

# 第9章 磁盘和文件系统管理

分区、几个步骤、重点除了9.6  
特殊权限，三种特殊权限的作用  
文件目录修改用户和组

9.1

Linux硬盘分区

9.2

Linux文件系统简介

9.3

创建文件系统

9.4

挂载和卸载文件系统

9.5

设置开机自动挂载文件系统

9.6

使用交换空间

9.7

权限设置

在Linux系统中，如果需要在某个磁盘上存储数据，则需要将磁盘进行分区，创建文件系统，最后将文件系统挂载到目录下才可以。

为了控制文件和目录的访问，可以设置文件和目录的访问权限，甚至可以更改文件和目录的所有权。

## 9.1 Linux硬盘分区

在安装Fedora 17系统时，可以对硬盘进行分区，本节主要介绍安装系统之后如何使用fdisk命令进行分区。

## 9.1.1 硬盘分区知识

要对计算机硬盘进行分区，首先需要知道什么是分区、分区有哪些类型以及什么是格式化等概念。

# 1. 什么是硬盘分区

分区就是硬盘的“段落”，如果用户希望在计算机上安装多个操作系统，将需要更多的分区。假设要同时安装Windows XP和Windows 7，那么需要两个分区，原因是不同的操作系统原则上采用不同的文件系统。

硬盘分区的原因是要使用不同的文件系统

## 2. 分区类型

硬盘分区一共有3种：主分区、扩展分区和逻辑分区。

硬盘技术规定主分区最多支持4个分区  
主分区和扩展分区数字加起来最多到4，5及以上是逻辑分区的编号  
扩展分区：主分区以外的分区，不能直接使用，要占用一个主分区的数字（最多4个）  
逻辑分区：对扩展分区进行再分区，可以使用

### 3. 分区和格式化

不同的操作系统具有不同的硬盘分区工具，Windows系统下非常有名的分区工具是fdisk，在Linux系统下进行分区可以使用fdisk，或者使用相同功能的图形界面程序。

每个主分区和逻辑分区都会被存储为一个识别文件系统的附加信息。

通过分区当然不能产生任何文件系统。分区必须要进行格式化。



## 9.1.2 使用fdisk进行硬盘分区

Linux系统使用fdisk命令能将磁盘划分成为若干个区，同时也能为每个分区指定分区的文件系统，比如ext3，ext4，FAT 32，SWAP，FAT 16以及其他类UNIX操作系统的文件系统等。

# 1. fdisk的介绍

使用fdisk命令可以对磁盘进行分区。

命令语法:

```
fdisk [-b <分区大小>][-uv][磁盘设备名]
```

```
fdisk [-l][-b <分区大小>][-uv][磁盘设备名]
```

```
fdisk [-s <分区编号>]
```

linux文件系统的分区号是83，swap  
的分区号是82

```
fdisk /dev/sda(b)  
进入fdisk命令界面
```

表9-1

fdisk命令参数

子命令	含 义
m	显示所有在fdisk中使用的命令
p	显示硬盘分区信息
a	设置硬盘启动区
n	创建新的分区
e	创建扩展分区
p	创建主分区
t	更改分区文件系统
d	删除硬盘分区
q	退出fdisk，不保存硬盘分区设置
w	保存硬盘分区设置并退出fdisk

## 2. Linux系统下硬盘分区举例

- (1) 进入fdisk界面，列出所有命令。
- (2) 显示硬盘分区信息。
- (3) 创建和删除主分区。
- (4) 创建扩展分区和逻辑驱动器。
- (5) 查看并转换文件系统。
- (6) 保存分区设置信息，并退出fdisk。
- (7) 在非交互式界面下显示当前硬盘的分区信息。

parted命令也可以进行分区

## 9.2 Linux文件系统简介

文件系统通过为每个文件分配文件块的方式把数据存储在存储设备中，这样就要维护每一个文件的文件块的分配信息，而分配信息本身也要存在磁盘上，不同的文件系统用不同的方法分配和读取文件块。

## 9.2.1 Linux文件系统的工作原理

有两种常用的文件系统的分配策略：块分配和扩展分配。块分配是当文件变大的时候每一次都为这个文件分配磁盘空间，而扩展分配则是当某个文件的磁盘空间不够的时候，一次性为它分配一连串连续的块。

