOader)可以从文件系统引导。

从硬盘引导，主引导记录的代码检查分区表(也在主引导记录扇区中)，确认活动分区(标

记为可引导的分区),从该分区读引导扇区,然后启动该引导扇区的代码。该分区的引导扇区的代码做的与软盘所做的相同:从该分区读入核心并启动。但细节不同,因为一般只给核心映像做一个单独的分区是没什么用的,所以分区引导扇中的代码不能只顺序地读磁盘,它必须找到文件系统把它们放在哪些扇区中。有几个方法解决这个问题,但最通常的方法是使用 LILO。(关于如何做的细节与这里的讨论无关;更多的信息请看LILO文档,它很全面)。

用LILO引导时,它读入并引导缺省核心。也可以设置LILO,使之能引导若干个核心之一,甚至其他操作系统,也可以在引导时让用户选择引导哪个核心或操作系统。LILO可以设置为如果有人引导时按住 alt、shift或ctrl键(LILO启动时),LILO将不立即引导缺省的而问用户引导哪个。LILO可以设置为带一个timeout选项并询问,当超时时,就引导缺省核心。

从软盘和硬盘启动各有优势,但通常从硬盘启动更好,因为这避免了关于软盘的争论。然而,安装相同从硬盘启动可能有更多的麻烦,因此很多人先用软盘引导,然后当相同工作很好后,再安装LILO从硬盘引导。

Linux核心被读入内存后,才真正启动了,概述如下:Linux核心是被压缩安装的,所以它首先得解压自己。核心映像开头包括一个解压的小程序。如果你有Linux可识别的super-VGA卡,且支持一些特殊的文本模式(如100列40行),Linux会问你要用哪个模式。编译核心时,可能预定了一个视频模式,就不会问了。这也可以用LILO或rdev完成。然后,核心检查还有什么其他硬件(硬盘,软盘,网卡...),并配置适当的设备驱动;同时,输出查找结果的信息。例如,我引导时,得到类似如下信息:

```
LILO boot:
Loading linux.
Console: colour EGA+ 80x25, 8 virtual consoles
Serial driver version 3.94 with no serial options enabled
tty00 at 0x03f8 (irq = 4) is a 16450
tty01 at 0x02f8 (irq = 3) is a 16450
lp_init: lp1 exists (0,using polling driver
Memory: 7332k/8192k available (300k kernel code, 384k reserved, 176k data)
Floppy drive(s): fd0 is 1.44M, fd1 is 1.2M
Loopback device init
Warning WD8013 board not found at i/o = 280.
Math coprocessor using irq13 error reporting.
Partition check:
hda: hda1 hda2 hda3
VFS: Mounted root (ext filesystem).
Linux version 0.99.pl9-1 (root@haven) 05/01/93 14:12:20
```

精确的文本在不同系统上不同,依赖硬件、Linux版本、及其配置。

之后核心试图mount根文件系统。位置可在编译时设置,或在任何时候使用rdev或LILO。文件系统类型自动检测。如果根文件系统mount失败,例如因为你忘了在核心中包含相关的文件系统驱动,核心将失败,系统停止(此时没什么可做了)。

根文件系统通常被只读mount(这可用与位置相同的方法)。这可使文件系统在mount上时检查;检查一个可读写的已mount的文件系统可不是个好主意。然后,核心在后台启动程序init(位于/sbin/init)(它的进程号是1)。init做许多启动工作。确切的事依赖于设置。它至少要启动一些必要的后台守候程序。init然后切换到多用户模式并启动getty,提供虚拟控制台和串

行线。getty 是一个让用户通过虚拟控制台和串行终端登录的程序。init 还可能启动一些其他程序，这些基于设置。至此，引导完成，系统启动并正常运行。

### 4.6.3 重新启动系统

重新启动就是完全关闭系统，关掉电源，然后再打开。简单方法是用 shutdown 重新启动系统而不是仅停止系统。这要使用 shutdown 的 -r 选项，例如命令 shutdown -r now。许多Linux系统在按ctrl-alt-del键时运行 shutdown -r now。这是可设置的，比如在多用户系统中设置一定的延时也许更好。如果是谁都能接触到的系统，那么最好设置为按 ctrl-alt-del 什么也不干。

### 4.6.4 紧急启动(软)盘

并非总可以从硬盘引导。例如，LILO设错了，系统可能就无法引导。这时，需要另一个总能引导的方法。对于典型的PC，可能是软驱。

许多Linux distributions允许在安装时产生一张紧急引导盘(emergency boot floppy)。然而，有些这样的引导盘只包含核心。有时这些程序是不够的：例如，你可能需要回存备份，而备份/回存软件在Linux安装盘里没有。因此，可能需要自己产生 root盘。Graham Chapman写的 Bootdisk HOWTO([Cha]) 包含关于此的指导。当然，你必须记得使紧急引导盘和 root盘最新。

root盘被mount上时，不能用软驱干其他任何事，因此如果你只有一个软驱可能不太方便。然而，如果你有足够的内存，可以设置引导盘将 root盘加载到RAM盘上(为此，引导盘的核心需要特殊设置)。一旦root盘被加载到RAM盘中，软驱就可以用于mount其他盘了。

### 4.6.5 有关关机的更多信息

关闭Linux系统时，遵循正确的过程是很重要的。否则，文件系统可能成为废物，文件可能变成杂乱的。这是因为Linux使用磁盘缓存，并不立即将数据写到磁盘，而是间歇地回写。这极大地改善了性能，但同时也意味着如果你只是关闭电源，cache可能保留着大量数据，而磁盘上的数据可能不是一个全部的正在工作的文件系统(因为有些数据已经回写到硬盘，而有些没有)。

另一个不能直接关闭电源的原因是：在多任务系统中，后台可能运行着很多东西，关闭电源可能损失惨重。使用正确的关机顺序，可以保证所有的后台进程得以保存它们的数据。正常关闭Linux系统的命令是shutdown。它通常使用2种方法之一。如果系统只有你一个用户，使用shutdown 的通常方法是退出所有运行程序，从所有虚拟控制台注销，用 root登录(如果你已经是root，当然不必再注销、登录，但应该换到根目录，以免 unmount由于出现问题)，然后运行命令 shutdown -h now(虽然单用户时一般不必要，但如果需要一个延时，用一个加号加一个表示分钟的数目代替now)。如果系统是多用户，使用命令 shutdown -h +time message，time是到系统停止的分钟数，message是告知所有用户系统关机原因的短信息。

```
# shutdown -h +10 'We will install a new disk. System should  
> be back on-line in three hours.'
```

上面的命令警告所有用户，系统将在 10分钟后关闭，他们最好保存信息，否则将丢失。警告将显示在所有登录的终端上，包括所有的 xterm 上：

```
Broadcast message from root (tty0) Wed Aug 2 01:03:25 1995...  
We will install a new disk. System should  
be back on-line in three hours.
```

The system is going DOWN for system halt in 10 minutes !!

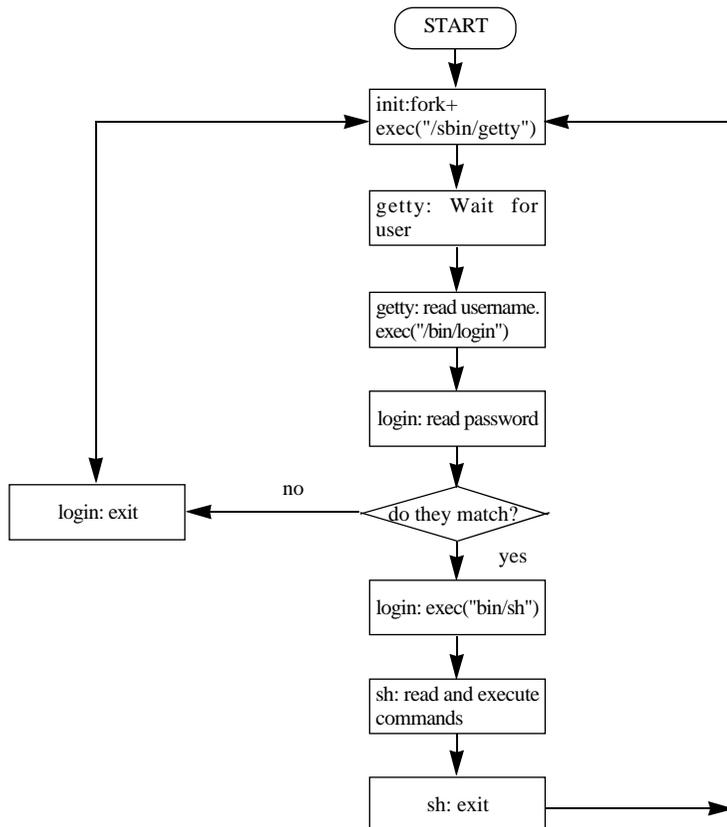
警告在系统关闭前将自动重复数遍，随着时间流逝，间隔越来越短。当延时之后关闭系统真正开始时，所有文件系统(除了根)被unmount，所有用户进程(如果有人还未注销)被终止，守候进程被关闭，所有东西都停下来。此后，init打印出一条信息告知你可以关掉电源了。此时，也只有在此时，你才可以关闭电源。有时(虽然在任何好的系统上极少)，系统可能不能正常关闭。例如，核心紊乱、崩溃等不正常情况，可能无法键入任何命令，因此正常关机可能有些困难，这是只能直接关机。问题可能没那么严重，比如，有人误动了你的键盘，核心和update程序还在正常运行，等待一些时间可能是个好建议，这能使update有机会将缓冲cache中的数据回存硬盘，然后再直接关机。

