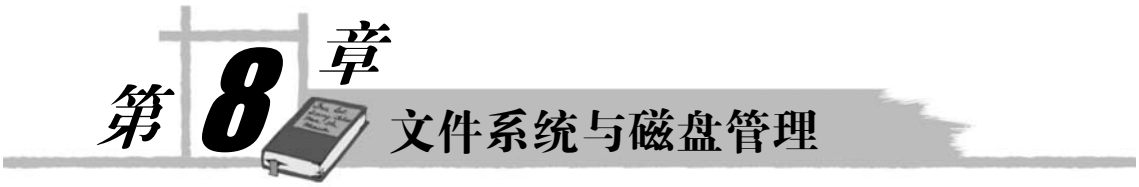

第8章 文件系统与磁盘管理

本章要点：

- ext2 和 ext3 文件系统。
- /proc、sysfs、swap 文件系统。
- 加载、挂载与卸载。
- 交换空间的使用。
- 硬盘结构和分区的使用。
- 磁盘配额的用法。
- 权限设置的方法。

8.1 Linux 文件系统概述

操作系统必定涉及到文件的存储和管理，这个功能就是操作系统的文件管理系统，简称文件系统。有文件的存储、读写等操作的存在，就一定涉及到另一功能即磁盘管理。文件管理需要三方面的支持：文件管理软件、被管理文件和管理信息数据库。文件管理实际上就是系统对存储设备关于文件管理需求的空间分配和组织工作。Linux 的文件管理和磁盘管理方式有其独到之处，熟悉文件管理和磁盘管理也是用好 Linux 的最基本的能力要求。

8.1.1 Linux 的文件系统类型

1. ext2 和 ext3 文件系统

所谓的 ext 系统是 Extended File System 的缩写，它包括 ext1、ext2 和 ext3 三种。ext 系列是专为 Linux 设计的，具有鲜明的 UNIX 特征。它采用了三级索引结构和目录树结构。此外一个很特别的地方是 ext 系统将各种设备都当成文件来管理。ext2 系列文件管理系统功能强大、易学易用，目前占据 Linux 系统中的大部分市场，也是大多数 Linux 系统默认的文件管理系统。

ext2 文件系统用的索引节点功能类似 Windows 的文件分配表（FAT），用来记录文件信息。可以将索引节点看成一个数据结构，其中包含了文件长度、创建时间、修改时间、权限、所属关系、存放路径等文件管理中所需的相关信息。一个索引节点就是一个数组，唯一对应一个文件或目录。为了便于管理，系统自动给每个索引节点分配一个索引节点号，用来唯一标识这个节点。每个索引节点号和其对应文件的文件名被同时保存在目录中。目录就是将文件名及其对应的索引节点号相关联的表。目录中的每一对关联被称为一个“连接”。需要注意的是，索引节点号和文件名是 1:n 的关系，一个文件的索引节点号是唯一的，但一个索引节点可以对应多

个文件名。这么设计的原因是为实现通过不同的路径去访问磁盘上的同一个文件。

默认情况下 Linux 使用的文件系统是 ext2，其特点是运行稳定、处理高效。但当 Linux 系统被更多地应用到很多极其重要的行业和环境中后，ext2 的缺点也就暴露无遗，ext2 是非日志文件系统（NLFS）。普通的应用不存在太大的问题，但对于那些极其重要的行业就是一个致命的弱点。这也是为什么要发展 ext3 系统的原因。

ext3 从 ext2 文件系统发展而来，是 ext2 的加强版，对日志功能加以强化。当前的 ext3 系统经过长时间的应用和改进已经非常成熟稳定可靠，并且完全兼容 ext2 系统。ext2 用户可以无障碍地使用 ext3 系统，无需学习和培训过程。

ext3 系统具有以下特点：

- 高稳定性。ext3 系统稳定性极好，容错能力非常强。在 ext3 系统被非法关机（如突然断电等情况）后，系统不需要检查文件系统。而 ext3 系统恢复的时间只需大概十几秒。
- 数据完整性保护功能。ext3 系统极大地提高了文件系统的完整性，能避免非法关机对文件系统造成的破坏。在数据完整性保护功能上，ext3 提供了两种模式供用户选择。最常用也最受欢迎的就是“同时保持文件系统及数据的一致性”模式。这种模式下，将永远杜绝由于非法关机而滞留磁盘上的临时文件、文件碎片等垃圾。
- 较高的运行速度。ext3 在速度这一性能上对比 JFS2 等系统并不占优势，而且 ext3 在存储数据时有可能还需要进行多次写数据操作。但其综合性能无疑是最好的。尤其是 ext3 支持从 ext2 系统升级，而无需备份和恢复数据。另外 ext3 相比 ReiserFS 和 XFS 等高速系统，拥有更低的 CPU 使用率。这一切都是因为 ext3 的日志功能对磁盘的驱动器读写头进行了优化，设计师在设计逻辑算法时就已经兼顾了各方面的考量。
- 支持数据转换功能。从 ext2 转成 ext3 非常方便，理论上只需键入简单命令即可完成整个转换工作，且不用提前做备份、恢复、格式化分区等工作。这很适合一贯使用 ext 系列系统的用户，也兼顾了新用户的体验。ext3 还提供了一个小工具 tune2fs，可用于将 ext2 很方便地转换为 ext3。反过来 ext3 也可以不经任何更改，而直接加载成为 ext2 文件系统，做到了良好的向下兼容。
- 支持多种日志模式。ext3 支持多种日志模式，一种日志模式下（data=journal 模式）ext3 将对所有的文件数据及 metadata（定义文件系统中数据的数据，即数据的数据）进行日志记录；另一种日志模式（data=ordered 或者 data=writeback 模式）下则只对 metadata 记录日志，而不对数据进行日志记录。系统管理员可以根据系统的实际工作需要，在系统的工作速度与文件数据的一致性之间做出选择。

将 ext2 文件系统转换为 ext3 文件系统的操作很简单，只需要输入以下命令：

```
tune2fs -j /dev/hda9
```

转换结果如图 8.1 所示。

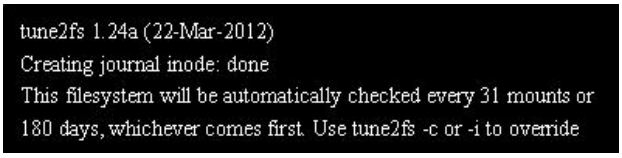

图 8.1　ext2 成功转换为 ext3

注意：将 ext2 文件系统转换为 ext3 文件系统时，不用先卸载分区再转换，直接输入 tune2fs 命令即可。但要记住在完成转换工作后，一定要将/etc/fstab 中所对应分区的文件系统由原来的 ext2 更改为 ext3。

ext3 日志文件可以存放在另一个非默认的存储设备上，如/dev/hda8 中。如果希望在/dev/hda8 上创建一个 ext3 文件系统，并将日志存放在另一个外部设备/dev/hda2 上，可以通过输入以下命令实现：

```
mke2fs -J device=/dev/hda8 /dev/hda2
```

新的 e2fsprogs 中的 e2fsck 支持 ext3 文件系统。当一个 ext3 文件系统被破坏时，先卸载该设备，再用 e2fsck 修复。方法很简单，只需要输入以下命令即可：

```
umount /dev/hda8
e2fsck -fy /dev/hda8
```