传统的UNIX文件系统使用的块分配的机制提供了一个灵活而高效的文件块分配策略。

可以通过优化文件块的分配策略（尽可能为文件分配连续的块）来避免文件块的随机分配。

每一次当文件扩展的时候，块分配的算法就要写入一些关于新分配的块所在位置的信息。

## 9.2.2 Linux主流文件系统

文件系统是指文件在硬盘上的存储方法和排列顺序。在Linux系统中，每个分区都是一个文件系统，都有自己的目录层次结构。

Linux系统最重要特征之一就是支持多种文件系统，这样它更加灵活，并可以和许多其他操作系统共存。

虚拟文件系统使得Linux可以支持多个不同的文件系统。

随着Linux系统的不断发展，它所支持的文件格式系统也在迅速扩充。



下面介绍Linux系统最常用的几种文件系统。

1. ext3
2. ext4
3. JFS
4. ReiserFS
5. **XFS**

## 6. 其他文件系统

(1) Minix

(2) Xia

(3) IS09660

(4) NFS

(5) SysV

(6) VFAT

FAT32，外设设备常用，不支持4G以上的

## 9.2.3 查看Fedora 17支持的文件类型

不同版本的Linux上所支持的文件系统类型和种类都有所不同，下面以Fedora 17为例，介绍如何查看该系统所支持的文件系统。

以超级用户root登录Fedora 17系统，执行下面命令，显示目录内容。

```
[root@PC-LINUX ~]# ls -l /lib/modules/3.3.4-5.fc17.i686.PAE/kernel/fs
```

## 9.3 创建文件系统

如果要加载一个分区，首先需要确认文件系统的类型，然后才能挂载使用，比如通过mount加载或者通过修改/etc/fstab文件开机自动加载都可以实现该功能。

## 9.3.1 创建文件系统简介

对一个新的硬盘进行分区以后，还要对这些分区进行格式化并创建文件系统。一个分区只有建立了某种文件系统后，这个分区才能使用。建立文件系统的过程，就是用相应格式化工具格式化分区的过程，这个过程和在Windows系统中格式化某个分区为NTFS分区的过程类似。

Linux系统支持目前主流的文件系统，如VFAT，ext3，ext4，ReiserFS，ISO 9660和SWAP交换分区等。如果在计算机上新增加了一块硬盘，需要格式化成Linux的文件系统，最好选择ext3或ext4文件系统。

如果需要使用某个文件系统存放数据，一般要经过以下操作步骤。

- (1) 使用fdisk命令在硬盘上创建分区。
- (2) 使用mkfs命令在分区上创建文件系统。
- (3) 使用mount命令挂载文件系统，或修改/etc/fstab文件使得开机自动挂载文件系统。
- (4) 使用umount卸载文件系统。

## 9.3.2 创建文件系统

以Fedora 17系统为例，常用的创建文件系统的命令有`mkfs`，`mkfs.ext3`，`mkfs.ext2`，`mkfs.msdos`，`mkfs.vfat`和`mkswap`等。



# 1. 使用mkfs命令创建文件系统

使用mkfs命令可以在分区上建立各种文件系统。

命令语法：

```
mkfs -t [文件系统类型][磁盘设备名]
```

## 【例9.1】 使用mkfs命令创建文件系统。

(1) 查看当前磁盘上的分区情况，该磁盘设备是/dev/sda。

```
[root@PC-LINUX ~]# fdisk -l /dev/sda
Disk /dev/sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0001c201
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	2048	206847	102400	83	Linux
/dev/sda2		206848	31664127	15728640	83	Linux
/dev/sda3		31664128	35858431	2097152	82	Linux swap / Solaris
/dev/sda4		35858432	41943039	3042304	5	Extended
/dev/sda5		35860480	35880959	10240	83	Linux
/dev/sda6		35883008	41943039	3030016	c	W95 FAT32 (LBA)