有人喜欢用sync三遍来关闭系统，等到磁盘I/O停止，然后再关闭电源。如果没有什么程序运行着，这和用shutdown等效。然而，它不unmount任何文件系统，可能导致ext2fs的“干净文件系统”标志出问题。这种3遍sync的方法是不推荐使用的。

## 4.7 登录和注销

### 4.7.1 通过终端登录

图4-6显示了通过终端登录发生的事情。首先，init确认有一个getty程序提供给终端连接



通过终端登录：init、getty、login和shell的交互

图 4-6

(或控制台)。getty 侦听终端等候用户告知它要登录(这通常意味着用户必然键入些什么)。当它注意到一个用户, getty 输出一个欢迎信息(存在/etc/issue中), 并提示用户名, 最后运行 login 程序。login 作为一个参数得到用户名, 并提示用户输入口令。如果正确, login启动给此用户设置的 shell; 否则退出并终止进程(可能再给用户一个机会输入用户名和口令之后)。init 注意到进程终止, 就给这个终端启动一个新的 getty。

注意, 惟一的新进程是由 init 产生的(用fork 系统调用); getty 和login 只是替代进程运行的程序(使用exec 系统调用)。

为注意用户, 串行线需要一个单独的程序, 因为终端活动时(传统上也是)变得复杂。getty 也适应连接的速度和其他设置, 这对拨号连接特别重要, 因为连接和连接的参数可能不同。

getty和init有多个版本在使用, 各有优缺点。学习你的系统的版本也了解其他版本是个好主意(可以用 Linux Software Map来找)。如果你没有拨入, 可能不必考虑 getty, 但 init 仍然很重要。

#### 4.7.2 通过网络登录

一个网络中的两台计算机通常通过一个物理电缆连接。当通过网络通信时, 参与通信的每个计算机里的程序通过虚拟连接 virtual connection通信, 即一些虚构的电缆。虚拟连接的每端的程序, 独占自己的(虚拟)电缆。然而, 因为这电缆不是真的, 只是虚构的, 所有计算机的操作系统可以在同一物理电缆上有多条虚拟连接。这样, 只用一条电缆, 多个程序可以不必考虑其他通信而相互通信。使用同一电缆使多台计算机通信是可能的; 两台计算机间存在的虚拟连接, 其他计算机会忽略它们不参加的连接。

那是一个复杂和抽象的真实描述。但可能足够理解网络登录与普通登录的不同的重要原因。不同计算机上的两个程序要通信时, 虚拟连接建立。由于理论上可能从网络上的任何一台计算机登录到任何一台计算机, 因此可能有极大数量的潜在的虚拟通信。因此, 为每个潜在的login启动一个getty 是不现实的。

有一个进程inetd(与getty 协同)处理所有的网络登录。当它发现一个进来的网络登录(即发现某台其他计算机来的新的虚拟连接), 它启动一个新进程来处理那个登录。原来的进程继续侦听新的登录。

更复杂的是, 网络登录有多个通信协议。两个最重要的协议是 telnet 和rlogin。除了登录, 还有许多其他虚拟连接可能建立(为FTP、Gopher、HTTP和其他网络服务)。为给侦听的每种类型的连接提供一个更有效的进程, 只用一个侦听器(inetd)来识别连接的种类, 并启动正确的程序来提供服务。

#### 4.7.3 login干了些什么

login 程序负责认证用户(确认用户名和口令相配), 并建立串行线, 启动 shell, 建立用户的初始环境。

部分初始化设置是输出文件 /etc/motd(每天的短信息)的内容, 并检查电子邮件。可以在用户家(home)目录中产生一个叫.hushlogin 的文件来使上面所述的失效。

如果存在文件/etc/nologin, 就不允许登录。这个文件一般由 shutdown 及其相关的东西产

生。login 检查这个文件，如果这个文件存在，就拒绝接受登录。如果这个文件确实存在，login 就会在退出之前，将它的内容输出到终端。

login 将所有失败的登录企图登记在系统 log 文件中（通过 syslog）。它也登记所有的 root 的登录。这些都对跟踪入侵者有用。

当前登录着的用户列在 /var/run/utmp 中。这个文件直到系统下次启动或关机前有效。系统刚启动时它被清空。它列出了每个用户和用户使用的终端（或网络连接），以及一些有用的信息。who、w 及其他类似的命令可以查看 utmp 文件都有谁登录着。

所有成功的登录记录在 /var/log/wtmp 中。这个文件将无限制地增大，所以必须有规律地清除，例如有个每周的 cron 任务来清除它。last 命令浏览 wtmp 文件。

#### 4.7.4 存取控制

用户数据库传统上包含在 /etc/passwd 文件中。有些系统使用影子口令 shadow passwords，并把口令移到 /etc/shadow 中。许多计算机的场所可以用 NIS 或其他存储用户数据库的方法共享帐户，它们可能也自动从中心位置复制数据库到所有其他计算机。

用户数据库不仅包含口令，还包括有用户的其他信息，比如其真实姓名、家目录、登录 shell 等。其他信息需要公用，使所有人都能读。因此口令是加密保存的，它的缺点是，任何人取得加密的口令，可以用不同的加密方法猜试口令，而不用试着真正登录到计算机。影子口令试图用把口令移动到其他文件的办法避免这种情况，只有 root 能读（口令还是加密保存的）。

不管有没有口令，确认系统中的所有口令是好的是很重要的，即不易猜。crack 程序可用于破解口令；任何可以精确地找到的口令都不是好的口令。同时 crack 可以为入侵者运行，也可由系统管理员运行，以避免坏的口令。好的口令也可以被 passwd 程序强制实现；这样对 CPU 周期来说很有效，因为破解口令需要许多计算。

用户组数据库保存在 /etc/group 文件中；有影子口令的系统，是 /etc/shadow/group。root 通常不能通过更多的终端或网络登录，只能通过列在 /etc/securetty 文件中的终端登录。这使得必须能够物理存取这其中的一个终端。当然也可能通过任何终端用任何用户登录，然后使用 su 命令变成 root。

#### 4.7.5 Shell 启动

当一个交互的登录 shell 启动时，它自动执行一个或更多预定义的文件。不同的 shell 执行不同的文件；更多的信息见每个 shell 的文档。

多数 shell 首先运行一些全局文件，例如， Bourne shell (/bin/sh) 和它引出执行的 /etc/profile；另外，它们执行用户家目录中的 .profile。/etc/profile 允许系统管理员建立一个公用的用户环境，特别是建立 PATH，以包括本地命令目录。另外，.profile 允许用户通过覆盖按照自己的口味客户化环境，如果必要，使用缺省环境。

### 4.8 管理用户

系统管理员本身的职责就是负责整个 Linux 系统的运行稳定，增加系统安全性，校验使用者的身份，新增使用者或删除来路不明或会危害系统安全的使用者，并明确定义每一个在机

器上的使用者权限等等。所以管理者需对系统付出相当大的心血，来管理好一个系统。系统管理员的部分工作是与人打交道。这些人是另外一些负责你站点的管理员（如果你不是单独工作的话），或者负责其他站点的管理员和使用你站点的用户。最难打交道的是用户，因为它们之间的绝大多数的思想沟通中包含着抱怨和要求。也许你可以保留所有的表扬和感谢以便保持高昂的士气。

在处理用户请求时记住以下几点：当安装好 Linux 之后，整台机器的惟一的使用者，也就是权限最大的 root。如果 Linux 机器的用途是个人使用，和外界没有任何联系，机器上只有一个用户。在这种情况下，考虑用户帐号和系统管理员义务也许会觉得多余。

不能以这种方式去考虑问题，不管是在家中，在一台单用户的个人计算机上，还是在大型机关中，在一台拥有 150 个用户的与 Internet 全面连接的服务器上，只要你有 root 帐号，就必须慎重对待。你可能为一些同学或亲朋好友建立 Linux 机器上的帐号，让大家共同使用 Linux 机器。如果是用于商业用途，如公司给予每一个员工一个 Linux 的使用帐号时，就应该对每一个使用者先做身份确认的操作，让他们填个帐号申请单之类的文件，确定身份无误后，才使用 adduser 指令为他们建立帐号。