综上所述，tune2fs 是目前 Linux 系统由 ext2 文件系统转换为 ext3 文件系统的最简单且兼容性最好的途径，其实现手段技术门槛最低。又因为 ext3 是直接从 ext2 发展而来，文件系统过渡升级过程最为平滑，这么做可以最大限度地保证系统数据的安全性。所以当前 Linux 系统要使用日志文件系统，最保险的方式就是选择升级到 ext3。

2. /proc 文件系统

/proc 文件系统是一个伪文件系统，它以文件系统的方式为用户访问系统内核数据提供接口。所谓“伪文件系统”是指/proc 是虚拟的，存在于内存中而不是硬盘上。/proc 文件系统是一种内核和内核模块向进程（process）发送信息的机制，这也是/proc 的得名来历。这个伪文件系统使用户能与存在于内核内部的重要数据结构进行交互，从而获取有关进程的有用信息，并在运行中通过改变内核参数来改变系统设置。用户和应用程序都可以通过/proc 文件系统得到系统信息，并能通过/proc 改变内核的某些信息。需要注意的是，正由于这是一个虚拟的系统，因此里面的文件都不是真实存在的，/proc 的内容都是动态创建的，它只存在于内存当中。

开发/proc 的初衷是为了提供一个让用户获取系统进程信息的方便渠道，结果在使用过程中发现它非常有用，超出了预期的要求。于是干脆让内核中的很多元素也用它来报告信息，或启用动态运行时配置。

/proc 是由内核控制的，并没有一个物理上承载/proc 的设备。这是因为/proc 主要用于存放由内核控制的系统状态信息，而大部分这些信息的逻辑位置在由内核控制的内存中。因此如果对/proc 做如下操作：

```
ls -l
```

会很惊奇地看到一个现象，大部分文件的长度都是 0，但如果同时查看这些文件的内容，里面却又实实在在的有很多信息。这是因为/proc 和其他常规的文件系统一样都将自己注册到虚拟文件系统（VFS）中。但它毕竟只是一个虚拟的文件系统，因此在 VFS 要调用它请求文件、目录的 i-node 时，/proc 才会根据内核中的信息在内存中建立相应的文件和目录。这也是为什么那些文件平时看总是为 0 的原因。

用户可以通过输入以下命令来加载/proc 文件系统：

```
mount -t proc proc /proc
```

没有问题的情况下就会成功加载/proc 文件系统到用户机器上。

/proc 文件可用于获取有关内核状态、计算机属性、正在运行的进程状态等相关信息。而且大部分/proc 中的文件和目录还提供系统物理环境最新的信息。如上所述，/proc 中的文件

是虚拟的，并不真正存在于硬盘上，但用户还是可以像使用任何正常文件一样，用文件编辑器或类似 more、less、cat 等命令查看它们。如有编辑程序试图打开一个虚拟文件，/proc 会先获取内核中的相应信息，然后“凭空”创建这个文件。

表 8.1 说明了一些重要的读取系统信息的/proc 文件（都是只读的）及其具体作用。

表 8.1　读取系统信息的/proc 文件及其具体作用

文件名	作用说明
Cmdline	内核启动参数
Cpuinfo	CPU 信息
Iomem	IO 设备的内存使用信息
Interrupts	显示被占用的中断信息和占用者的信息
Ioports	I/O 端口的使用
Kcore	系统物理内存映像，不可读取
Loadavg	系统平均负载
Meminfo	物理内存和交换分区使用信息
Modules	加载模块列表
Mounts	挂载的文件系统
Partitions	系统识别的分区表
Swaps	交换空间的利用情况
Version	内核版本信息
Uptime	系统运行时间

在/proc 目录下所有以数字命名的子目录都用于存储进程相关信息，而每个子目录的数字目录名就是其对应的进程 PID，这些文件也是只读属性。此外还有一个特殊的/proc/self 目录，它指向当前正在执行进程的符号连接。例如当用户执行 cat/proc/self/cmdline 时，会返回命令本身。进程相关信息如表 8.2 所示。

表 8.2　进程相关信息文件描述

文件	描述
Cmdline	进程启动参数
Environ	环境变量
Limits	进程限制信息
Status	进程状态
Maps	进程使用的动态链接库文件

表 8.1、表 8.2 只是列出了最常见、最有用的/proc 文件，实际上的/proc 文件要远远多于这个数量。如果读者想进一步了解自己感兴趣的文件，可以对相应/proc 文件使用 more 命令，也可在线获取相关信息。之所以使用 more 命令而非 cat 命令，是因为无法预知生成的文件大小，

而有些文件如 kcore 就有可能非常长。这样的话用 cat 显然是不合适的。

3. sysfs 文件系统

sysfs 文件系统类似/proc，也是个特殊的文件系统。它的作用在于将系统中的设备组织成层次结构，然后向用户或有需要的程序提供内核的数据结构信息。这个文件系统放在/sys 目录下。表 8.3 是其子目录及其所含内容说明。

表 8.3　sysfs 文件系统的子目录及作用

子目录	作用
Block	所有的块设备
Bus	系统中所有的总线类型
Class	系统中的设备类型，如声卡、网卡等
Devices	系统中所有的设备，且以设备挂载的总线类型组织成层次结构

4. tmpfs 文件系统（虚拟内存文件系统）

Linux 的虚拟内存和 Windows 的组成方式不同，Linux 的虚拟内存由两部分组成：物理内存和交换分区。tmpfs 的最大容量就是这两部分的和。tmpfs 很灵活，既可以使用物理内存，也可以使用交换分区。而且它的大小可根据需求动态分配，因此不会造成资源浪费。此外它的读写速度很快，一般服务器都用它来提升性能。

5. swap 文件系统

和 tmpfs 不同，swap 只用于交换分区。Linux 有一个特点，即交换分区的大小是系统物理内存的两倍，而交换分区则是 Linux 运行的必须分区。通常把交换分区叫作 swap 分区是因为它只作用于交换分区，且交换分区只能使用 swap 文件系统。swap 不由用户控制，而是 Linux 管理的对象。

6. 其他所支持的文件系统

虚拟文件系统是一个很有用的概念，可以实现对多种文件系统的支持。此外，虚拟文件系统还允许用户在同一块磁盘上划分不同的文件系统。这样既提高了 Linux 的灵活性，又实现了不同操作系统、环境间的资源共享。除了上面所述的五种 Linux 文件系统外，Linux 还支持很多文件系统，如 FAT16、FAT32、sysV、nfs、iso9660 等。有一点要说明的是，赫赫有名的 NTFS 并不被 RHEL 5 Sever 直接支持，需要安装相应的 RMP 包并挂载才可以使用。

8.1.2　创建 Linux 文件系统

当需要加载一个分区（即一个文件系统）时，首先要做的就是确认这个分区的文件系统类型（即分区格式化），然后才能挂载使用。挂载的方法有很多，例如 mount，或者修改 /etc/fstab 实现开机自动加载。当新增一个新的分区，或是加一块新的硬盘时，就需要用分区工具来添加分区，然后才能创建分区的文件系统，最后才是挂载文件系统。

1. 分区

分区是对存储设备而言的，最常见的就是本地硬盘或移动硬盘（当然也包括 U 盘等移动存储设备）。分区是为了将一块较大的存储空间分成若干个较小的区域以满足使用需求。在 Linux 中进行硬盘分区操作的工具有 fdisk、parted、cfdisk 等，虽然后两者在某些方面有优点