```
[root@PC-LINUX ~]# ls /dev/sda*  
/dev/sda /dev/sda1 /dev/sda2 /dev/sda3
```

```
[root@PC-LINUX ~]# partprobe
```

Warning: 无法以读写方式打开 /dev/sr0 (只读文件系统)。/dev/sr0 已按照只读方式打开。

Warning: 无法以读写方式打开 /dev/sr0 (只读文件系统)。/dev/sr0 已按照只读方式打开。

Error: 无效的分区表 - /dev/sr0 出现递归分区。

//使用fdisk工具只是将分区信息写到磁盘，如果需要mkfs创建文件系统则需要事先重启系统，而使用partprobe则可以使kernel重新读取分区信息，从而避免重启系统。

```
[root@PC-LINUX ~]# ls /dev/sda*  
/dev/sda /dev/sda1 /dev/sda2 /dev/sda3 /dev/sda4 /dev/sda5 /dev/sda6
```

(2) 格式化/dev/sda5分区，创建ext4文件系统。  
对计算机硬盘分好分区之后重启计算机，使用如下命令创建文件系统。

```
[root@PC-LINUX ~]# mkfs -t ext4 /dev/sda5  
mke2fs 1.42 (29-Nov-2011)
```

文件系统标签=

OS type: Linux

块大小=1024 (log=0)

分块大小=1024 (log=0)

Stride=0 blocks, Stripe width=0 blocks

2560 inodes, 10240 blocks

512 blocks (5.00%) reserved for the super user

第一个数据块=1

Maximum filesystem blocks=10485760

2 block groups

8192 blocks per group, 8192 fragments per group

1280 inodes per group

Superblock backups stored on blocks:

8193

Allocating group tables: 完成

正在写入inode表: 完成

Creating journal (1024 blocks): 完成

Writing superblocks and filesystem accounting information: 完成

当然也可以把分区格式化成其他的文件系统。比如可以把/dev/sda5格式化为ext3、reiserfs、vfat或msdos，命令格式如下：

```
[root@PC-LINUX ~]# mkfs -t ext3 /dev/sda5
```

```
[root@PC-LINUX ~]# mkfs -t reiserfs /dev/sda5
```

```
[root@PC-LINUX ~]# mkfs -t vfat /dev/sda5
```

```
[root@PC-LINUX ~]# mkfs -t msdos /dev/sda5
```

## 2. 使用其他命令创建文件系统

在这里主要介绍mkfs.ext4, mkfs.ext3, mke2fs, mkdosfs, mkfs.msdos和mkfs.vfat命令, 使用mkfs命令在执行的时候, 也是调用的这些命令。

【例9.2】 使用mkfs.ext4命令将/dev/sda5设备格式化成ext4文件系统。

```
[root@PC-LINUX ~]# mkfs.ext4 /dev/sda5
mke2fs 1.42 (29-Nov-2011)
文件系统标签=
OS type: Linux
块大小=1024 (log=0)
分块大小=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
2560 inodes, 10240 blocks
512 blocks (5.00%) reserved for the super user
第一个数据块=1
Maximum filesystem blocks=10485760
2 block groups
8192 blocks per group, 8192 fragments per group
1280 inodes per group
Superblock backups stored on blocks:
    8193
```



Allocating group tables: 完成

正在写入inode表: 完成

Creating journal (1024 blocks): 完成

Writing superblocks and filesystem accounting information: 完成



其他创建文件系统的命令和mkfs.ext3类似，如下所示：

//将分区/dev/sda5格式化成ext3文件系统

```
[root@PC-LINUX ~]# mkfs.ext3 /dev/sda5
```

//将分区/dev/sda5格式化成ext2文件系统

```
[root@PC-LINUX ~]# mke2fs /dev/sda5
```

//将分区/dev/sda5格式化成FAT32文件系统

```
[root@PC-LINUX ~]# mkfs.vfat /dev/sda5
```

//将分区/dev/sda5格式化成FAT16文件系统，msdos文件系统就是FAT16

```
[root@PC-LINUX ~]# mkfs.msdos /dev/sda5
```

//将分区/dev/sda5格式化成FAT16文件系统

```
[root@PC-LINUX ~]# mkdosfs /dev/sda5
```

## 9.4 挂载和卸载文件系统

使用mount和umount命令可以实现挂载和卸载功能，这样用户才可以使用相应的设备存储数据。

## 9.4.1 挂载文件系统

使用mount命令可以将某个分区、光盘、软盘或是U盘挂载到Linux系统的目录下。

命令语法：

mount [选项][设备名称][挂载点]

# 1. 挂载硬盘

【例9.3】 挂载分区/dev/sda5到/mnt/kk目录中。