作为系统管理员，必须以普通用户和守护人双重身份在机器上工作。用普通用户的身份去完成所有日常工作和不需要特权用户帐号就能完成的系统管理工作。只有当绝对必要时，才以守护人的身份去完成系统管理工作。当你拥有一台机器，尤其是一台和其他用户一起工作的机器的 root 帐号时，你就有义务和责任。不能掉以轻心，不能由怀有不良意图的人利用你的特权。特别是当你建立有自己的 Internet 网站，如 WWW、BBS、FTP 时，如何管理好登录你机器的帐号，也不是一件轻松的事情。

#### 4.8.1 系统管理员

Linux 是一个多用户系统，在 Linux 中，一般用户只管运行自己的应用程序，属于系统支持方面的工作，则由被指定为系统管理员的人来负责。通常，系统管理员都是专职的，其主要工作为：

- 1) 格式化磁盘。
- 2) 建立或取消用户的户头。
- 3) 保证所有的硬件都能正常工作。
- 4) 维护文件系统，为用户提供磁盘空间。
- 5) 把本系统和其他的 Linux 系统建立连接。
- 6) 防止未经授权的用户侵入系统。
- 7) 负责监督系统的使用并测试系统性能。
- 8) 负责记帐和维护系统安全。

在多用户系统中，系统管理员就是 root 用户——超级用户。他要负责给每一个合法的用户提供户头，并必须记录用户的活动情况。

##### 1. 系统管理员的登录

在 Linux 系统中，系统管理员有一个专用的帐户。开机后，系统初始化完毕，出现提示符 login：，要求输入你的用户帐号。你可以在 login：后输入 boot 代表以系统管理员登录，键入密码后即可进入系统。例如：

```
$ login root
Password: *****
#
```

如果已作为一个一般用户登录了，则可以通过替换用户命令 (su) 进入系统管理员的户头。  
例如：

```
$ su
Password: *****
#
```

不管是如何进入系统管理员的户头，shell都将显示一个特别提示符(#)，这个提示符表明用户已不再是作为一般用户进入系统的，它已成为超级用户——一个具有系统管理员的巨大权限的用户。

超级用户可以对系统中的任何文件和目录进行访问，作为超级用户注册后，便可在系统中不受限制地工作。鉴于超级用户的巨大权限，在一个Linux系统中，只应有一个人作为超级用户。即使有超级用户的权限，也不应随便使用。一般情况下，以普通用户登录，只有在要进行管理工作时，再以超级用户的身份进入。

## 2. 系统管理员设置目录

目录/etc/是专为系统管理员(或超级用户)而设置的，含有系统管理所需的绝大多数配置文件和命令。一般命令放在/bin和/usr/bin中，可用完整的路径名来指定任何命令，若只想输入命令名而省去路径的话，则可在初始化文件中放入下列路径设置命令。该初始化文件在Borune shell中为.profile，在C shell中则为.login。

在Borune shell中加入如下命令：

```
PATH = /etc/bin:/usr/bin:$HOME/bin:
export PATH
```

在C shell中加入如下命令：

```
Set path = (/etc/bin /usr/bin $HOME/bin.)
```

## 3. 系统的初始化文件

正如每个用户(包含超级用户)都有一个初始化文件一样，每个Linux系统也有一个系统的初始化文件。这就是/etc/rc.rc的含义为“执行命令(run commands)”，这个词经常在Linux中使用。正如用户进入系统时要执行放在.profile或.login文件中的语句一样，文件中的语句要在整个系统引导时执行。

## 4. 给用户发送消息

同系统相连的用户有时可能是分在许多不同地方的。而作为系统管理员，需要不时地同他们联系，mail命令虽然可完成这项工作，但速度太慢。下列命令是比较好的选择：

wall发往所有用户的命令(Write—All)；

news命令；

/etc/motd每日消息(Message Of The Day)；

命令wall同命令write类似，所发消息会很快传送给已登录的用户。

例如：发往所有用户的命令。

```
# wall
The system will be down for maintenance at 3:00
```

```
Ctrl + D
```

也可以把所要发送的消息放到 /usr/new 文件中，然后再使用命令将它送给用户。

另外选择时使用每日消息文件 /etc/motd。存放在此文件中的消息，每次在用户登录时都会自动显示在用户的屏幕上。

例如：使用每日消息文件的例子。

```
# cat /etc/motd
All users will have to logout by 6:00 pm on Wednesday
#
```

作为用户，为答复来自系统管理员的消息，可用电子邮件把消息送给 root，如：

```
$ mail root answer-sys
```

在这个例子中，文件 answer-sys 中的消息由命令 mail 邮递给了系统管理员。

#### 4.8.2 管理职责

对于一个 Internet 站点上的系统管理员来说，在站点上发生的大多数事情都是由用户完成的，因此要进行的管理工作绝大部分与用户有关。当站点上有一个定期改变的用户库时更是如此，因此，如何更好地管理用户，更大地发挥效率，是每一个系统管理员应该掌握的知识。

##### 1. 与用户交互

1) 要敬业。作为 Web 站点管理员也许是你所喜欢的工作，但它同样也是一种职业。如果你发现不喜欢某个用户或者发现难于与之相处。最佳的策略就是保持微笑、保持友好的态度并高效地完成工作。问题解决得越快，就可越快地转做其他事情。

2) 要有耐心。你可能会遇到技术比较差，对自己能力不敢确信或者对计算机一无所知的用户。通常在与这些用户打交道时需要从他们那里提取所需信息以解决它们的问题。可以使用一种简单的方法，里面包含一些需要用户回答的问题（例如，用户运行的系统平台，什么软件发生了问题，所见的错误信息）。

3) 反应敏捷。对于用户来说，一个严重的错误需要等待数周甚至几个月的时间才能获得解答是一件令人沮丧的事情。

4) 保持坚定的立场。

5) 保持权威性。

##### 2. 处理引起问题的用户

在你的站点上处理引起问题的用户可能是比较困难的工作之一。需要耐心、公正、老练和决断。有许多由于这些用户的挑战而需要处理的理由。

你要确定用户确有问題。毕竟，帐户也许只是被盗用，或者用户的抱怨并非反映真实的情况。要一些证明有问题的证据是有帮助的。证据包括日志、E-mail、文件等。

用户可能危害正当的操作，如果你在规则中指明了什么样的事情不允许，或不可容忍，以及违反者将被如何处置，你将有足够的理由。

一些有问题的用户是恶意的而且有丰富的计算机知识，这些用户有能力损害你的系统。因此，应该在通知用户后立即删去其帐户。

### 3. 删除或挂起用户

作为系统管理员，在有些情况下需要删除用户。

1) 如果一名用户通知你他想要在月末取消其帐户，应该通知他你什么时候删除其主目录下的所有数据。一旦这个日期到来，你可以期望用户已经将其所需保留的所有文件拷贝下来了。随后可以开始通过删除相应的用户主目录来完全删去帐户，帐户的提示信息保留在文件 `/etc/passwd` 和文件 `/etc/shadow` 中(如果你使用它的话)。

2) 挂起对用户的访问是很简单的。进入 `/etc/passwd` 文件或者 `/etc/shadow` 文件中，并在用户口令的开始加上星号(\*)。处理有问题的用户要立即，尤其是那些恶意的并且具有丰富的计算机知识的用户。不要冒破坏站点的危险。如果你最终决定删除用户，请按上述的过程进行。

### 4.8.3 用户管理

Linux是一个多用户系统。既然是多用户系统，那么如何识别每一个登录到系统的用户呢？这就牵涉到帐号的问题。当一台计算机可以被很多人使用时，为了使个人文件保持私有化，每个用户给定一个单独的用户名，这个被用于登录到系统的用户名就是帐号。并且为了不让别人使用自己的帐号，每个帐号同时又对应着一个口令，这就构成了用户管理的基本结构。

另外就是每一个使用者(包含root)对自己文件的权限管理。因为Linux是多用户多任务系统，每一个使用者都有可能将其工作的内容或是一些机密性的文件放在Linux工作站上，所以各个文件或目录的归属和使用权，都要有非常明确的规定。举例来说，如果在某私人机构或是公家机构，一位主管把他的机密文件误设为任何人(other)可读可写，则该文件将会很容易地被该机器上的其他使用者取得，甚至任意地修改或破坏而造成公司机密外泄。因此每一个使用者对于自己目录(Home)的文件都应该非常清楚地给予它们合适的权限，才能维护自己本身的文件安全，防止被窃取之类的事情发生。说了这么多，主要目的就是要告诉读者，用户管理非常的重要，所以好的系统管理员就需要掌握Linux系统中用户管理的基本方法和技巧。

