

UNIX/Linux 文件系统的 shell 命令

3.1 文件系统的安装和卸载

在使用文件系统以前,先了解一下文件系统的安装和拆卸。

1. 文件系统的安装

除根文件系统外,UNIX/Linux 文件系统在使用之前,必须安装到系统中的某个目录下。

1) 图形用户界面下的安装

对于光盘、优盘、移动硬盘等一类的移动存储设备上的文件系统,在 GNOME 图形用户界面下,Linux 系统支持即插即用的功能。用户在使用的时候,只要将优盘(BOOTABLE)和光盘(Fedora 11 i386 DVD)连接到计算机上,系统就能自动识别,并将其相应的图标显示在桌面上。图 3-1 是 GNOME 示例。只要用鼠标双击图标,即可打开并浏览优盘或光盘上文件系统的内容。

2) 字符命令行模式下的安装

在字符命令行模式,用户如果需要使用移动存储设备上的文件系统,则需要显式地运行安装命令 `mount`,将文件系统安装到系统中的一个目录下。由于 `mount` 命令没有建立安装点的功能,因此应该确保执行 `mount` 命令时,安装点已经存在。安装命令的格式如下:

```
mount [-o<选项>] [-参数] [设备名称] [安装点]
```

即

```
mount -o options -t type device mount-point
```

其中常用的参数有:

(1) `[-参数]`(即`[-t type]`): 是要安装指定设备上的文件系统类型。常见的有:



图 3-1 GNOME 中的光盘和优盘

①minix 为 Linux 最早使用的文件系统；②ext2 为 Linux 目前常用的文件系统；③msdos 为 MS-DOS 的 fat,就是 fat16；④vfat 为 Windows 98 常用的 fat32；⑤nfs 为网络文件系统；⑥iso9660 为 CD-ROM 光盘标准文件系统；⑦ntfs 为 Windows NT/2000/XP 的文件系统；⑧hpfs 为 OS/2 文件系统。

(2) -o<选项>：指定安装文件系统时的选项，有些也可用在/etc/fstab 中。常用的有：codepage=XXX(代码页)，iocharset=XXX(字符集)，ro 以只读方式安装，rw 以读写方式安装，nouser 使一般用户无法安装，user 可以让一般用户安装设备。

(3) [设备名称](即 device)：是要安装的文件系统所在的设备(又叫做特别文件)名称。在 Linux 上，第一个软驱的设备名为 fd0，第二个软驱为 fd1；第一个硬盘为 hda，第二个硬盘为 hdb；第一个硬盘的第一个分区为 hda1，第二个分区为 hda2；以此类推。USB 盘为 sdb1。

(4) [安装点](即 mount-point)：是安装的目录名，mount-point 不一定是一个空目录，但一旦挂载了文件系统直到卸载这个文件系统后，原目录中的内容才可以被访问。

为了将软盘安装到 Linux 系统，可以使用如下命令，先制作 ext2 文件系统，再进行安装：

```
a. #mke2fs /dev/fd0 1440
b. #mount -t ext2 /dev/fd0 /mnt/floppy
```

安装 DOS 系统的 USB 存储设备或软盘：

```
#mount -t msdos /dev/sdb1 /mnt/usb
```

或

```
mount -t msdos /dev/fd0 /mnt
```

将 Windows 2000/XP 的 E 分区上的 fat32 文件系统安装到/mnt/nt 下，使用如下命令：

```
#mount -t vfat /dev/hda6 /mnt/nt
```

安装完成以后，整个文件系统的逻辑结构如图 3-2 所示。

如果要显示软盘上的 readme.txt 文件，只要输入“cat/mnt/floppy/readme.txt”即可。

3) 自动安装

虽然除根文件系统外，都应该显式地使用 mount 命令将需要访问的文件系统所在分区安装上。但如果每次开机都进行安装操作，显然太烦琐。为此，Linux 提供了自动安装需要挂载的 Linux 各个分区以及其他设备的方法。这是通过编辑在/etc 目录下的 fstab 文件，来实现所需文件系统的自动安装。在 fstab 文件里，给出了 Linux 开机时自动安装的文件系统的列表。