```
[root@PC-LINUX ~]# mkdir /mnt/kk
```

```
[root@PC-LINUX ~]# mount /dev/sda5 /mnt/kk
```

//创建需要放置文件系统的目录/mnt/kk，然后将/dev/sda5挂载到该目录中

```
[root@PC-LINUX ~]# ls /mnt/kk
```

lost+found

//查看目录/mnt/kk，其中没有任何的文件

```
[root@PC-LINUX ~]# touch /mnt/kk/abc
```

//在目录/mnt/kk中创建空文件abc

```
[root@PC-LINUX ~]# ls /mnt/kk
```

abc lost+found

//再次查看目录/mnt/kk的内容，可以看到刚才创建的文件，其实它是存在于/dev/sda5分区上

```
[root@PC-LINUX ~]# df
```

文件系统	1K-块	已用	可用	已用%	挂载点
rootfs	15674684	9442696	5445556	64%	/
devtmpfs	504952	0	504952	0%	/dev
tmpfs	513924	84	513840	1%	/dev/shm
tmpfs	513924	2136	511788	1%	/run
/dev/sda2	15674684	9442696	5445556	64%	/
tmpfs	513924	0	513924	0%	/sys/fs/cgroup
tmpfs	513924	0	513924	0%	/media
/dev/sda1	100604	43743	51741	46%	/boot
/dev/sr0	3820164	3820164	0	100%	/run/media/root/Fedora 17 i386
/dev/sda5	10084	1290	8282	14%	/mnt/kk

【例9.4】 以只读方式挂载/dev/sda5分区到/mnt/kk目录中。

```
[root@PC-LINUX ~]# mount -o ro /dev/sda5 /mnt/kk
```

//以只读方式挂载/dev/sda5分区

```
[root@PC-LINUX ~]# mkdir /mnt/kk/a
```

Mkdir: 无法创建目录“/mnt/kk/a”: 只读文件系统

//在目录/mnt/kk中创建目录a，无法创建，因为它是只读的

## 2. 挂载光盘、软盘、U盘

Linux系统在使用光盘、软盘、U盘以及移动硬盘时，必须先执行挂载命令。挂载命令会将这些存储介质指定成系统中的某个目录，以后直接访问相应目录即可读写存储介质上的数据。

【例9.5】 将光盘放入光驱，挂载光盘到/mnt/cdrom目录中。

```
[root@PC-LINUX ~]# mkdir /mnt/cdrom
```

```
[root@PC-LINUX ~]# mount -t iso9660 /dev/cdrom /mnt/cdrom
```

```
mount: block device /dev/sr0 is write-protected, mounting read-only
```

//创建挂载目录/mnt/cdrom，并将光盘挂载到该目录中

```
[root@PC-LINUX ~]# ls /mnt/cdrom
```

```
images isolinux LiveOS Packages repodata TRANS.TBL
```

//可以在光盘挂载目录/mnt/cdrom中看到光盘的内容

【例9.6】 将软盘放入软驱，挂载软盘到/mnt/floppy目录中。

```
[root@PC-LINUX ~]# mkdir /mnt/floppy
```

```
[root@PC-LINUX ~]# mount -t vfat /dev/fd0 /mnt/floppy
```

//创建挂载目录/mnt/floppy，并将软盘挂载到该目录中



【例9.7】 挂载文件系统是FAT32格式的U盘。