#### 1. 添加用户口令文件

每个用户需要有自己的帐号才能进入系统操作，所以添加用户帐号是最常见的用户管理操作。达到这个目的有两种办法：一是直接使用添加用户的命令，二是修改相关文件。

##### (1) 使用useradd命令添加用户

该命令开关较多，如果不指定则按默认值处理。

```
Useradd [-c comment] [-d home_dir] [-e expire_date] [-f inactive_time] [-g initial_group] [-Ggroup[...]] [-m] [-k skeleton_dir] [-s shell] [-u uid[-o]] [-n] [-r] login
```

```
Useradd -D [-g default_group] [-b default_home] [-f default_inactive] [-e default_expire_date] [-s default_shell]
```

上面是两种常用的格式，不带-D选项的格式将在 `/etc/passwd` 中增加一条新记录，并且添加一个新用户。如果使用了 `-m` 选项将建立一个新目录。在加入该用户以后，还需要使用 `passwd` 命令设置口令，来使该用户有效。

如果使用 `-D` 选项时，`useradd` 命令将显示系统设置的有关默认值；如果不指定这些值，则

系统将使用默认属性对新建立用户进行设置。如果在命令行中指定了 `-g`、`-f`等选项，则可以改变相应默认值。

- `-c` comment 对被创建用户的说明。
- `-d` home\_dir 用户的起始目录。
- `-e` expire\_date 用户的过期时间，在此时间之前允许使用该帐户，过期后自动作废。日期格式为 mm/dd/yy，即月/日/年。
- `-f` 从该用户帐号有效算起，在 inactive 设置的天数内，如果用户没有登录过该帐号，那么此帐号也就失效。
- `-g` 指定用户所属的组。该组必须已经存在，默认的组编号为 1。
- `-G` 设置用户所属的其他附加组。每两个组之间使用 ‘,’ 号分隔，不能插入空格。默认值为没有附加组。
- `-m` 如果用户的起始目录不存在，就建立一个。
- `-k` skeleton\_dir 指定用户的缺省配置文件所在目录。该目录的内容将被复制到用户的起始目录中，如果不指定，则使用 /etc/skel 中的设置。该选项一定要和 `-m` 组合使用。
- `-M` 不创建用户起始目录。
- `-n` 建立一个和所建用户名同名的组。
- `-r` 创建一个系统帐号。就是说，该用户的 UID 比 /etc/login.defs 文件中 UID-MIN 所定义的最小值还要小。Useradd 不会为这种用户创建起始目录，而不管 /etc/login.defs 中的默认值如何。用户必须使用 `-m` 来指定创建新目录。
- `-s` 指定用户使用的 shell 名称。如果不指定将使用系统默认值。
- `-u` 为用户分配用户标示。如果不使用 `-o` 选项，则该值不得与其他现存 UID 重复，且值为非负。默认值是比 99 大，且比任何已有用户的 UID 大。0~99 是为系统帐号保留的。
- `-b` default\_home 用户初始目录的缺省父目录。如果创建用户没有使用 `-d` 选项，则用户的初始目录是 /default\_home/Login\_name。
- `-f` default\_inactive 指定在帐号有效后若不登录系统则取消帐号的天数。
- `-g` default\_group 指定用户所属组的名称。
- `-s` default\_shell 指定用户登录的 shell。

如果没有指定上述参数，则 # useradd 将显示所有的缺省值。

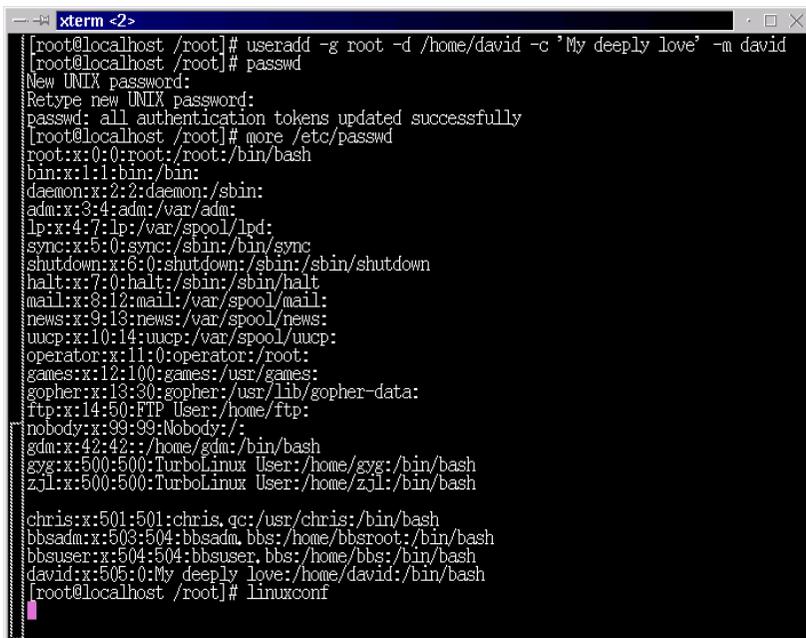
例如：添加一个名为 benson 的帐号，属于 622 组，设定起始目录为 /home/benson。键入命令如图 4-7 所示。

这时可以看出，在 /etc/passwd 文件的最后一行，新创建的 benson 用户的信息已经成功写入到 /etc/passwd 文件中。

要记住，useradd 命令并不为用户设置口令。必须使用 passwd 命令进行设置后，该用户才可以正式使用。

## (2) 手工添加用户

这种办法适合于那些对 Linux 系统已经比较熟悉的用户。其实刚才使用的添加用户的命令也就是往一些文件中添加条目，建立新目录，复制一些文件。已授权用户的信息存放在文件 /etc/passwd 中，所以如果直接往 /etc/passwd 文件中写入相应的条目，并且建立相应的目录，一样可以达到相应的效果。



```
root@localhost /root]# useradd -g root -d /home/david -c 'My deeply love' -m david
root@localhost /root]# passwd
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
root@localhost /root]# more /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
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:
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:
operator:x:11:0:operator:/root:
games:x:12:100:games:/usr/games:
gopher:x:13:30:gopher:/usr/lib/gopher-data:
ftp:x:14:50:FTP User:/home/ftp:
nobody:x:99:99:Nobody:/:
gdm:x:42:42::/home/gdm:/bin/bash
gyg:x:500:500:turbolinux User:/home/gyg:/bin/bash
zjl:x:500:500:turbolinux User:/home/zjl:/bin/bash

chris:x:501:501:chris,gc:/usr/chris:/bin/bash
bbsadm:x:503:504:bbsadm,bbs:/home/bbsroot:/bin/bash
bbsuser:x:504:504:bbsuser,bbs:/home/bbs:/bin/bash
david:x:505:0:My deeply love:/home/david:/bin/bash
root@localhost /root]# linuxconf
```

useradd和passwd命令执行情况

图 4-7

Linux系统的基本用户数据库是文本文件，/etc/passwd文件，也就是口令文件。它列出所有有效用户名及其相关信息。每个用户的信息占据文件的一行，每行分为用“:”分隔的7个域：

- 用户注册名 该名称是用户在终端登录到系统时输入的名称，同一系统中用户名应该是唯一的。其长度根据不同的系统而定，一般是八位，就算是超过八位的用户名，也只有前八位才有效。
- 用户的口令 由于在该系统中另外还采用了/etc/shadow文件，也就是说将加密后的口令放到了/etc/shadow文件中，所以在/etc/passwd文件中口令这一项使用一个“x”来表示。如果用户没有设置口令，则该项为空。
- 用户标识(UID) 登录帐号是从系统外来标识用户的，而系统内部是用另外一个数值来标识的，那就是用户标识(UID)。标识号分配是有一定规则的。在分配上首先将用户分为系统用户和普通用户两类。系统用户是安装系统就生成的(当然也可以后来加入)，一般都肩负有管理一部分系统的任务。其中最特殊的一个用户自然就是常说到的超级用户——root。之所以为超级用户，就是因为该用户拥有对系统的一切操作权限，所以使用时更是要万分小心。该用户的UID为0，而所有的系统用户的UID都在100之下，普通用户的UID就要从100开始往上编号了。由于在系统内部UID占有两个字节，所以最大用户标识只能达到32767。UID并不惟一，也就是说可以有几个用户使用同一个UID。这样一来，系统认为这些用户都是一个用户，只是使用不同的用户标识而已。
- 组标识(GID) 用户所属组的标识(GID)，意义大致和UID一样，只是对于组而已。组标识也是一个从0~32767之间的整数。
- 用户全名或者本帐号的其他说明 该域记录了用户本身的一些情况。该域由于是历史原因产生的，所以现在也就没有什么太大用处了。

- 起始目录 该域是用户的起始目录名。用户登录系统以后将首先进入该目录，用户的一切文件等也将存放在该目录中。
- 登录使用 shell(登录时运行的程序) 所谓shell其实就是对用户输入的命令进行解释和执行的程序，比如DOS操作系统中的command.com。在Linux中，有各种各样的shell可以让用户选择使用。每种shell都有各自的特点，但完成的基本功能是相同的。常用的shell有：/bin/sh(Bourne Shell)、/bin/csh(C Shell)、/bin/ksh(Korn Shell)等。