Linux 系统在启动时会自动地从/etc/fstab 文件读取信息，并将此文件中指定的文件系统安装到指定的目录。/etc/fstab 文件的格式如图 3-3 所示。

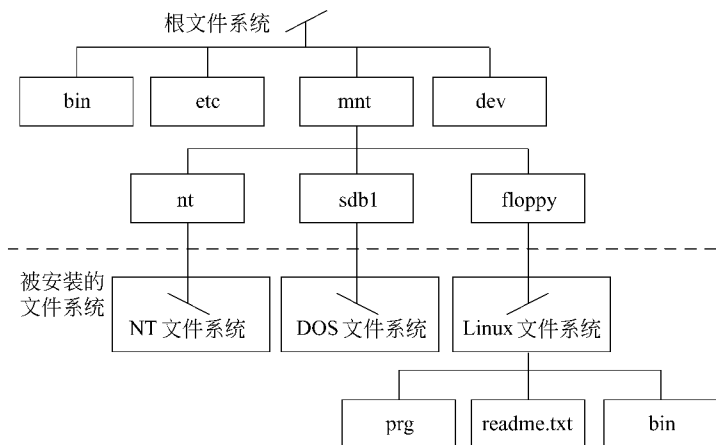


图 3-2 安装了其他文件系统后的系统结构

```

# /etc/fstab
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or vol_id(8) for more info
UUID=bdedb5a9-113b-4e7f-a2b3-7f59a8194230 /boot ext3 defaults 1 2
/dev/mapper/VolGroup-lv_root / ext4 defaults 1 1
/dev/mapper/VolGroup-lvswap swap swap defaults 0 0
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts defaults 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
  
```

图 3-3 fstab 文件

在 fstab 文件中，每一行都包括 6 个域，用 Tab 键或空格键分开。这 6 个域从左至右分别为 device、mount point、type、options、dump、pass。

(1) device：要安装的文件系统的设备名称或块信息，也可以是远程的文件系统。设备也可以使用卷标(label)或全局唯一标识符(UUID)来表示。图 3-3 例子里的第 1 行就是使用 UUID。

(2) mount point：安装点。对于交换分区(swap)，这个域填写 swap，表示没有安装点。

(3) type：指定文件系统的类型。

(4) options：安装时使用的选项。参数 defaults 实际上包含了一组默认参数，defaults 代表选项 rw、suid、dev、exec、auto、nouser 和 async。rw 以可读写模式安装；suid 开启用户 ID 和小组 ID 设置位；dev 可解读文件系统上的字符或区块设备；exec 可执行二进制文件；auto 自动安装；nouser 使一般用户无法安装；async 以异步方式执行文件系统的输入输出操作。

可以看到在这个列表里,光驱和软驱是不自动安装的,参数设置为 noauto。如果要设成自动安装,必须保证每次开机时,光驱和软驱里都要有盘。

(5) dump: 1 表示将整个文件系统的内容备份,0 表示不备份。一般选 0。

(6) pass: 指定如何使用 fsck 来检查硬盘。0 表示不检查,大于 0 的数字表示 fsck 检查硬盘时的次序。根目录必须首先检查,应该在这里填写 1。

如果用户希望系统自动加载 Windows 系统的磁盘分区 D 上的文件系统,可以遵循上面的格式,在 fstab 文件中添加如下一行:

```
/dev/hda5 /mnt/wind vfat defaults, codepage=936, iocharset=cp936 0 0
```

这里还加了 codepage=936 和 iocharset=cp936 参数,以支持中文文件名。

4) 查看安装的文件系统

若 mount 后无任何选项,则显示系统已安装的各种文件系统。

2. 卸载文件系统

命令的格式:

umount 特别文件名

功能: 将已安装的指定的文件系统拆卸下来。

注意: 在取出安装的介质(如软盘)之前,要先使用 umount 卸载掉软盘上的文件系统,否则会导致信息丢失。

例如,umount dev/fd0 命令执行完后,才可以取出相应的介质。为了检查一下该文件系统是否已正常卸载,可用不带参数的 mount 命令检查是否已经看不到了。

需要说明的是,mount 和 umount 命令只能由超级用户使用。

3.2 了解 Linux 文件系统的目录结构和文件属性

1. Linux 文件系统的目录结构

在 Linux 文件系统根目录下,有如下一些子目录: bin、etc、dev、home、install、lib、mnt、proc、root、tmp、user、var (不同版本,目录会略有不同)。