```
[root@PC-LINUX ~]# fdisk -l /dev/sdb
Disk /dev/sdb: 1014 MB, 1014874624 bytes
32 heads, 61 sectors/track, 1015 cylinders
Units = cylinders of 1952 * 512 = 999424 bytes
Disk identifier: 0x69737369
```

This doesn't look like a partition table  
Probably you selected the wrong device.

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1	?	957875	1044294	84344761	69	Unknown

Partition 1 has different physical/logical beginnings (non-Linux?):

phys=(68, 13, 10) logical=(957874, 21, 37)

Partition 1 has different physical/logical endings:

phys=(288, 115, 43) logical=(1044293, 15, 36)

Partition 1 does not end on cylinder boundary.

/dev/sdb2	?	871681	1829612	934940732+	73	Unknown
-----------	---	--------	---------	------------	----	---------

Partition 2 has different physical/logical beginnings (non-Linux?):

phys=(371, 114, 37) logical=(871680, 1, 61)

Partition 2 has different physical/logical endings:

phys=(366, 32, 33) logical=(1829611, 4, 30)

Partition 2 does not end on cylinder boundary.

/dev/sdb3	?	2	2	0	74	Unknown
-----------	---	---	---	---	----	---------

Partition 3 has different physical/logical beginnings (non-Linux?):

phys=(371, 114, 37) logical=(1, 10, 12)

Partition 3 has different physical/logical endings:

phys=(372, 97, 50) logical=(1, 10, 11)

Partition 3 does not end on cylinder boundary.

/dev/sdb4		1	1759792	1717556736	0	Empty
-----------	--	---	---------	------------	---	-------

Partition 4 has different physical/logical beginnings (non-Linux?):

phys=(0, 0, 0) logical=(0, 0, 1)

Partition 4 has different physical/logical endings:

phys=(0, 0, 0) logical=(1759791, 23, 37)

Partition 4 does not end on cylinder boundary.

Partition table entries are not in disk order

//使用fdisk命令查看外置U盘的设备号，默认为/dev/sda，如果已经存在SCSI硬盘，那么U盘设备为/dev/sdb，本例U盘设备为/dev/sdb

```
[root@PC-LINUX ~]# mkdir /mnt/disk
```

```
[root@PC-LINUX ~]# mount -t vfat /dev/sdb /mnt/disk
```

//创建U盘挂载目录为/mnt/disk，并将U盘挂载到该目录中

## 9.4.2 卸载文件系统

使用umount命令可以将某个分区、光盘、软盘或是U盘进行卸载。

命令语法：

```
umount [选项] [-t <文件系统类型>][文件系统]
```

# 1. 卸载硬盘

【例9.8】 卸载分区/dev/sda5文件系统。

```
[root@PC-LINUX ~]# umount /dev/sda5
```

```
[root@PC-LINUX ~]# df
```

文件系统	1K-块	已用	可用	已用%	挂载点
rootfs	15674684	9442696	5445556	64%	/
devtmpfs	504952	0	504952	0%	/dev
tmpfs	513924	84	513840	1%	/dev/shm
tmpfs	513924	2136	511788	1%	/run
/dev/sda2	15674684	9442696	5445556	64%	/
tmpfs	513924	0	513924	0%	/sys/fs/cgroup
tmpfs	513924	0	513924	0%	/media
/dev/sda1	100604	43743	51741	46%	/boot
/dev/sr0	3820164	3820164	0	100%	/run/media/root/Fedora 17 i386

//使用df命令可以看到分区/dev/sda5已经卸载

## 2. 卸载光盘、软盘、U盘

【例9.9】 卸载光盘。

```
[root@PC-LINUX ~]# umount /dev/cdrom
```

```
[root@PC-LINUX ~]# umount /mnt/cdrom
```

//可以通过以上任何一种方式卸载光盘

【例9.10】 卸载软盘。

```
[root@PC-LINUX ~]# umount /dev/fd0
```

```
[root@PC-LINUX ~]# umount /mnt/floppy
```

//可以通过以上任何一种方式卸载软盘

【例9.11】 卸载U盘

```
[root@PC-LINUX ~]# umonuu /mnt/disk
```

```
[root@PC-LINUX ~]# umount /dev/sdb
```

//可以通过以上任何一种方式卸载U盘

## 9.4.3 查看分区挂载情况

要查看Linux系统分区挂载情况，除了可以使用df命令之外，还可以通过mount -s命令以及查看/etc/mtab文件获取信息。

# 1. 使用 *mount* -s 命令

【例9.12】 使用mount命令查看分区挂载情况。

可以用df -h