在了解了/etc/passwd文件的条目构成以后，用户就可以直接编辑该文件来添加用户了，一般情况下分为以下几个步骤：

- 1) 编辑/etc/passwd文件，添加相应的用户条目；
- 2) 在需要建立用户起始目录的地方建立用户目录；
- 3) 将一些用户配置文件复制到用户起始目录下；
- 4) 其他一些设置和修改。

完成上述步骤后，就产生了一个可以使用的新用户了。

## 2. 设置口令

在建立一个新用户以后还要为该用户设置口令，以免其他非法入侵者对系统造成破坏。设置口令是一件非常重要的事情，无论是对用户自己还是对于系统。要设置一个好的口令，不容易为他人获得或者破解，大致有以下几点需要注意：

- 不要使用与自己有关系的单词、数字等等作为口令。
- 口令应该尽可能长，每长一位，破解的可能性就会降低好几个数量级。
- 不同地方的登录口令不要相同。
- 在网络上传递口令时一定要加密，最好是不要通过非面授的方式传递口令。
- 口令应该经常更换。

设置口令的命令是passwd，但是只有超级用户才能使用该命令。命令格式为

```
# passwd [-u] [user_name]
```

使用该命令的过程如图4-7所示。

如果某个用户忘记了自己的口令，那么就只好由超级用户先删除掉/etc/passwd中的口令域，重新设置。

## 3. 删除用户

删除用户和添加用户是相对的，既然添加用户有两种不同的途径，那么删除用户也有两种途径。其一是通过userdel命令删除；其二是手工删除，即通过修改相关文件达到目的。

### (1) 使用userdel命令删除用户

使用命令删除用户十分简单，命令格式如下：

```
# userdel [-r] user_name
```

其中-r是指将该用户的起始目录及其中的文件全部删除。

### (2) 手工删除用户

删除一个用户包括几个部分，按如下步骤：

- 1) 首先从/etc/passwd文件中删除该用户的相关条目。
- 2) 然后删除该用户的目录及所有相关文件。

如果只是暂时封冻该用户，则只需在/etc/passwd文件的该用户的口令域中添加一个\*号注

释掉即可。以后需要解冻的时候，将\*号去掉。

#### 4.8.4 用户组管理

每个用户都属于特定的用户组，而用户组则是一些具有相同特性的用户的集合。Linux是多用户操作系统，它根据各个用户所享有的文件权限的不同而分为不同的用户组。一个用户至少属于一个用户组，该组就是用户的基本组，但同时还可以属于其他很多的附加组。用户在系统中某一时刻所属组为其当前组，也可以使用 `newgrp` 命令来切换所属组。

##### 1. 添加用户组

和添加新用户相似，添加用户组也有两种不同途径。

##### (1) 使用 `groupadd` 命令添加用户组

其命令格式为：

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

- -g gid

设定组标示。如果不使用 `-o` 选项，则组标示 (GID) 必须是惟一且非负的。默认值是比 500 大且比其他任何现存组的组标示大。0~499 是为系统用户保留的。

- -o

此选项是和 `-g` 选项一起使用的。如果使用该选项，则允许使用相同的组标示 (GID)。

- -r

该选项使 `groupadd` 添加一个系统用户。如果不使用 `-g` 选项则第一个小于 499 的 GID 将被分配给用户组。

- -f

这是一个强制选项。当用户试图建立一个已经存在的组的时候，`groupadd` 将会停止并返回一个错误信息，如果使用了该选项则不返回错误信息。

##### (2) 手工添加用户组

其实添加组的过程就是在 `/etc/group` 文件中添加一项而已，既然使用命令添加用户组只是在文件中进行修改，那么也就可以通过直接修改 `/etc/group` 文件来达到目的。下面先介绍一下该文件的结构。

该文件每一行就代表一个用户组，每一行由四个域组成，其间用“:”分隔开。图 4-8 显示了 `/etc/group` 文件的结构。

- 用户组名

该域指出的就是用户组的名称。

- 用户组口令

该域显示的是经过加密后的用户组口令。设置过口令的用户组，当其他非本组用户需要进入的时候要输入口令。

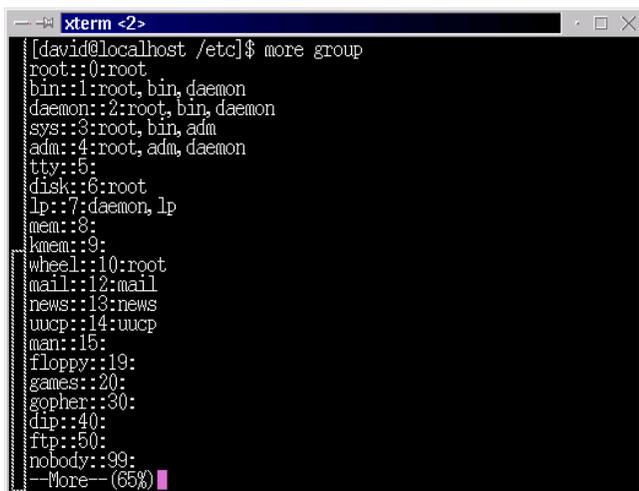
- GID

用户组标识，用于系统识别一个用户组。GID 相同，系统将认为这是同一个组。所以如果几个组使用同一个 GID，系统将认为这是同一个组。

- 以该组为附加组的用户

显示有哪些用户将此用户组作为附加组。

在了解该文件结构以后，增加一个用户组其实也就是在 `/etc/group` 文件中添加个新条目而已。具体的操作请读者参考文件自己加入。



```
xterm <2>
[david@localhost /etc]$ more group
root::0:root
bin::1:root, bin, daemon
daemon::2:root, bin, daemon
sys::3:root, bin, adm
adm::4:root, adm, daemon
tty::5:
disk::6:root
lp::7:daemon, lp
mem::8:
kmem::9:
wheel::10:root
mail::12:mail
news::13:news
uucp::14:uucp
man::15:
floppy::19:
games::20:
gopher::30:
dip::40:
ftp::50:
nobody::99:
--More-- (65%)
```

图 4-8

## 2. 为用户组设置口令

当为新用户设置口令时，可以使用 `passwd` 命令；当为用户组设置口令时就没有现成的命令了，需要进行一些操作，但还是需要利用 `passwd` 命令。下面介绍一下设置用户组口令的步骤。

### (1) 首先添加一个新用户

假设添加一个用户名为 `david` 的用户(只是假设，其实随便取什么名字都是可以的)：

```
# useradd -m david
```

### (2) 为该用户设置口令

命令格式为：

```
# passwd david
```

### (3) 查找加密后的口令

在 `/etc/shadow` 中查找 “ `david` ” 字符串所对应的加密口令，并将其输出到一个文件中。其命令格式为：

```
# grep /etc/shadow > david.tmp
```

### (4) 编辑 `/etc/group` 文件

将 `david.tmp` 文件中的字符串复制到 `/etc/group` 文件中需要加密的用户组的第二个域中，这就是设置好的口令。

### (5) 删除用户

将添加的临时用户删除，将临时文件删除。

要注意的是，有的 Linux 系统有一个 `/etc/gshadow` 的文件，那么字符串就应该复制到该文件中的对应部分。

## 3. 删除用户组

删除用户组命令很简单，同样也是两种途径：一是使用 `groupdel` 命令；二是直接编辑

/etc/group文件，删除对应条目。

(1) 使用groupdel命令删除用户组

其命令格式为：

```
groupdel group
```

(2) 手工删除用户组

编辑/etc / group文件，将牵涉所有将要删除组的条目通通删除干净。

#### 4. 修改用户组属性

当已经建立好了一个新的用户组以后，却发现需要对其中的一些属性进行修改，该怎么办呢？这时就要求助于groupmod命令了，该命令就是用于改变用户组属性的。groupmod命令格式如下：

```
groupmod [-g gid[-o]] [-n group_name] [group]
```

其中-g选项是设置用户组标识，-o是允许使用相同的GID，-n是修改用户组名。

不过如果需要改变用户组的口令时，还得参照上面所介绍的步骤来进行。而且如果需要修改附加组的有关信息，也只能通过修改/etc/group文件的第四个域来实现。

### 4.8.5 使用LinuxConf管理用户

在上面的小节中，我们介绍了用户管理的一些常用命令、用户组的概念和命令。读完之后，读者是不是对我们介绍的有关Linux的用户管理和用户组的管理的命令的复杂性觉得有些难以掌握呢？不错，为了建立一个新用户，我们要在命令行下为它建立用户名，分配 Home Directory，修改口令，设定用户组，这是够麻烦的啦！

幸运的是，TurboLinux 4.0简体中文版包括了一个易于使用的集成工具LinuxConf，你可以通过使用LinuxConf轻易地在一个界面中完成上面我们在命令行下按步骤完成的所有操作。是不是很想看一看如何使用它？很简单，键入：`# LinuxConf`，出现界面，如图4-9所示。