（如 parted 中的数据备份功能），但就目前应用情况来看最好用的还是 fdisk。

2. 格式化

格式化能够让分区成为某种具体的文件系统，如上面讲到过的 sysfs 等。常用的格式化工具有 mkfs、mkfs.ext3、mkfs.reiserfs、mkfs.ext2、mkfs.msdos、mkfs.vfat、mkswap 等。需要注意的是，有的机器会在格式化成 reiserfs 时提示不能创建 reiserfs 文件系统，这是因为该机器上没有 mkfs.reiserfs 或者 mkreiserfs，从网上下载一个即可解决这种问题。

- mkfs 的使用方法

命令格式：mkfs -t filesystem device

命令说明：这里的 filesystem（文件系统）必须要指定，如 ext3、reiserfs、ext2、fat32、msdos 等。而 device（设备）就是存储设备，如一个硬盘的分区、软盘、光驱等。需要注意的是，在格式化分区之前一定要查看硬盘分区情况，从而保证是有针对性的格式化。使用 fdisk-l 来查看分区情况。这个信息很重要，例如有一个用户想格式化一个 U 盘 sda 中的一个分区，则应该输入：

```
fdisk -l /dev/sda
```

但如果希望将 sda 格式化成 ext3，则应该使用如下命令：

```
mkfs -t ext3 /dev/sda
```

成功之后就可以直接挂载这个分区了。

- 其他几种用法

mkfs.ext3 /dev/sda6	将分区格式化成 ext3 文件系统
mke2fs -j /dev/sda6	将分区格式化成 ext3 文件系统
mkfs.ext2 /dev/sda6	将分区格式化成 ext2 文件系统
mke2fs /dev/sda6	将分区格式化成 ext2 文件系统
mkfs.reiserfs /dev/sda6	将分区格式化成 reiserfs 文件系统
mkfs.vfat /dev/sda6	将分区格式化成 fat32 文件系统
mkfs.msdos /dev/sda6	将分区格式化成 fat16 文件系统，msdos 文件系统就是 fat16
mkdosfs /dev/sda6	将分区格式化成 fat16 文件系统，同 mkfs.msdos

- mkswap 的用法

mkswap /dev/sda6	创建此分区为 swap 交换分区
swapon /dev/sda6	加载交换分区
swapoff /dev/sda6	关闭交换分区

- 查看 swap 分区的方式

swapon /dev/sda6	加载交换分区
swapon -s	查看 swap 分区

8.1.3 挂载和卸载文件系统

和 Windows 不同，Linux 中所有存储设备都要经过挂载操作才能使用。尤其是常用的硬盘、U 盘、光驱等设备，初学者很容易忘记挂载，或是使用完毕之后的卸载操作，这一点值得重视。

所谓挂载，就是将存储介质中的内容做一个映射，放到一个指定的目录中去。这个目录就

成了所谓的挂载点。这样一来，对存储介质的访问就转化成了对挂载点的访问。又因为这个映射是一对一关系，因此一个挂载点同时只支持一个存储设备。

/etc/fstab 下的文件是各磁盘分区的默认设置。如果不指明挂载点就由它来指定挂载目录，且实现关机时自动卸载操作。此外，移动存储设备如光驱、U 盘、移动硬盘等既可以设置为启动自动挂载，也可以在其他时候根据需要手动挂载或卸载。注意，当移动介质使用完毕之后，一定要卸载才能取出，否则会出现错误。

1. 挂载命令 mount

命令格式：mount [-t vfstype] [-o options] device dir

命令说明：mount 命令可以将指定的设备挂载到已存在的目录，当文件系统挂载完成后用户可通过对该目录操作来实现对指定设备的文件读写的操作。

参数说明：

- -a：挂载 fstab 中包含的所有文件系统。
- -f：模拟整个挂载的过程，并不实际挂载设备。
- -F：和-a 参数一起使用以并行顺序挂载每个设备。
- -l：显示所有已挂载的系统，包括卷标签。
- -L<标签>：只挂载符合指定标签的分区。
- -n：在挂载过程中不向/etc/mtab 中写入资料。
- -o：设置文件系统挂载时的操作参数。
- -r：以只读模式挂载设备。
- -t：指定要挂载的文件类型。
- -w：以读写模式挂载设备。
- --bind：重新挂载一个设备到新目录在两个挂载点下设备都可以用。
- --move：把挂载设备挂载到新的目录下。

注意：mount 命令不能建立挂载点，所以如果该挂载点不存在则应该先建立该挂载点，完成挂载后就可以用挂载点目录来实现对光驱的读写操作。

2. 卸载命令 umount

命令格式：umount [-ahnrvV][-t <filesystem type>][filesystem]

命令说明：umount 命令可以卸载文件系统。

参数说明：

- -a：卸载/etc/mtab 中记录的所有文件系统。
- -h：显示帮助。
- -n：卸载时不要将信息存入/etc/mtab 文件中。
- -r：若无法成功卸载，则尝试以只读的方式重新挂入文件系统。
- -t<filesystem type>：仅卸载选项中所指定的文件系统。
- -v：执行时显示详细的信息。
- -V：显示版本信息。
- [filesystem]：除了直接指定文件系统外，也可以用设备名称或挂载点来表示文件系统。

注意：虽然利用设备名或挂载点都能 umount 文件系统，不过最好还是通过挂载点卸载，以免使用绑定挂载（一个设备，多个挂载点）时产生混乱。

8.2 使用交换空间

8.2.1 添加交换空间

交换空间即 8.1 节所讲的交换分区，一般是 swap 格式的。很多时候用户会发现默认的 swap 交换空间无法满足某些软件的最低安装要求。例如 Linux 下的 Oracle，当有一个用户分配的物理内存为 1G，则还需要 2G 的 swap 交换空间，可是默认的交换空间只有 1G，就需要对其进行扩展。添加 swap 交换空间的步骤如下。

（1）检查磁盘上的剩余空间，交换分区要用到磁盘，因此要确保有足够的空间来用作 swap 交换空间。推荐使用 KVM 工具。检查好后可以进入下一步。

（2）在一个独立的文件系统中添加一个 swap 交换文件，并且在/opt/image 目录中添加 2G 的 swap 交换文件。添加交换文件的命令如下：

```
dd if=/dev/zero of=/opt/image/swap bs=1024 count=2048000
```

如果成功，一般会在几分钟到十几分钟后看到显示器上返回如下结果：

```
2048000+0 records in
2048000+0 records out
2097152000 bytes (2.1 GB) copied, 272.867 seconds, 7.7 MB/s
```

（3）创建并设置交换分区，命令为 mkswap，使用方法如下：

```
mkswap /opt/image/swap
```

如果成功，显示器上会返回如下结果：

```
Setting up swapspace version 1, size = 2097147 kB
```

（4）检查现有的交换空间大小，命令为 free：

```
free -m
```

如果成功，显示器上会返回如下结果：

```
Total  used        free        shared          buffers          cached
Mem: 1011          989         21          0               1                875
-/+ buffers/cache: 112         898
Swap:1027 0             1027
```

也可以通过检查 meminfo 文件来实现，方法如下：

```
grep SwapTotal    /proc/meminfo
```

（5）启动新增加的 2G swap 交换空间，命令为 swapon：

```
swapon /opt/image/swap
```

（6）确认新增加的 2G 交换空间已经生效，命令为：

```
free -m
Total  used        free        shared          buffers          cached
Mem: 1011          995         15          0               4                877
-/+ buffers/cache: 113         897
Swap:3027          0           3027
```