```
[root@PC-LINUX ~]# mount -s
```

```
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
```

```
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
```

```
devtmpfs on /dev type devtmpfs
```

```
(rw,nosuid,seclabel,size=504952k,nr_inodes=126238,mode=755)
```

```
devpts on /dev/pts type devpts
```

```
(rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000)
```

```
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel)
```

```
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,mode=755)
```

```
/dev/sda2 on / type ext4 (rw,relatime,seclabel,user_xattr,barrier=1,data=ordered)
```

```
selinuxfs on /sys/fs/selinux type selinuxfs (rw,relatime)
```

```
tmpfs on /sys/fs/cgroup type tmpfs (rw,nosuid,nodev,noexec,relatime,seclabel,mode=755)
```

```
cgroup on /sys/fs/cgroup/systemd type cgroup
```

```
(rw,nosuid,nodev,noexec,relatime,release_agent=/usr/lib/systemd/systemd-cgroups-agent,name=systemd)
```



cgroup on /sys/fs/cgroup/cpuset type cgroup  
(rw,nosuid,nodev,noexec,relatime,cpuset)  
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup  
(rw,nosuid,nodev,noexec,relatime,cpuacct,cpu)  
cgroup on /sys/fs/cgroup/memory type cgroup  
(rw,nosuid,nodev,noexec,relatime,memory)  
cgroup on /sys/fs/cgroup/devices type cgroup  
(rw,nosuid,nodev,noexec,relatime,devices)  
cgroup on /sys/fs/cgroup/freezer type cgroup  
(rw,nosuid,nodev,noexec,relatime,freezer)  
cgroup on /sys/fs/cgroup/net\_cls type cgroup  
(rw,nosuid,nodev,noexec,relatime,net\_cls)  
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)  
cgroup on /sys/fs/cgroup/perf\_event type cgroup  
(rw,nosuid,nodev,noexec,relatime,perf\_event)  
systemd-1 on /proc/sys/fs/binfmt\_misc type autofs  
(rw,relatime,fd=22,pgrp=1,timeout=300,minproto=5,maxproto=5,direct)

tmpfs on /media type tmpfs (rw,nosuid,nodev,noexec,relatime,seclabel,mode=755)  
mqueue on /dev/mqueue type mqueue (rw,relatime,seclabel)  
debugfs on /sys/kernel/debug type debugfs (rw,relatime)  
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel)  
configfs on /sys/kernel/config type configfs (rw,relatime)  
securityfs on /sys/kernel/security type securityfs (rw,relatime)  
/dev/sda1 on /boot type ext4  
(rw,relatime,seclabel,user\_xattr,barrier=1,data=ordered)  
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)  
/dev/sr0 on /run/media/root/Fedora 17 i386 type iso9660  
(ro,nosuid,nodev,relatime,uid=0,gid=0,ioccharset=utf8,mode=0400,dmode=0500,u  
helper=udisks2)  
gvfs-fuse-daemon on /run/user/root/gvfs type fuse.gvfs-fuse-daemon  
(rw,nosuid,nodev,relatime,user\_id=0,group\_id=0)  
/dev/sda5 on /mnt/kk type ext4  
(rw,relatime,seclabel,user\_xattr,barrier=1,data=ordered)

## 2. 查看/etc/mtab文件

【例9.13】 通过/etc/mtab文件查看分区挂载情况。

```
[root@PC-LINUX ~]# cat /etc/mtab
```

```
rootfs / rootfs rw 0 0
```

```
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
```

```
sysfs /sys sysfs rw,seclabel,nosuid,nodev,noexec,relatime 0 0
```

```
devtmpfs /dev devtmpfs
```

```
rw,seclabel,nosuid,size=504952k,nr_inodes=126238,mode=755 0 0
```

```
devpts /dev/pts devpts
```

```
rw,seclabel,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0
```

```
tmpfs /dev/shm tmpfs rw,seclabel,nosuid,nodev 0 0
```

```
tmpfs /run tmpfs rw,seclabel,nosuid,nodev,mode=755 0 0
```

```
/dev/sda2 / ext4 rw,seclabel,relatime,user_xattr,barrier=1,data=ordered 0 0
```