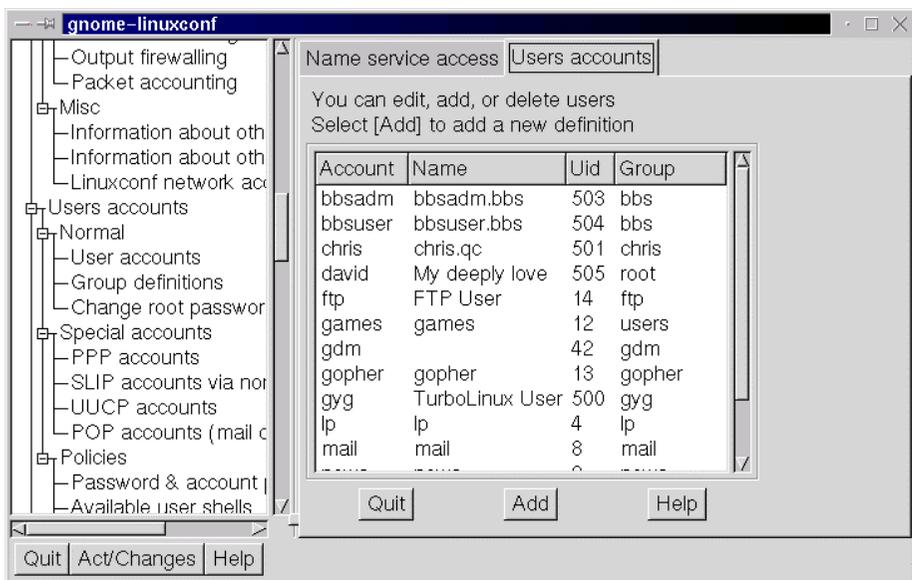


图 4-9

使用鼠标操作，拉动图4-9中左边的竖形条，可以看到 Users accounts这一项，如果它的左端为+，点击+，在 Users accounts下出现上图所示的 Normal、Special accounts和Policies子项，各子项下有各自的功能项。用鼠标点击选择你需要的功能项，同时图的右面出现所选的功能的操作界面。

图4-9是 Users accounts Normal User accounts的界面操作图，在图的右面，可以看到系统中所有的 User account。如果需要添加一个新用户，点击按钮 Add，出现界面如图4-10所示。

在 login name中输入新用户的名称， Full name中输入用户全称， group中输入用户所属的组名称，在 supplementary groups中输入补充的组名，在 Home directory中输入此用户的起始目录，在 Command interpreter中可选择命令解释 shell，可选的 shell 有 /bin/bash、/bin/ash、/bin/等，选择你需要的即可。所有选项选择完毕后，点击按钮 Accept，如果输入没有错误，出现输入用户口令界面如图4-11所示。

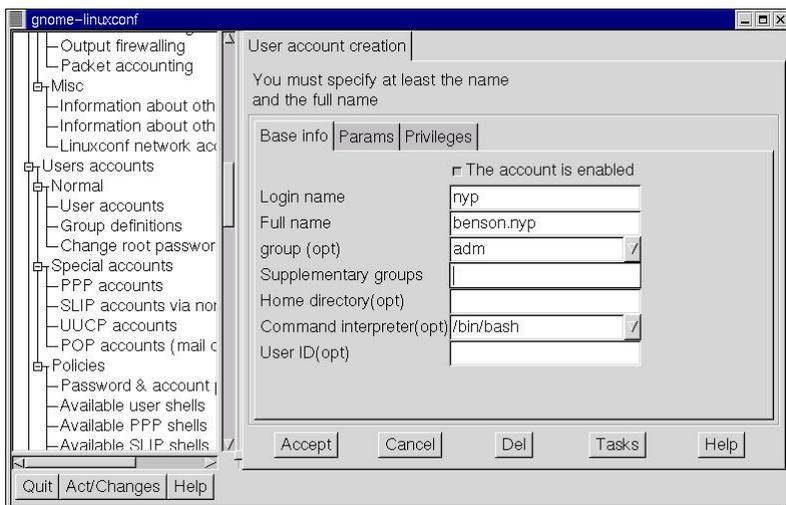


图 4-10

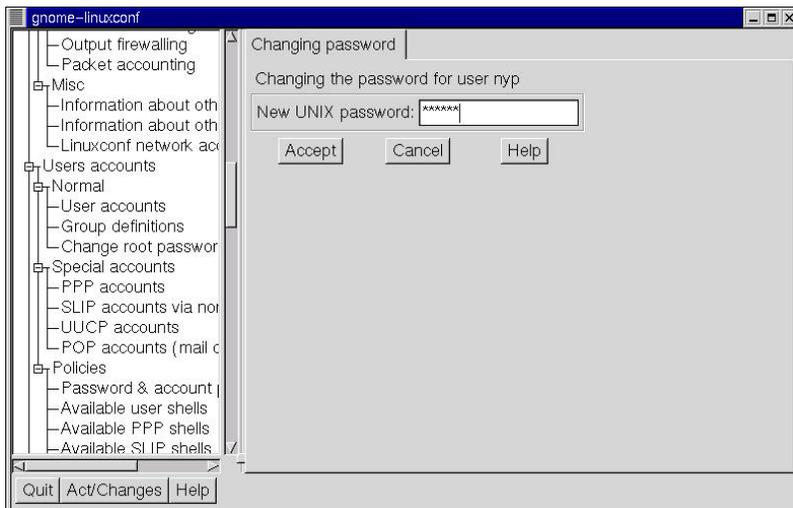


图 4-11

在New UNIX password中输入新建用户的密码，密码不小于6位，输入后点击按钮Accept，然后要求你再次输入密码以确定界面将出现，再次键入密码。两次输入密码无误后，点击按钮Accept，系统将接受你建立的新用户并出现界面如图4-12所示。在此图中，将看到新用户nyp已经出现在User account中，表明你的新用户已经成功建立。

同上面的步骤相似，删除用户，修改用户口令，建立新用户组等用户管理方面的操作都可以在Linuxconf中完成，非常方便。

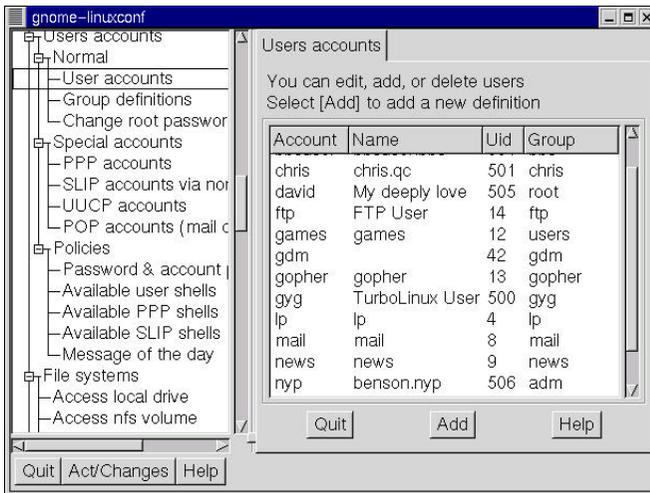


图 4-12

## 4.9 内存管理

### 4.9.1 什么是虚拟内存

Linux支持虚拟内存，就是使用磁盘作为RAM的扩展，使可用内存相应地有效扩大。核心把当前不用的内存块存到硬盘，腾出内存给其他目的。当原来的内容又要使用时，再读回内存。这对用户全透明：运行于Linux的程序只看到大量的可用内存而不知哪部分在磁盘上。当然，读写硬盘比真的内存慢（慢千倍），所以程序运行较慢。用作虚拟内存的这部分硬盘叫对换空间。

Linux可以使用文件系统中的普通文件或单独的分区作为对换空间。对换分区更快，但对换文件更易于改变大小（无须对硬盘重分区）。如果知道要多少对换空间，应该用对换分区；如果不能确认，可以先用对换文件，用一段时间后再根据所需空间建立对换分区。

Linux允许同时使用多个对换分区和/或对换文件。即如果偶尔需要更多的对换空间，可以随时建立一个额外的对换文件。

### 4.9.2 产生对换空间

对换文件是普通文件，对核心没有什么特别的。惟一不同是它没有孔，用mkswap准备。必须在本地盘上，不能在通过NFS mount的文件系统中。

孔是很重要的。对换文件保留了磁盘空间，使核心能快速对换出一页，而不必经过如文

件的定位磁盘扇区的全部事情。核心只用分配给这个文件的所有扇区。由于文件中的孔意味着没有为文件中这个位置分配磁盘扇区，这对核心使用不利。产生没有孔的对换文件的一个好办法是通过如下命令：

```
$ dd if=/dev/zero of=/extra-swap bs=1024 count=1024
1024+0 records in
1024+0 records out
$
```