也可以通过检查 meminfo 文件来实现，方法如下：

```
grep SwapTotal    /proc/meminfo
```

（7）修改/etc/fstab 文件，使新添加的 swap 交换区启动自动运行，方法是在文件的最后加入如下字符串：

```
/opt/image/swap  swap swap defaults 0 0
```

至此完成添加 swap 交换分区的工作。

8.2.2　删除交换空间

当交换区不再被需要，就要及时删除交换分区以释放被占用的空间。交换文件一般都具有相当大的体积。及时释放它们对提高系统性能、增加资源利用率有很重大的意义。但是删除之前一定要确认该空间不再被需要。删除的过程如下所示：

（1）必须先让硬盘驱动器不能继续被使用，即分区不能被挂载，交换分区不能被启用。最简单的办法是在救援模式下启动系统。如果驱动器不包含任何被使用的分区，则可以使用 swapoff 命令（回忆一下启动交换分区的命令是什么）来关闭硬盘驱动器上的所有交换空间。

（2）以 root 身份进入系统，输入以下命令来确定交换分区已被禁用（这里的/dev/hdb2 是交换分区）：

```
swapoff /dev/hdb2
```

（3）从/etc/fstab 中删除该项目。

（4）使用 parted 命令来删除分区：

```
parted /dev/hdb
```

注意：这里的/dev/hdb 是要删除存在于其中的 swap 交换区的硬盘驱动器的设备名，用户可以根据需要换成自己的设备名。

（5）执行上一步后会进入 parted 提示，在这个提示下输入 print，查看现存的分区并判定想删除的交换分区的次设备号，然后输入 rm MINOR，这里的 MINOR 是用户想删除的分区的次设备号。这一点很重要，因为对系统的改变会立即生效，这个过程是不可逆的，因此必须一次性键入正确的次设备号。

（6）键入 quit 退出 parted 提示符。

上述步骤全部完成后就已经在系统中删除了交换分区，但交换文件仍然存在于磁盘之上，且占用着大量的空间，因此还要执行如下操作删除交换文件：

1）以 root 身份进入系统，禁用交换文件，命令如下：

```
swapoff /swapfile
```

此处的 swapfile 就是交换文件，每个系统、每个交换文件的名称都会不同，用户应该替换成自己的交换文件名。

2）从/etc/fstab 中删除该项目。

3）删除文件，命令如下：

```
rm /swapfile
```

至此，一个交换空间就完全被删除了，空间得以释放。

8.3　权限设置

Linux 系统中的每个文件和目录都有访问许可权限，用户能否访问一个文件，以什么方式

来访问文件是由文件的权限属性决定的。通过修改文件的访问权限，文件的所有者能够决定哪些用户可以读取、修改和运行文件。对于系统和用户来说，谨慎设置每一个文件和目录的访问权限是 Linux 系统安全的基础。

8.3.1 文件与目录的权限

文件或目录的访问权限分为读、写和执行三种。系统用 r 代表读权限、w 代表写权限、x 代表执行权限。r、w 和 x 具体代表的含义如表 8.4 所示。

表 8.4 文件或目录的访问权限

	r（读）	w（写）	x（执行）
文件	允许读文件的内容	允许写和修改文件	允许运行程序或 Shell 程序
目录	允许列出目录中的内容	允许在目录中建立或删除文件、子目录	允许进入和退出目录

如果拥有一个文件，那么该用户自动拥有对该文件的读、写和可执行权限，以便于对文件进行阅读和修改。用户也可根据需要把访问权限设置为需要的任何组合。

有三种不同类型的用户可对文件或目录进行访问：文件/目录所有者（owner）、同组用户（group）和其他用户（other）。文件所有者一般是文件的创建者，它可以允许同组用户访问该文件，还可以将文件的访问权限赋给系统中的其他用户。同组用户指拥有该文件的用户组中的任何用户。其他用户即不属于拥有该文件的用户组的某一用户。

Linux 是一个多用户操作系统，它允许多个用户同时登录和工作。因此 Linux 将一个文件或目录与一个用户和组联系起来。每个文件或目录的访问权限都有三组，每组用三位表示，分别为文件属主的读、写和执行权限；与属主同组的用户的读、写和执行权限；系统中其他用户的读、写和执行权限。

在命令行模式下用 ls -l 命令显示文件或目录的详细信息，如下所示：

```
[root@RHEL5 ~]# ls -l /home/John
总计 8
drwxr-xr-x 2 John John 4096 02-13 22:01 Desktop
```

其中，与文件权限相关联的是第一、第三和第四个字段。第三个字段表示文件所有者，第四个字段表示文件所属的组，而第一个字段表示文件的访问权限。以文件 Desktop 为例，文件所有者是 John，所属的组是 John，文件的访问权限是 drwxr-xr-x。文件访问权限字段由 10 个字符组成，可以把它们分为四组，具体含义如表 8.5 所示。

表 8.5 文件访问权限字段的分组

权限字段	分组	drwxr-xr-x
文件类型	第 1 个字符	d（目录）
属主权限标志	第 2～4 个字符	rwx
属组权限标志	第 5～7 个字符	r-x
其他用户权限标志	第 8～10 个字符	r-x

第 1 个字符为文件类型。由于 Linux 系统把设备、目录、文件都当作是文件来处理，因此

该字符表明此文件的类型，其字符与对应的含义如表 8.6 所示。

表 8.6　文件访问权限字段的第 1 个标志字符与文件类型的对应关系

文件标志	文件类型	示例
-	普通文件	数据文件、ASCII 码文本文件或者程序
d	目录	/bin、/mnt 和/home
b	块设备	/dev/hda（第一个 IDE 硬盘）
c	字符设备	/dev/ttyS0（与 DOS 的串口 1 等同）
s	套接字	/dev/log
p	命名管道	/dev/initctl（与“|”等同）
l	符号链接	/etc/grup.conf->../boot/grub.conf

权限字段的后九位权限标志与三类用户对应，分为三组，每组三个，分别用 r、w、x 来表示，分别对应 owner、group 和 other。当没有相应的权限时，该权限对应的位置用短线“-”标志。作为 root 用户，它自动拥有了对所有文件和目录的读、写和执行权限，所以没有必要明确指定它的权限。其他三类用户则可以在单个文件或目录的基础上分别授予或撤销权限。

注意：目录权限的优先级要大于文件权限的优先级，目录的权限将会覆盖该目录中文件的权限。

8.3.2　设置文件和目录权限

在 Linux 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或目录，就需要改变文件的权限。改变权限的方法有多种，但只有当文件的所有者对某个文件有写权限，他才能够改变该文件的权限（但 root 用户能对所有文件进行权限设置），而其他用户无权改变该文件的权限设置。

1. 使用文件管理器来设置文件与目录的权限

在 Red Hat Enterprise Linux 5 中，使用图形界面的文件管理器能够轻松地设置文件与目录的权限。在“文件管理器”窗口中右击要设置权限的文件或目录，在弹出的快捷菜单中选择“属性”命令，然后在打开的属性对话框中单击“权限”标签，则可打开“权限”选项卡，如图 8.2 所示。

在“权限”选项卡的“访问权限”选项组中，列出了该文件的所有者、文件所有者所在组和其他用户的读、写和执行权限。要想改变其权限，可以通过在相应的下拉框中选择“可查看并修改内容”、“可查看内容”或“禁止”。