(1) /bin 目录中存放的是系统程序文件。

(2) /etc 目录中存放的是系统配置文件等。

(3) /dev 目录中存放的是设备驱动程序文件,用来访问系统设备和资源,如磁盘设备、调制解调器、内存等。/dev 目录下常用的设备文件有:

① /fd0、/fd1: /fd0 代表第一个软盘,/fd1 代表第二个软盘。

② /console: 指系统控制台,也就是直接连接系统的监视器。

③ /hda1: 代表第一个硬盘。

④ /tty: /tty 开头的设备名代表用户系统上的虚拟控制台。

⑤ /tty1: 代表第一个虚拟控制台。

- ⑥ /tty2: 代表第二个虚拟控制台。
- ⑦ /sda、/sdb、/sdc: 代表 usb 接口的存储设备。

2. Linux 系统的文件名和文件的属性

1) 文件名

UNIX/Linux 文件系统采用带链接的多级树型的目录结构。为此,文件用路径名表示。与 PC-DOS 不同,UNIX/Linux 用正斜杠“/”分割目录的层次结构。如一个目录路径“/usr/bin/”。Linux 文件名没有标准格式。除“/”字符外,通常文件名可以包含任何字符,但“~”字符不能作为文件名的第一个字符。文件名的长度限制在 256 个字符以内。Linux 的文件名也提供了功能强大且精确的通配符。

2) 文件的访问权限

文件的访问权限是 Linux 系统安全的核心部分,可以执行的权限有:

- (1) r(read,可读)权限: 表示可以读取文件内的数据。
- (2) w(write,可写)权限: 表示可以改变或删除该文件。
- (3) x(execute,可执行)权限: 表示可以把该文件当成程序执行。

3) 文件的类型和属性

UNIX 和 Linux 系统把文件分成三种类型: 普通文件、目录文件和特殊文件。假定用“ls -l”给出文件的属性如图 3-4 的显示。其中,显示的左边的第一列就表示文件的可能类型,以后依次为文件的主人(owner,所有者)、小组用户(group)和其他用户允许的存取权限,文件的链接数,文件的主人名称(所有者名称),小组用户名称,文件或目录的大小(对于特殊文件,则为主设备号和次设备号),创建或修改的日期和时间以及文件或目录名。

普通文件:	-	rw-	r--	1	bin	bin	2208	May 28 1987	/etc/rc
								└─以字节为单位的文件或目录的大小	
目录文件:	d	rw-	r-x	2	bin	bin	240	Nov 13 1987	dos
特殊文件:	b	rw-	rw-	3	bin	bin	2, 7	Jan 12 15:07	/dev/fd0
	c	-w-	-w-	2	bin	bin	6, 0	Jan 11 13:37	/dev/Lp
	↑						↑ ↑	日期和时间	文件名
文件类型	所有者存取权限	同组用户存取权限	其他用户存取权限	链接数	所有者名称	小组用户名称	主设备号	次设备号	

图 3-4 执行 ls -l 的显示

对于普通文件和目录文件,类型分别为“-”和“d”。

对于特殊文件,其中可能有的类型中,b 代表块设备,c 代表字符设备,p 代表有名管道文件,s 代表网络套接字文件,l 代表符号链接文件。若是一个符号链接文件,对于文件名部分,一般显示为“符号链接文件名->目标文件的路径名”。

3.3 shell 中的特殊字符

(1) 星号(*): 可以匹配任何字符串, 包括空串。

(2) 问号(?): 可以匹配任何单个字符。

(3) 方括号([]): 可以匹配括号中括起来的所有字符中的任何一个单个字符, 可以在两个字符间用一个短划线“-”表示字符范围。例如:

```
$ls file[12349]或 ls file[1-49]
```