/extra-swap是对换文件名，大小由 count=给出，大小最好是4的倍数，因为核心写出的内存页是4KB。如果不是4的倍数，最后那几KB将不可用。

对换分区也没什么特别。就像产生其他分区一样产生；惟一的不同是它作为原始分区使用，即没有任何文件系统，最好将对换分区标记为类型 82(Linux swap)，虽然这对核心没有影响，但这使分区列表更清晰。

产生对换文件或对换分区后，需要写个标记起用它，这包括核心要用的一些管理信息。命令是 mkswap ，用法如下：

```
$ mkswap /extra-swap 1024
Setting up swapspace, size = 1044480 bytes
$
```

注意，对换空间现在还没用，它存在，但核心还没用它提供虚拟内存。

请一定小心使用 mkswap ，因为它不检查文件或分区是否被其他东西使用。你可能用 mkswap很容易地覆盖了重要文件和分区！幸好，你只需在安装系统时使用 mkswap。

Linux内存管理限制了每个对换空间约为 127MB(由于技术原因，实际限制为  $(4096 - 10) \times 8 \times 4096 = 133\,890\,048$  字节或者 127.7M字节。可以同时使用最多 16个对换空间，总计差不多 2GB。

### 4.9.3 使用对换空间

用swapon将一个初始化的对换空间可用。此命令告诉核心对换空间可以用了，对换空间的路径作为参数，启动一个临时对换文件可以用如下命令：

```
$ swapon /extra-swap
$
```

对换空间如果列入/etc/fstab ，就可自动使用。

```
/dev/hda8 none swap sw 0 0
/swapfile none swap sw 0 0
```

启动手稿运行命令 swapon -a ，它将启动/etc/fstab 中所列的所有对换空间。因此 swapon 命令只有在启动额外的对换空间时才使用。

可以用free 监视对换空间的使用，它将给出所有使用的对换空间。

```
$ free
total used free shared buffers
Mem: 15152 14896 256 12404 2528
-/+ buffers: 12368 2784
Swap: 32452 6684 25768
$
```

前一行输出 (Mem:) 显示物理内存。Total 列不显示核心使用的物理内存 (通常大约 1MB)。Used 列显示被使用的内存总额 (第二行不计缓冲)。Free 列显示全部没使用的内存。Shared 列显示多个进程共享的内存总额。Buffers 列显示磁盘缓存的当前大小。

后一行 (Swap:) 对换空间, 显示的信息类似上面。如果这行为全 0, 那么没使用对换空间。通过 top, 或使用 proc 文件系统的 /proc/meminfo 文件可以得到相同的信息。得到某个对换空间的使用信息目前还比较困难。

可用 swapoff 取消对换空间, 一般不必这样, 除非是临时对换空间。对换空间中的要用的页被换入 (swap->RAM), 如果没有足够的物理内存, 就被换出 (RAM->swap, 到其他对换空间)。如果没有足够的虚拟内存放进所有页面, Linux 将开始震荡 (thrash), 很长时间以后应该能恢复, 但此时系统不可用。取消一个对换空间前, 应该检查 (例如用 free) 是否有足够的物理内存。

用 swapon -a 自动使用的所有对换空间可以用 swapoff -a 取消, 它查看文件 /etc/fstab 得知要取消什么。任何手工启用的对换空间将被依然使用。

即使有许多空闲的物理内存, 有时许多对换空间也被使用着。这种情况是由于在某个时间需要对换, 但后来一个占用大量物理内存的大进程终止并释放了内存。直到被换出的数据要被使用之前它们并不自动换入。不必顾虑这种情况, 但知道为什么会发生这种情况会更安心。

#### 4.9.4 与其他操作系统共享对换空间

许多操作系统内置虚拟内存。由于它们只需在运行时使用, 即不会同时, 那么除了当前运行的, 其他所有对换空间都浪费着。如果他们共享同一个对换空间将更有效。这是可能的, 但需要一些 Hacking 工作。Tips-HOWTO 包含了一些如何完成这项任务的忠告。

#### 4.9.5 分配对换空间

也许有人告诉你, 应该分配二倍于物理内存的对换空间, 但这是个虚假的规律。下面说明如何才是正确的。

估计你的全部内存需求。这是你可能需要的最大量, 即你要同时运行的所有程序所需的内存要求的总和。可以同时运行你可能同时运行的所有程序试试。

例如, 如果想运行 X, 得分配 8MB 给他, gcc 要求数 MB (有些文件偶尔可能需要很大量, 数十 MB, 但一般 4MB 差不多), 等等。核心自己使用 1MB, Shell 和一些小工具可能需要几百 KB (或说, 总共 1MB)。不必太精确, 粗略估计就行, 但可以较悲观地考虑。

记得如果将有多人同时使用系统, 他们将都消耗内存。如果两个人同时运行相同的程序, 总内存消耗一般并非加倍, 因为代码页和共享库是单一的。

free 和 ps 命令对估计内存需求很有用。第一步的估计加上一些安全量。因为对程序大小的估计很可能是错误的, 因为你可能忘了一些要运行的程序, 并确定你有一些额外空间。应该有数 MB。(分配太多对换空间比分配太少好, 但不必过分, 因为不使用的对换空间对你来说是浪费)。

基于以上计算, 知道了总共需要多少内存。减去你的实际物理内存, 就是对换空间。(有些版本的 UNIX 中, 你还需要分配物理内存的映像空间, 所以第二步中计算的所需的就不

能减)。

如果你计算的对换空间比实际物理内存大得多(大好几倍以上),那么你也许需要更多的物理内存,否则系统性能将太低。即使计算显示你无须对换空间,最好还是至少有一些。Linux有些侵略性地使用对换空间,这样保持一定的空闲物理内存。即使内存还不为什么程序所需, Linux也会换出一些不用的内存页,这样在需要的时候就可以避免因对换的等待,即对换可以在硬盘空闲的时候提早完成。

对换空间可以分在几个硬盘中,这有时可以提高性能,依赖于这些盘的相对速度和存取模式。可以尝试几种方案,但要知道,正确的尝试是很困难的。不要相信某种方案比其他方案好的断定。

#### 4.9.6 高速缓存

与存取(真正的)内存相比,从磁盘读是很慢的。另外,在相对短的一段时间里,多次读硬盘相同的部分是很常见的。例如,可能先读了一封电子邮件,然后回复时又将它读入编辑器,然后复制它到一个文件夹时又用邮件程序读它。或者,考虑命令 `ls` 可能被系统上的很多用户频繁地使用。只从磁盘读一次信息,并保持在硬盘中,知道不再需要,除了第一次读,其他都会较快。这就叫磁盘缓存 `disk buffering`,用于此目的的内存叫 `buffer cache`。

不幸的是,由于内存是有限且缺乏的资源, `buffer cache` 一般不会足够大(大到能够装下所有人可能用到的数据)。当 `cache` 满时,最长时间不用的数据将被丢弃,内存释放给最新的数据。

磁盘缓冲也用于写操作。要写的数据经常马上又被读(例如一个源代码文件保存到文件中后又被编译器读出),所以将要写的数据放在缓冲里是个好主意。另外,只将数据放入 `cache` 而不马上写到磁盘,写操作的程序执行速度更快。写操作然后可以在后台完成,而不降低其他程序的速度。

许多操作系统有 `buffer caches` (即使名称不同),但并非都根据上述原理。有些是透写 `write-through`: 数据马上写到磁盘(当然也同时写到 `cache`),不马上写的 `cache` 叫回写 `write-back`。回写比透写更有效,但也更容易出错,如果系统崩溃,或电源突然掉电,或软盘在 `cache` 回写前被取出,那么 `cache` 中改变的数据将丢失。这可能意味着文件系统 `is not in full working order`,可能由于未写数据包含了系统记录信息的重要的变化。

因此,千万不要不经过正常的关闭过程直接关闭电源,或没有 `unmount` 就取出软盘(如果是 `mount` 的),或什么程序还在用着软盘,或软盘灯还在闪。 `sync` 命令刷新缓冲,即强制将所有未写数据写回磁盘,如果要确保所有数据安全回写,可以用它。传统的 UNIX 系统中,有个 `update` 程序在后台运行,它每 30 秒运行一次 `sync`,所以通常无须使用 `sync`。Linux 有一个另外的守候程序 `bdflush`,它克服了 `sync` 有时因磁盘 I/O 负荷太重(因为频繁的操作)而导致有时系统突然呆住的问题。

Linux 下, `bdflush` 由 `update` 启动。一般无须考虑它,但如果 `bdflush` 偶尔因为什么原因死了,核心会给出警告,此时应该手工启动它 (`/sbin/update`)。

`cache` 并不真正缓冲文件,而是块,就是磁盘 I/O 的最小单元(Linux 下,一般是 1KB)。这样,所有的目录、超级块、其他文件系统记录数据和无文件系统磁盘都可以被缓冲。`cache` 的效果决定于其大小。太小的 `cache` 几乎无用;它只能 `cache` 很少的数据,而可能在被重用前就被清除