另外，可单击“高级权限”按钮打开“高级权限”对话框，如图 8.3 所示。

在“访问权限”选项组中有三个特殊标志的复选框：设置 UID、设置 GID 和粘性，这些是文件或目录的特殊权限。其含义如下。

（1）设置 UID。对可执行文件，若选中“设置 UID”，则将文件所有者的执行权限的设置位表示成 s（当开启执行权限时）或 S（当关闭执行权限时）。即表明当该文件被执行时将把进程的所有者设置为该文件的所有者。

图 8.2 “权限”选项卡

图 8.3 “高级权限”对话框

例如，假设可执行文件 test.out 的文件所有者是 John，原来的权限为“rwxrw-r--”，即该文件只有所有者具有执行权限，而其他用户不能执行该文件。如果现在将 test.out 文件的“设置 UID”选中，其权限变为“rwsrw-r--”。那么当其他用户 David 执行 test.out 文件时，系统就会创建相应的进程，并将该进程的所有者 David 设置成可执行文件 test.out 的所有者，并具有该文件所有者的相应权限。

由此可见，有 s 的权限就可能会拥有“特权”，特别是当可执行文件的所有者是 root 用户

时，黑客经常会利用这种权限对系统进行攻击。所以，在整个系统中特别是 root 本身要谨慎对文件设置 s 权限。

（2）设置 GID。同样的原则也适用于设置 GID，只不过将文件所有者变成了文件所有者所在的用户组。若选中“设置 GID”，则将文件所有者所在组的执行权限的设置位表示成 s（当开启执行权限时）或 S（当关闭执行权限时）。即表明当该文件执行时将把进程的所有者所在组设置为该文件所有者所在组。

（3）粘性。对文件或目录若选中“粘性”，则将其他用户的执行权限的设置位表示成 t（当开启执行权限时）或 T（当关闭执行权限时）。粘性意味着该文件或目录只有其所有者才可以删除，即使某个同组用户具有和所有者同等的权限。

例如，若用 ls -l 命令查看/tmp 目录的详细信息如下：

```
drwxrwxrwt   13 root root   4096 02-13 23:52 tmp
```

系统已默认将/tmp 目录的其他用户的执行权限设为 t。由于系统中的/tmp 和/var/tmp 目录可供所有用户及应用程序临时存放文件，任何用户都可拥有完整的权限进入该目录。因此，如果不采取相应的措施，任何用户都能删除其中的文件，这会给其他用户带来危险。但是当给/tmp 和/var/tmp 目录设置了“粘性”这项特殊权限后，存放在其中的文件就仅允许其所有者删除、移动，这就避免了其他用户的骚扰。

2. 使用 chmod 命令改变文件与目录的权限

确定了一个文件的访问权限后，用户可以利用 chmod 命令来重新设定不同的访问权限。根据权限的不同表示方式，chmod 命令有两种用法，一种是包含字母和操作符表达式的文字设定法；另一种是包含数字的数字设定法。

（1）文字设定法。

命令格式：chmod [选项] [操作对象] [操作符] [权限] 文件或目录名

命令中各项的含义为：

选项：常用的选项主要是-R，表示递归改变目录及其子目录的权限。

操作对象：操作对象可以是下述字母中的任一个或者各字母的组合。

- u 表示“用户（user）”，即文件或目录的所有者。
- g 表示“同组（group）用户”，即文件所有者所在组的同组用户。
- o 表示“其他（others）用户”。
- a 表示“所有（all）用户”，它是系统默认值。

操作符：对权限进行设定。

- + 表示添加某个权限。
- - 表示取消某个权限。
- = 表示赋予指定权限并取消原有的权限。

权限：设置权限可用下述字母的任意组合。

- r 表示可读。
- w 表示可写。
- x 表示可执行。
- s 表示在文件执行时把进程的所有者或所有者所在组设置为该文件的所有者或文件所有者所在组。“u+s”设置文件的 UID 为 s，“g+s”设置 GID 为 s。

- t 表示粘附。
- u 表示与文件所有者拥有一样的权限。
- g 表示与文件所有者所在组的用户拥有一样的权限。
- o 表示与其他用户拥有一样的权限。

【例 8.1】使用 chmod 命令的文字设定法修改文件 text 的访问权限，执行结果如下所示：

```
[John@RHEL5 ~]$ ls -l text
-rw-r--r-- 1 John John 0 02-13 23:59 text
// 设定所有用户对文件 text 具有读和执行的权限
[John@RHEL5 ~]$ chmod a+rx text
[John@RHEL5 ~]$ ls -l text
-rwxr-xr-x 1 John John 0 02-13 23:59 text
// 设定文件 text 的所有者和同组用户写的权限，其他用户删除执行权限
[John@RHEL5 ~]$ chmod ug+w,o-x text
[John@RHEL5 ~]$ ls -l text
-rwxrwxr-- 1 John John 0 02-13 23:59 text
```

（2）数字设定法。

命令格式：chmod [选项] [权限] 文件或目录名

数字设定法是与文字设定法功能等价的设定方法，即用数字表示权限。数字表示的属性的含义为：0 表示没有权限，1 表示可执行权限，2 表示可写权限，4 表示可读权限，然后将其相加。所以数字属性的格式应为 3 个从 0～7 的八进制数，其顺序是（u）、（g）、（o），其他与文字设定法基本一致。

可对照表 8.7 计算八进制的权限值。

表 8.7　计算八进制的权限值

文件所有者（u）	同组用户（g）	其他用户（o）
r　w　x	r　w　x	r　w　x
4　2　1	4　2　1	4　2　1

使用表 5.8 可以方便地计算出相应的权限值，只要分别针对文件所有者、同组用户和其他用户把相应权限下面的数字相加即可。

例如，文件 myfile 具有这样的权限：

```
r　w　-　　r　-　-　　r　-　-
4+2　　4　　4
```

把相应的权限位所对应的值加在一起，就是 644。

【例 8.2】使用 chmod 命令的数字设定法设定文件 text 的访问权限，执行结果如下所示：

```
// 设定文件所有者读和写的权限，同组用户和其他用户读权限
[John@RHEL5 ~]$ chmod 644 text
[John@RHEL5 ~]$ ls -l text
-rw-r--r-- 1 John John 0 02-13 23:59 text
// 设定文件所有者读、写和执行的权限，所有其他用户读的权限
[John@RHEL5 ~]$ chmod 744 text
[John@RHEL5 ~]$ ls -l text
```

-rwxr--r-- 1 John John 0 02-13 23:59 text

注意：只有文件的所有者或 root 用户才有权使用 chmod 命令。

8.3.3　改变文件与目录的所有者和所有者所在组

一般情况下，文件所有者、文件所有者所在组是创建文件时用户和用户所在组。一旦用户拥有某个文件，就可以改变它的所有权，把它的所有权交给另外一个/etc/passwd 文件中存在的合法用户。在改变一个文件的所有权时，相应的 suid 也将被清除，这是出于安全性的考虑。只有文件所有者和 root 用户可以改变文件的所有权。一旦将文件的所有权交给另外一个用户，就无法再重新收回它的所有权。如果真的需要这样做，则只有求助于系统管理员了。