这两条命令都列出当前目录下的以 file1、file2、file3、file4 和 file9 为文件名的 5 个文件的有关信息。

(4) 分号(;): 当一个命令行含有多个命令时, 各命令之间用“;”分隔。

例如:

```
$Who ;Ls
```

(5) 反斜杠(\): 为了将 shell 的特殊字符“* ? [] & ;”变为一个普通字符, 则必须在这些字符前加一个反斜杠“\”。

另外, 当反斜杠放在行尾时, 它的作用是续行符。使该行继续到下一行。

(6) #: 是 shell 中的注释符, 与 C 语言中的“/*”相同。以 # 号开始, 以回车符结尾的行都作为注释语句。

例如:

```
$echo hello #hello
Hello
```

显然, #hello 被看成注释语句了。

(7) 双引号("): 使其所包含的字符中除“\$”、“\”以及“,”以外的其他字符都失去其特定含义。

例如:

```
$export var=date
$echo "$var"
date
```

例如:

```
$echo "TYPE a\ $if you are happy."
Type a $if you are happy.
```

(8) 单引号('): 使其所包含的字符中所有特殊字符失去特定含义。

例如:

```
$echo 'TYPE a$ * if you happy.'
TYPE a$ * if you are happy.    #这时的$已成为普通字符被显示
```

例如：

```
$ export var=date
$ echo '$var'      # 显示的结果如下：
$ var
```

(9) 倒单引号(''),是与~在同一个键上的符号): shell 命令解释程序把倒单引号中的内容作为命令运行,并把程序的输出作为自变量。

例如：

```
$ echo date
date
```

又例如：

```
$ echo 'date'
Tue NOV 7 21:08:12 1991
```

从上述的两条命令的输出结果可以清楚地看出倒单引号的作用。

3.4 目录操作

3.4.1 创建目录和删除目录

(1) 与 PC-DOS 中的 md 的功能类似,UNIX/Linux 系统中的 mkdir 命令可以创建一个或多个目录。

命令格式：

```
mkdir [选项] 目录 1 目录 2 ...
```

常用的选项有：

-m: 指定目录的访问模式,即权限,用 3 位八进制数字紧跟在后面,以给出创建者、小组用户和其他用户对创建的目录的访问方式。

例如：

```
$ mkdir bin lib temp
```

在当前目录下创建了 3 个子目录,名字分别为 bin、lib 和 temp。

又例如：

```
$ mkdir test -m 775
```

在当前目录下创建名为 test 的目录,它的访问权限为 775,即该目录的所有者和小组有权读、写和执行这个目录,其他用户可以读和执行。

(2) UNIX/Linux 的 rmdir 命令一次可以删除多个目录,同样,删除目录时,要求目录为空。其格式为：

```
rmdir [选项] 目录 1 目录 2 ...
```

另外, Linux 系统还可以使用删除文件命令 `rm` 用来强行删除不空的目录。

3.4.2 列出某个目录下的内容

这个命令与 PC-DOS 中的 `dir` 类似, 列出指定目录下的各文件的属性。
格式:

```
$ls [选项]
```

选项的作用:

- l: 按长格式列出每个文件的全部属性信息(如文件名、文件大小、文件的存取方式等)。
- a: 显示指定目录下的所有文件, 包括以“.”和“..”在内的隐含文件。
- s: 对每个文件按块给出其大小。
- i: 在第一列给出每个文件的 i 节点号。
- t: 按最近修改时间顺序(为倒排序), 列目录。
- R: 递归列出指定目录下的子目录和文件的名字。

3.4.3 改变工作目录和确定当前所在的目录位置

(1) 改变当前目录。需要注意的是, 当转到别的用户目录时, 能执行的操作取决于转之前那个用户授予的权力。

命令格式:

```
cd [目录路径名]
```

当不给出路径名时, `cd` 命令的执行就转到注册目录(HOME)中, 即转到由环境变量 `$HOME` 指定的目录, 否则, 转到按路径名指定的目录。

若用户正工作在 `/home/liming/txt` 目录下, 输入如下命令:

```
$cd ✓ 或 cd ~    #都转到注册目录(HOME)
```

(2) 使用 `pwd`, 显示当前所处的目录位置。

若当前目录为 `/usr/xiaoming`, 则执行 `pwd` 命令后, 就显示如下信息:

```
$pwd ✓  
/usr/xiaoming
```

3.4.4 显示特定目录的大小

命令格式:

```
du [选项] [目录名]
```

`du` 命令的作用是显示指定目录占用磁盘空间的情况。`du` 命令不跟参数时, 显示当前目录占用磁盘的情况, 第一列显示以 KB 为单位的空间大小, 第二列显示的是子目录的名字。