```
selinuxfs /sys/fs/selinux selinuxfs rw,relatime 0 0
```

```
tmpfs /sys/fs/cgroup tmpfs rw,seclabel,nosuid,nodev,noexec,relatime,mode=755 0 0
```

```
cgroup /sys/fs/cgroup/systemd cgroup
```

```
rw,nosuid,nodev,noexec,relatime,release_agent=/usr/lib/systemd/systemd-cgroups-agent,name=systemd 0 0
```

```
cgroup /sys/fs/cgroup/cpuset cgroup rw,nosuid,nodev,noexec,relatime,cpuset 0 0
```

```
cgroup /sys/fs/cgroup/cpu,cpuacct cgroup
```

rw,nosuid,nodev,noexec,relatime,cpuacct,cpu 0 0  
cgroup /sys/fs/cgroup/memory cgroup rw,nosuid,nodev,noexec,relatime,memory 0 0  
cgroup /sys/fs/cgroup/devices cgroup rw,nosuid,nodev,noexec,relatime,devices 0 0  
cgroup /sys/fs/cgroup/freezer cgroup rw,nosuid,nodev,noexec,relatime,freezer 0 0  
cgroup /sys/fs/cgroup/net\_cls cgroup rw,nosuid,nodev,noexec,relatime,net\_cls 0 0  
cgroup /sys/fs/cgroup/blkio cgroup rw,nosuid,nodev,noexec,relatime,blkio 0 0  
cgroup /sys/fs/cgroup/perf\_event cgroup rw,nosuid,nodev,noexec,relatime,perf\_event 0 0  
systemd-1 /proc/sys/fs/binfmt\_misc autofs  
rw,relatime,fd=22,pgrp=1,timeout=300,minproto=5,maxproto=5,direct 0 0  
tmpfs /media tmpfs rw,seclabel,nosuid,nodev,noexec,relatime,mode=755 0 0  
mqueue /dev/mqueue mqueue rw,seclabel,relatime 0 0  
debugfs /sys/kernel/debug debugfs rw,relatime 0 0  
hugetlbfs /dev/hugepages hugetlbfs rw,seclabel,relatime 0 0  
configfs /sys/kernel/config configfs rw,relatime 0 0  
securityfs /sys/kernel/security securityfs rw,relatime 0 0  
/dev/sda1 /boot ext4 rw,seclabel,relatime,user\_xattr,barrier=1,data=ordered 0 0  
fusectl /sys/fs/fuse/connections fusectl rw,relatime 0 0  
/dev/sr0 /run/media/root/Fedora\04017\040i386 iso9660  
ro,nosuid,nodev,relatime,uid=0,gid=0,iocharset=utf8,mode=0400,dmode=0500 0 0  
gvfs-fuse-daemon /run/user/root/gvfs fuse.gvfs-fuse-daemon  
rw,nosuid,nodev,relatime,user\_id=0,group\_id=0 0 0  
/dev/sda5 /mnt/kk ext4 rw,seclabel,relatime,user\_xattr,barrier=1,data=ordered 0 0

## 9.5 设置开机自动挂载文件系统

将某个分区或是设备挂载了以后才能使用，但是当计算机重新启动以后，又需要重新挂载，这个时候可以通过修改/etc/fstab文件实现开机自动挂载文件系统。

## 9.5.1 /etc/fstab文件简介

/etc/fstab文件是一个配置文件，它包含了所有分区以及存储设备的信息。其中包含了磁盘分区和存储设备如何挂载，以及挂载在什么地方的信息。

如果在Linux系统中不能访问Windows的分区，或者作为一名普通用户，不能挂载光驱和向软盘中写入数据，或者在管理CD-RW的过程中遇到了问题，就有可能是错误地配置了/etc/fstab文件，通常可以通过编辑/etc/fstab这个文件来解决前面提到的问题。

`/etc/fstab`文件是一个简单的文本文件，可以用任何文本编辑器去编辑它，但是必须要以root用户登录才可以编辑该文件。



## 9.5.2 /etc/fstab文件详解

由于每台计算机系统的分区和设备属性不同，所以/etc/fstab文件也不一样，但是基本的结构总是相似的。下面就是其中的一行：