1. 使用文件管理器改变文件与目录的所有权

在文件管理器窗口中打开文件或目录的属性对话框，然后单击“权限”标签，打开如图 5.10 所示的“权限”选项卡。

在“权限”选项卡的“所有者”选项组中，分别在“用户”和“群组”对应的文本框中输入所有者和所有者所在组，就可改变该文件或目录的所有者、所有者所在组。

2. 使用 chown 命令改变文件与目录的所有者

功能描述：该命令用于改变某个文件或目录的所有者和所属的组，即可以向某个用户授权，使他变成指定文件或目录的所有者或者改变文件所属组。

命令格式：chown [选项] 用户和用户组的组合 文件或目录名

命令中各项的含义为：

- 选项：常用选项主要是-R，表示递归地改变指定目录下的所有文件及子目录的文件所有者、文件所有者所在组。
- 用户和用户组的组合：用户可以是用户名或用户标识码（UID），用户组可以是用户组名或用户组标识码（GID）。其组合方式有四种，如表 8.8 所示。

表 8.8　用户和用户组的组合方式

组合方式	说明
用户	只改变文件所有者
:用户组	只改变文件所有者所在组
用户:	改变文件的所有者，并且将文件所有者所在组改为用户所在组（主要组）
用户:用户组	改变文件的所有者、文件所有者所在组

- 文件或目录名：可以是以空格分开的需要被改变的文件或目录名列表，支持通配符。

说明：此命令只有文件所有者或 root 用户才有权使用。

【例 8.3】使用 chown 命令将/home/John/myfile 目录下的所有文件和子目录的所有者改为用户 John，所有者所在组改为用户组 users，执行结果如下所示：

```
[root@RHEL5 ~]# ls -l /home/John
总计 20
drwxr-xr-x 2 John John 4096 02-13 22:01 Desktop
drwxr-xr-x 2 root root 4096 02-14 00:06 myfile
-rwxr--r-- 1 John John    0 02-13 23:59 text
[root@RHEL5 ~]# chown -R John:users /home/John/myfile
```

```
[root@RHEL5 ~]# ls -l /home/John
总计 20
drwxr-xr-x 2 John John    4096 02-13 22:01 Desktop
drwxr-xr-x 2 John users 4096 02-14 00:06 myfile
-rwxr--r-- 1 John John         0 02-13 23:59 text
```

3. 使用 chgrp 命令改变文件与目录的所有者所在组

命令格式：chgrp [选项] 用户组 文件或目录名

命令中各项的含义为：

- 选项：常用选项主要是-R，表示递归地改变指定目录下的所有文件及子目录的文件所有者所在组。
- 用户组：用户组可以是用户组名或用户组标识码（GID）。
- 文件或目录名：可以是以空格分开的要被改变的文件或目录名列表，支持通配符。

此命令只有文件所有者或 root 用户才有权使用。

【例 8.4】使用 chgrp 命令将/home/John/myfile 目录下的所有文件和子目录的所有者所在组改为用户组 John，执行结果如下所示：

```
[root@RHEL5 ~]# chgrp -R John /home/John/myfile
[root@RHEL5 ~]# ls -l /home/John
总计 20
drwxr-xr-x 2 John John 4096 02-13 22:01 Desktop
drwxr-xr-x 2 John John 4096 02-14 00:06 myfile
-rwxr--r-- 1 John John         0 02-13 23:59 text
```

8.4 硬盘分区

8.4.1 硬盘分区简介

Linux 中的分区是一个比较容易让初学者费解的地方。Linux 规定，每一个硬盘设备最多只能有四个主分区（含扩展分区），扩展分区也要占一个主分区号，即一个硬盘中的主分区和扩展分区总数最多为四个。

主分区的物理第一扇区（0 面 0 道 1 扇区）是引导扇区，存放有 MBR（主引导记录），其作用就是引导加载操作系统。而逻辑分区没有 MBR，一般是用来存放数据（当然主分区也用来存放数据，只不过 MBR 位于其上）。这就是主分区和扩展分区及逻辑分区的最大区别。在指定安装引导 Linux 的 boot loader 时，都必须指定装在主分区上。

Linux 规定主分区（或者扩展分区）占用 1～16 中的前 4 个号码。例如第一个 IDE 硬盘的主分区（或者扩展分区）就会占用 hda1、hda2、hda3、hda4，而逻辑分区则占用从 hda5 到 hda16 的剩下 12 个号。由此可见 Linux 系统中每块硬盘总共最多有 16 个分区。另外 Linux 还规定逻辑分区必须建立在扩展分区而不是主分区之上。这就是为什么虽然扩展分区能够提供更灵活的分区模式，但却不能引导加载操作系统的原因。

作者在这里强烈建议，有关于硬盘、分区的基础知识，读者最好查看由王路群教授主编、中国水利水电出版社出版的《计算机病毒原理及防范技术》一书第二章，里面从磁盘结构到分区再到文件系统划分都有非常详细且专业的讲解说明。

8.4.2　使用 fdisk 进行硬盘分区

fdisk 是一个传统的管理分区的工具。需要注意的是，MS-DOS 和 Windows 下也有 fdisk，功能也差不多，但是 Linux fdisk 的适用面更广，可以用它将分区的类型更改成一百多种类型。

命令格式：fdisk [-b <分区大小>][-uv][外围设备代号]

fdisk [-l][-b <分区大小>][-uv][外围设备代号...]

fdisk [-s <分区编号>]

命令说明：fdsik 能将磁盘划分为若干个区，同时也能为每个分区指定分区的文件系统。需要注意的是，Linux fdisk 采用了传统的问答式界面，而非类似 DOS fdisk 的 cfdisk 互动式操作界面。初次接触者会因此觉得使用不便，但其功能要远远强于 DOS fdisk。

参数说明：

- -b<分区大小>：指定每个分区的大小。
- -l：列出指定的外围设备的分区表状况。
- -s<分区编号>：将指定的分区大小输出到标准输出上，单位为区块。
- -u：搭配“-l”参数列表，会用分区数目取代柱面数目，来表示每个分区的起始地址。
- -v：显示版本信息。

【例 8.5】查看当前外围设备的分区表情况：

```
fdisk -l
```

结果如图 8.4 所示，具体情况会因各台机器环境而不同。

图 8.4　查看当前外围设备的情况

这里显示的信息十分完备，解读如下：

- Disk /dev/hda: 80.0 GB, 80026361856 bytes：硬盘总容量 80G。
- 255 heads, 63 sectors/track, 9729 cylinders：该硬盘有 255 个磁面，63 个扇区，9729 个柱面。
- Units = cylinders of 16065 * 512 = 8225280 bytes：容量是 8225280 bytes=8225.280 K=8.225280M，其中 512B 为一个扇区的大小。有关磁盘结构问题再次推荐读者阅读《计算机病毒原理及防范技术》一书。
- Device Boot Start End Blocks Id System

  …

  /dev/hda10 7752 9729 15888253+ 83 Linux

这一部分是磁盘的分区说明。Linux 系统的通用表示法为 hd*x 或 sd*x，其中*=a、b、c …，x=1、2、3…，hd 一般是指 IDE 硬盘，sd 一般为 SCSI 或移动存储设备。

8.5 磁盘配额

8.5.1 磁盘配额简介

磁盘配额英文为 quota，字面含义为限额的意思。在 Linux 中，quota 的出现是为了给每个用户分配可用磁盘空间的大小。

由于 Linux 系统是多用户多任务的，很多时候会出现多人共用一个硬盘的情况。如果其中有某些使用者用掉很多硬盘空间，则必然导致其他使用者的空间会因此降低，侵害了他人权益。所以管理员应该适当地开放硬盘的权限给具体使用者，合理分配系统资源。

quota 在 Linux 中用来管理硬盘空间。它是一个系统模块，使用时需要注意以下三点限制：

（1）quota 模块在运行过程中会对整个 partition（分区）上锁。例如，有一个用户的/dev/hda1 挂载在/home 下，则当 quota 运行时，整个/home 下的所有目录都会受到同样的限制。