了。大小有赖于有多少数据被读写，相同的数据的存取频度。惟一的方法是实验。

如果cache是固定大小，那么不应该太大，否则，会由于空闲内存空间太小而使用 swap(也很慢)。为了最有效地使用真实内存，Linux自动使用所有空闲内存作为 buffer cache，当程序需要更多内存时，自动减少 cache。

Linux下，对cache使用无须做任何工作，它完全是自动的。除了要正常关闭系统和取出软盘，无须关心cache。

## 4.10 X Window

当第一次登录时使用的是字符界面，许多人更喜欢图形界面。对Linux系统，图形界面的选择就是X Window系统。

为了运行X，必须已经安装了必需的包。如果在安装 Red Hat Linux时选择了X Window System组件，那么一切都已经准备好了。

### 4.10.1 如果没有安装X

如果在安装Red Hat Linux时没有选择X Window System组件，那么你的Red Hat Linux系统不会有必需的软件。虽然可以手工安装所需的包，但你会发现重新安装更容易，特别是如果你是Linux新手的话。

#### 1. XFree86配置

有三种方法为机器配置XFree86:

- Xconfigurator
- xf86config
- 手工

Xconfigurator 和 xf86config 在功能上是一样的，并且同样工作得很好。如果想得到更多的文档，请参见<http://www.xfree86.org>。

Xconfigurator 是一个全屏菜单驱动的程序，用来引导你配置X服务器。xf86config是随XFree86发布的面向命令行的程序。它不如Xconfigurator方便，但更完整。如果这些工具都不能提供一个可以工作的XF86Config文件，你可能有不支持的卡或者你必须手工写配置文件。通常情况是前者，所有在试图自己写配置文件之前，检查一下你的卡是否被支持。如果你的卡不被XFree86支持，也许得考虑使用商品化的X服务器。如果你对显示卡是否被支持有问题，可以检查<http://www.xfree86.org>得到关于XFree86的信息。

#### 2. X服务器

假定你在安装时选择了合适的显示卡，你的X服务器应该已经正确地安装。以后运行Xconfigurator或xf86config，必须选择相同的显示卡，不然自动检测将失败。

如果你认为显示卡安装了错误的X服务器，必须在配置之前安装一个正确的。例如，如果CD装在/mnt/cdrom，而你需要安装S3服务器，输入以下命令：

```
cd /mnt/cdrom/RedHat/RPMS
rpm -ivh XFree86-S3-3.1.2-1.i386.rpm
ln -sf ../../usr/X11R6/bin/XF86_S3 /etc/X11/X
```

这将安装S3服务器并做出正确的符号连接。

## Xconfigurator

你必须首先选择显示卡来配置 X Window。滚动支持的卡的清单直至找到你的卡。表 4-3 也许能帮助你决定适合你硬件的显示服务器。如果你的卡没有列出，可能它不被 XFree86支持。这时可以选择清单中的最后一项 (Unlisted Card) 或一个商业的 X Window 服务器。

下一步是选择显示器。如果你的显示器没有列出，可以选择某个通用的显示器类型或者 Custom，然后输入你自己的参数。用户化的显示器配置仅推荐给那些熟悉 CRT 显示器内部工作的人。一般用户应该从清单中选择一个通用的类型。然后要告诉 Xconfigurator 你有多少显示内存。将光标移到适合的条目，然后按 Enter 或 F12 键继续。对于下一步，建议选择默认值 (No Clockchip Setting)，但有经验的用户可以指定某个适合 clockchip。

选择服务器，如果你不清楚你的芯片组，最好的方法是查看卡。表 4-3 列出哪个芯片组需要哪个服务器。选择最适合你的硬件的一个。

表4-3 XFree86 X 服务器

Server	Chipset
8514	IBM 8514/A Boards and true clones
AGX	All XGA graphics boards
I128	#9 Imagine 128 (including Series II) boards
Mach32	ATI boards using the Mach32 chipset
Mach64	ATI boards using the Mach64 chipset
Mach8	ATI boards using the Mach8 chipset
Mono	VGA boards in monochrome
P9000	Diamond Viper (but not the 9100) and Others
S3	#9 Boards, most Diamonds, some Orchids, Others
S3V	Boards using the S3 ViRGE (including DX, GX, VX) chipset
SVGA	Trident 8900 & 9400, Cirrus Logic, C & T, ET4000, S3 ViRGE, Others
VGA16	All VGA boards (16 color only)
W32	All ET4000/W32 cards, but not standard ET4000's

如果以后你想提供显示器的频率，可以手工编辑配置文件或者再次运行 Xconfigurator，然后从清单中选择更接近的显示器。

最后的配置步骤是选择你想包含在 XF86Config 文件中的显示模式。用光标键在色深 (8、16 和 14bit) 清单中移动光标。用 Spacebar 选择分辨率，并用 Tab 键在域之间移动。当选择完想用的显示模式，将光标移到 OK 按钮并按 Enter，或者用 F12 快捷键，一个信息屏将显示模式的当前信息。

#### 4.10.2 如果已经安装了X

如果你在安装 Red Hat Linux 时选择了 X Window System 组件，那么你已经都设置好了。现在要做的就是让 X 运行起来。有两种方法：

- 登录后手工启动 X。
- 系统启动后自动启动 X。

让我们先介绍手工启动。

##### 1. 手工启动 X

Red Hat Linux 在安装完后不会自动为你启动 X。因此，在第一次启动 Red Hat Linux 系统

时,你将看到字符界面的登录提示。

为了启动X,先要登录(用你的非root帐号),然后输入startx命令。屏幕应该变成空白,然后(经过短暂延迟)看到一个图形的桌面带有一个或多个窗口。根据所安装的包以及其他变量的不同,所看到的桌面会有所不同。

结束后若想离开X,可以用鼠标左键单击桌面的任何部分(换句话说就是屏幕上不属于任何窗口的部分)。选择Exit Fvwm, Quit或logout菜单条目,X将关闭,回到原先的字符界面。然后你可以正常退出。

## 2. 自动启动X

注意,在使X能自动启动之前,必须确定你的X配置能正常工作。不然,将会使你难以登录你的Red Hat Linux系统。如果还没有做到这一点,请回到上一节。

可以将你的Red Hat Linux配置成在启动系统时自动启动X。当这样配置时,xdm会运行,并显示一个图形界面的登录屏幕。登录之后,你将运行在一个正常的X进程中,就像你手工输入了startx命令一样。

步骤如下所示。

### (1) 用telinit测试xdm

telinit命令是用来改变Red Hat Linux系统的运行级别。正是运行级别控制着系统操作的不同方面,包括是否启动xdm。新安装的Red Hat Linux系统使用级别3为默认值,这导致你在登录时看到字符界面。因为xdm是在级别5启动,必须输入命令/sbin/telinit 5。

注意 必须以root登录来使用telinit。在修改运行级别时不能在Red Hat Linux系统中运行任何程序,因为修改运行级别时将杀死所有运行着的程序。

如果一切正常,短暂延迟后将看到xdm登录屏幕。登录并检查X的桌面。然后退出,确信xdm重新出现。如果是这样的话,你的系统自动启动X的配置是正确的。如果有问题,可以通过运行telinit(如, '/sbin/telinit 3')或者重新启动回到级别3。

### (2) 编辑/etc/inittab

文件/etc/inittab是用来决定系统的默认运行级别以及其他事情的。我们需要把默认运行级别从3改到5。因此,我们必须编辑/etc/inittab。用你选择的编辑器修改/etc/inittab中的这一行:

```
id:3:initdefault:
```

当你完成时,你应该看到:

```
id:5:initdefault:
```

注意 确信你只是将数字3改成了5! 不要修改其他任何东西,否则你的Red Hat Linux系统可能根本无法启动! 修改之后,退出编辑器,用这条命令检查你的修改:

```
less /etc/inittab
```

(按Space键来翻页,按Q退出) 如果一切正常,就可以重启动了。

### (3) 重新启动

## 3. 修改你的桌面

wmconfig使你方便地修改你的桌面显示。选择 Preferences菜单条,然后(在WM Style下)可以选择不同的桌面(也叫窗口管理器)形式。

#### 4. 基于X的快捷工具

对于Red Hat Linux的新用户，有一些工具可以使工作更简单。它们都需要 X来运行，所以必须先设置好X。这些工具是：

用户信息工具：方便地修改你的 `gecos`，或基本帐号信息。运行 `/usr/bin/userinfo`。

用户口令工具：用它可以简单地修改口令。运行 `/usr/bin/userpasswd`。

文件系统装载(`mount`)工具：使装载或卸载文件系统更简单。每个可装载的文件系统必须在`/etc/fstab`中有 `user`选项(参见 `mount` 在线帮助以得到关于 `user`选项的更多信息)。运行 `/usr/bin/usermount`。

网络设备工具：用这个工具启动和结束网络界面变成了点击的操纵。运行 `/usr/bin/usernet`。要求每个用户控制的界面是用户可以控制的。