du 常用的选项有：

-a：显示所有文件而不仅是目录占用的空间。

-h：以可读形式列出各文件大小。

-s：只显示该目录的大小，目录中的内容占用的空间不显示。

例如：“du -h ../”命令将显示上一级目录下各级目录占用空间的情况。

3.4.5 复制目录和文件

UNIX 系统使用 COPY 命令实现复制目录，使用 cp 命令实现复制文件。Linux 下的 cp 既可复制目录，也可复制文件。命令格式相同。

格式：

```
$ cp [选项] 源文件或目录... 目标文件或目录
```

功能：将一个或多个源目录（当源目录为多个时，各目录之间用空格分隔）中的文件复制到目标目录下。或将一个（或多个）文件复制成一个（或多个）目标文件。

选项的作用：

-f：如果目标文件存在，则覆盖目标文件且不提示。

-i：当目标已存在时，在覆盖之前，先提示是否覆盖。

-u：只复制目标中不存在的文件或者修改过的文件。

-r 和 -R：递归的复制源目录及子目录下的文件。

-p：复制时保持文件的属性不变。

-d：复制时保持文件的链接关系。

-a：等同于 -dpR，即复制源目录下的所有目录和文件。

例如：

```
cp /usr/doc/* ./
```

把 /usr/doc/ 目录下的所有文件复制到当前目录下。

```
cp -R /tmp/test/index /home/user/
```

把 /tmp/test/ 目录及其子目录下所有名为 index 的文件都复制到 /home/user/ 下，复制时保持原来的目录结构。

3.4.6 目录和文件的改名和移动

这个命令与 PC-DOS 中的 rename 类似，既可以给文件或目录换名，也可以移动文件或目录。实现移动功能时，要求使用该命令的用户对目标有写权。格式如下：

(1) mv 源名 目标名

(2) mv 文件 1 文件 2 ... 目录

第(1)个命令完成的功能：

- 如果目标名不存在,将源名改为目标的名字。
- 如果目标名已存在,则将源移到目标名指定的目录下。

第(2)个命令完成的功能是将一个或多个文件移到指定的目录中。

3.4.7 改变文件或目录的存取权

在一个文件上可以实行读、写和执行三种操作。操作文件的用户又分成三类:文件所有者(即文件创建者)、同组用户和其他用户。这三类用户对文件可能执行的任何一种操作都可以被允许或者被禁止。文件创建时,由文件所有者规定各类用户的存取权限。当希望修改文件的操作权限时,由文件所有者或超级用户使用如图 3-5 所示的命令进行修改。

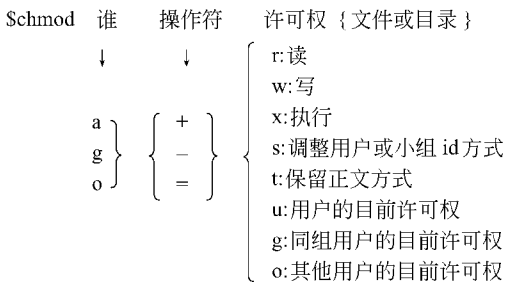


图 3-5 chmod 命令的格式

命令中的“谁”可以为 a、g、o、u。其中 a 代表 all,即所有的用户;g 代表 group,即小组用户;o 代表 others,即一般用户;u 代表 user,即文件的所有者。

“+”代表增加许可权;“-”代表删除许可权;“=”代表赋予指定的许可权。

“{文件或目录}”代表一个或多个文件或目录。

例如:

```
$ chmod u+w display
```

表示 u 代表的用户增加对文件 display 的写权。

```
$ chmod u+w,g=x display
```

表示 u 代表的用户增加对 display 文件的写权,g 代表的组用户增加执行权。

```
$ chmod go -r mytext
```

删除 g 代表的组和 o 代表的其他用户对文件 mytext 的读权。

```
$ chmod a-w+x hdish fdisk
```

表示所有用户对文件 hdish 和 fdisk 删除写权,增加执行权。

改变对文件和目录的操作许可权也可用数字方式实现,表 3-1 给出了各用户用数字表示的存取权限。