（2）不是所有版本的 Linux 都支持 quota，其内核必须支持 quota 模块才能使用它。不过可以通过内核扩展来实现在每个 Linux 机上都运行 quota。Mandrake Linux 比较好，默认开放 quota 模块。如果没有则需要用户自行编译内核。编译过程中一定要特别留意是否已经真的开放了 quota 模块。

（3）目前流行的新版 Linux 发布版，如 Mandrake 9.0、Red Hat 8.0 等都是使用的 Kernel 2.4.xx 内核，它支持 quota 模块，其配置文件 aquota.user、aquota.group 是由老版本的 quota.user、quota.group 升级而来。当然，即使用户使用的是老版本的 quota，也可以用 convert quota 程序升级到新的版本，其兼容性非常好。

quota 的限制内容主要有以下几个：

（1）soft limit：弹性容量限制（可超过）。这个限制给使用者一个缓冲时间，在缓冲期内，可以使用 soft 值的空间，但是必须要在缓冲时间达到前将占用的磁盘空间释放出来，将点用量降低至 soft 值之下。

（2）hard limit：刚性容量限制（不可超过）。一般 hard limit 设定要高于 soft limit，这个限制是不能打破的。设置它的作用在于提醒使用者不要超过配额。比如配额有 100M，则 hard

limit 一般就设定为 100M，假设 soft limit 为 75M，则当使用者用掉了 80M 时，系统就会发出警告。使用者收到警告信息后就要在缓冲时间内将占用量降低至 75M 以内。

（3）缓冲时间：缓冲时间的作用上面已经描述得非常清楚了。当使用者使用的空间介于 soft limit 和 hard limit 之间时，在设定的缓冲时间内，必须想办法将占用的磁盘空间降低到 soft limit 以内。一旦磁盘空间占用超过 soft limit，缓冲时间计时器就会自动启动，而当空间占用降到 soft limit 以内，这个计时器就自动退出运行，等待下一个弹性时间的到来。

8.5.2　配置磁盘

配置磁盘用的就是 quota 命令族。quota 命令族分为两大类，①查询功能，有 quota、quotacheck、quotastats、warnquota、repquota；②编辑功能，有 edquota、setquota。下面就最常用的几个命令加以说明。

1. quota 命令

命令格式：quota [-guvs] [user,group]

命令说明：显示当前某个群组或者某个使用者的 quota 限值。

参数说明：

- -g：显示 group 群组
- -u：显示 user
- -v：显示 quota 的值
- -s：以 inod 或硬盘空间来显示

【例 8.6】显示目前使用者的 quota 值：

```
quota -guvs
```

【例 8.7】显示使用者 user 的 quota 值

```
quota -uvs user
```

2. quotacheck 命令

命令格式：quotacheck [-auvgm] /yourpath

命令说明：扫描某个磁盘的 quota 空间。由于磁盘在运行时会有文件大小的改变情况出现，因此可能出现扫描结果错误。为解决这个问题，当使用 quotacheck 命令时，则被扫描区域将自动设为只读属性。扫描完成后，扫描结果将会写入该扇区最顶端。如果是初次扫描，扫描完成后会产生 aquota.user、aquota.group 两个文件，放在/home/aquota.xxx 目录下。如果已建立 quota，则只更新这两个文件。此外要特别注意的是当运行 quota 时，如果要重启，就必须先关闭 quota，以免发生意想不到的问题。

参数说明：

- -a：扫描所有在/etc/mtab 中已经 mount 的具有 quota 支持的磁盘
- -u：扫描使用者的文件与目录
- -v：显示扫描过程
- -g：扫描群组使用的文件与目录
- -m：强制进行 quotacheck

特别说明，如果碰到无法完成 quotacheck 操作的情况，这是由于新版的 Linux 发布版在 quota 模块上的设计问题导致。解决方法很简单，只需要用户手工建立 quota file 即可。

3. edquota 命令

命令格式：edquota [-u user] [-g group] [-t] 或 edquota -p user_demo -u user

命令说明：编辑使用者或使用者组的 quota 值。用 edquota -u username 或 edquota -g groupname 来设置 quota 值是最常用的。但是如果用户很多，且很多人或组的 quota 值都一样，这样手工一一设置很繁琐，于是就有了复制法，即将某个人或组的设定好的 quota 值复制到其他人或组上。

参数说明：

- -u：编辑 user 的 quota 值
- -g：编辑 group 的 quota 值
- -t：编辑缓冲时间
- -p：设定值复制操作

【例 8.8】将使用者 test 的 soft limit 和 hard limit 分别设为 0.75M 和 1M：

```
edquota -u test
```

结果如图 8.5 所示。

```
[root @test /root ]# edquota -u test
Disk quotas for user test (uid 501):
  Filesystem   blocks       soft      hard      inodes       soft      hard
  /dev/hda3    8            0         0         5            0         0
```

图 8.5 进入设定

设置限制值，注意不要修改 blocks 值，如图 8.6 所示。

```
[root @test /root ]# edquota -u test
Disk quotas for user test (uid 501):
  Filesystem   blocks       soft      hard      inodes       soft      hard
  /dev/hda3    8            750       1000      5            750       1000
```

图 8.6 设置 quote 值

对图中的几个数值说明如下：

① filesystem：分区 partition，即要设置的空间路径，本例中就是/dev/hda3。

② blocks：当前使用者（本例为 test，其用户号 uid=501）在 partition（本例为/dev/hda3）中所占用的磁盘空间，单位为 K。这个数值由 quota 自动计算得到，不要修改。

③ soft 与 hard：上面已有详细讲述，注意数值的单位是 K。

④ inodes：当前占用的节点状态，也是由 quota 自动计算得到，不要修改。

【例 8.9】将 user1 的 quota 值复制给 user2：

```
edquota -p user1 -u user2
```

【例 8.10】将缓冲时间设为 2 个小时：

```
edquota -t
```

结果如图 8.7 所示。

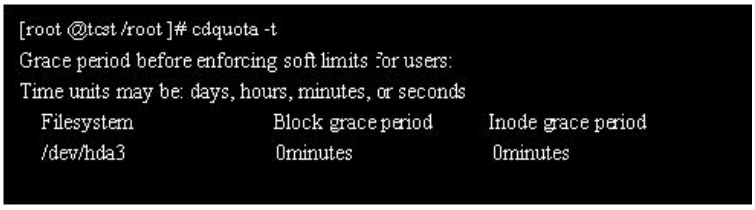

图 8.7　设置缓冲时间

只需要将下面的 0minutes 改为 120minutes 即可。

4. quotaon 命令

命令格式：quotaon [-a] [-uvg directory]

命令说明：启动 quota。这个指令是通过启动 aquota.group 和 aquota.user 这两个文件（上面有讲解）实现的，因此如果要用 quotaon 命令，就必须先执行至少一次 qutoacheck（想想为什么）。

参数说明：

- -a：启动所有的 quota（系统自动寻找/etc/mtab 中的设定）
- -u：启动使用者的 quota 设定
- -g：启动组的 quota 设定
- -s：显示信息

【例 8.11】只激活/home 下的使用者的 quota 设定，而不激活 group 的设定：

```
quotaon -uv /home
```

5. quotaoff 命令

命令格式：quotaoff -a

命令说明：关闭 quota 的限制。

参数说明：

-a：关闭所有的 quota 设定（系统自动寻找/etc/mtab 中的设定）。

8.6　项目实训：文件和目录权限的设定

项目需求：

Linux 操作系统

企业员工情况：

培训师：Tr1

人力资源部：员工 h1、员工 h2

财务部：员工 f1、员工 f2

Tr1 希望对人力资源部和财务部各自派出的两名员工进行有关文件和目录权限的培训，然后让他们回到本部门后再去传授给其他同事。现在 Linux 在公司中已经成为了主流操作系统，但是安全问题仍然时时困扰着 Tr1，因此他觉得有必要做这个培训。

解决方案：

（1）先用 ls 命令配-l 参数查看目录内的情况，这是最基本要求：

```
ls -l
```

结果如图 8.8 所示。

```
[root@localhost ~]#ls -l
total 368
-rw-r--r--    1 root root    12172       DEC     7 08:11   TEST1
drwxr-xr-x    2 root root    48          DEC     8 18:50   TEST2
-r--r--r--    1 root root    331844      DEC     18 19:09  TEST3
drwxr-xr-x    2 root root    48          DEC     11 19:26  TEST4
-rwxr-xr-x    1 root root    9776        DEC     12 11:42  TEST5
-rwxr-xr-x    1 root root    9776        DEC     6 19:18   TEST6
-rwxr-xr-x    1 root root    512         DEC     7 11:07   TEST7
drwxr-xr-x    2 root root    48          DEC     8 42:36   TEST8
```

图 8.8　当前系统所示情况

从这个结果可知文件 TEST1 的权限是 644（回忆一下如何计算），现在要将 TEST1 的权限改为 777：

chmod 777 TEST1

查看执行结果：

ls -l

在当前目录下可以看到如图 8.9 所示的结果。

```
[root@localhost ~]#ls -l
total 368
-rwxrwxrwx    1 root root    12172       DEC     16 21:08  TEST1
drwxr-xr-x    2 root root    48          DEC     8 18:50   TEST2
-r--r--r--    1 root root    331844      DEC     18 19:09  TEST3
drwxr-xr-x    2 root root    48          DEC     11 19:26  TEST4
-rwxr-xr-x    1 root root    9776        DEC     12 11:42  TEST5
-rwxr-xr-x    1 root root    9776        DEC     6 19:18   TEST6
-rwxr-xr-x    1 root root    512         DEC     7 11:07   TEST7
drwxr-xr-x    2 root root    48          DEC     8 42:36   TEST8
```

图 8.9　TEST1 的权限被成功修改为 777

很明显可以看到文件 TEST1 的权限已经被成功修改为 rwxrwxrwx，即 777。

还可以对文件加上特殊权限，这时就要用到 4 位数字，但是算法还是一样的。特殊权限的对应数值为：

s 或 S（SUID）：对应 4。

s 或 S（SGID）：对应 2。

t 或 T：对应 1。

现在将 TEST1 的权限再次修改为-rwS--S--T，输入：

chmod 7600 TEST1

ls -l

在当前目录下可以看见如图 8.10 所示的结果。

```
[root@localhost ~]# chmod 7600 TEST1
[root@localhost ~]# ls -l
total 368
-rwS--S—T   1 root root    12172       DEC     16 24:19  TEST1
drwxr-xr-x  2 root root    48          DEC     8 18:50   TEST2
-r--r--r--  1 root root    331844      DEC     18 19:09  TEST3
drwxr-xr-x  2 root root    48          DEC     11 19:26  TEST4
-rwxr-xr-x  1 root root    9776        DEC     12 11:42  TEST5
-rwxr-xr-x  1 root root    9776        DEC     6 19:18   TEST6
-rwxr-xr-x  1 root root    512         DEC     7 11:07   TEST7
drwxr-xr-x  2 root root    48          DEC     8 42:36   TEST8
```

图 8.10　特殊权限修改成功

还可以高效地一次性修改整个当前目录下所有文件的权限，甚至包括子目录中的文件权限也可以一次性修改，这就要用到-R 参数，-R 用来递归设置。

将/home/user 目录的权限设置为 rwxrwxrwx：

```
[root@localhost ~]# chmod 777 /home/user
```

上面的做法仅仅只对/home/user 目录中的文件有效，如果里面还有子目录则无效。不过可以用下面的方法连同整个/home/user 目录与其中的文件和子目录的权限都一次性设置为 rwxrwxrwx。

```
[root@localhost ~]# chmod -R 777 /home/user
```

（2）使用命令 chown 改变目录或文件的所有权。

文件与目录自身的权限的修改方法上面已经有详细的说明，下面介绍对文件和目录的所有权及所属用户组的修改方法。

还是先执行 ls -l 看看目录情况，一定要养成这个好习惯：

```
ls -l
```

可以看到如图 8.10 所示的结果。这个结果清楚表明文件 TEST1 的所属用户组为 root，所有者为 root。现在要将 TEST1 的所有权转移到用户 user。

```
chown user TEST1
```

可以看到如图 8.11 所示的结果。

```
[root@localhost ~]# chown user TEST1
[root@localhost ~]# ls -l
total 368
-rwS--S—T   1 user root    12172       DEC     16 39:26  TEST1
drwxr-xr-x  2 root root    48          DEC     8 18:50   TEST2
-r--r--r--  1 root root    331844      DEC     18 19:09  TEST3
drwxr-xr-x  2 root root    48          DEC     11 19:26  TEST4
-rwxr-xr-x  1 root root    9776        DEC     12 11:42  TEST5
-rwxr-xr-x  1 root root    9776        DEC     6 19:18   TEST6
-rwxr-xr-x  1 root root    512         DEC     7 11:07   TEST7
drwxr-xr-x  2 root root    48          DEC     8 42:36   TEST8
```

图 8.11　转移所有权给用户

还可以转移 TEST1 的所属组到组 users（注意不是 user）：

```
chown :users TEST1
```

结果如图 8.12 所示。

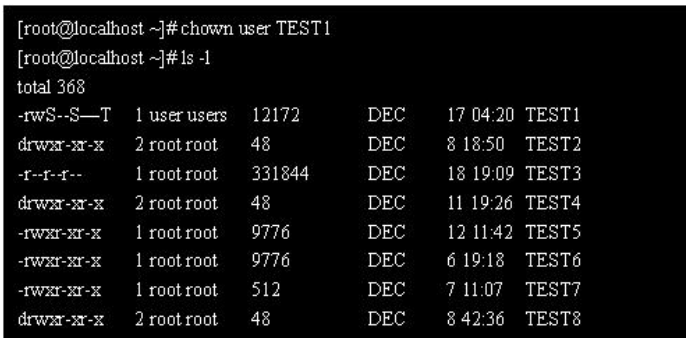

图 8.12　转移所有权给组

-R 参数仍然可以用于递归修改，以提高工作效率。

习题八

1．分别说说 ext2、ext3、sysfs、/proc 和 swap 的特点及应用范围。
2．如何装载和卸载一个移动硬盘：
3．权限的数字表示法是怎样计算的？
4．如何使用 chmod 命令来修改权限？
5．quota 命令的使用方法有哪些？