```
/dev/sda5 /mnt/kk ext4 defaults 0 0
```

每1行包含着1个设备或分区的信息，每1行又有多个列的信息。第1列是设备名或者设备UUID号，第2列是它的挂载点，第3列是它的文件系统格式，第4列是挂载参数，第5列是转储选项，第6列是文件系统检查选项。

通过名称挂载，通过UUID挂载，通过卷标挂载

## 1. 设备

使用设备名称或设备UUID号表示都可以。  
如果要查看某设备的UUID号，可以用下面命令：

```
ls -l /dev/disk/by-uuid 或 blkid
```

## 2. 挂载点

Linux系统为每个设备或分区设定了挂载目录。

## 3. 文件系统格式

Linux系统为每个设备或分区指定了文件系统格式。

## 4. 挂载参数

- (1) auto和noauto
- (2) user和nouser
- (3) exec和noexec
- (4) ro
- (5) rw
- (6) sync和async
- (7) defaults

## 5. 转储选项

dump选项检查文件系统并用一个数字来决定该文件系统是否需要备份。如果它是0，dump将会忽略该文件系统，不做备份。

## 6. 文件系统检查选项

fsck选项通过检验第6项中的数字来决定以何种顺序检查文件系统，如果它是0，fsck将不检查该文件系统。根文件系统（“/”）的默认值为1，其他文件系统可以为2-9。

## 9.6

## 使用交换空间

Linux系统中的交换空间在物理内存被用完时使用。如果系统需要更多的内存资源，而物理内存已经用完，内存中不活跃的页就会被转移到交换空间中。虽然交换空间可以为具有少量内存的计算机提供帮助，但是这种方法不应该被当作是对内存的取代。交换空间位于硬盘驱动器上，它比进入物理内存要慢。

## 9.6.1 添加交换空间

有时，用户需要在安装Linux系统后添加更多的交换空间。用户可以添加一个交换分区或添加一个交换文件，推荐使用添加一个交换分区。交换空间可以是一个专用的交换分区，也可以是一个交换文件，或是两者的结合。交换空间总的大小至少为计算机内存的1~2倍。

# 1. 添加交换分区

## 【例9.14】 添加交换分区。

### (1) 创建交换分区。

使用fdisk命令已经创建好分区/dev/sda5，假设将分区/dev/sda5创建为交换（SWAP）分区，在Shell提示下以root用户身份输入以下命令。

```
[root@PC-LINUX ~]# mkswap /dev/sda5
```

```
Setting up swapspace version 1, size = 10236 KiB
```

```
no label, UUID=8406e55c-ca76-44a0-b531-62836e208cfb
```

```
//将/dev/sda5分区创建为交换分区
```

```
[root@PC-LINUX ~]# free
```

	total	used	free	shared	buffers	cached
Mem:	1027848	941104	86744	0	73424	556208
-/+ buffers/cache:	311472	716376				
Swap:	2097148	588	2096560			

```
//因为当前还没有启动交换分区，所以使用free命令无法看到
```

## (2) 启用交换分区。

输入以下命令启用交换分区/dev/sda5。

```
[root@PC-LINUX ~]# swapon /dev/sda5
```

```
//启动交换分区/dev/sda5
```

```
[root@PC-LINUX ~]# free
```

	total	used	free	shared	buffers	cached
Mem:	1027848	941104	86744		0	73436
-/+ buffers/cache:		311448	716400			
Swap:	2107384	588	2106796			

//因为当前已经启动了交换分区/dev/sda5所以总的交换分区容量已经增加了



### （3）确认已经启用交换分区。

创建并启用了交换分区之后，使用如下命令查看交换分区是否已启用。

```
[root@PC-LINUX ~]# cat /proc/swaps
```

Filename	Type	Size	Used	Priority
/dev/sda3	partition	2097148	588	0
/dev/sda5	partition	10236	0	-1

//可以看到当前计算机的SWAP分区由/dev/sda3和/dev/sda5这两个分区构成

（4）如果要在系统引导时启用交换分区，编辑/etc/fstab文件添加如下内容。然后在系统下次引导时，就会启用新建的交换分区。

/dev/sda5	swap	swap	defaults	0	0
-----------	------	------	----------	---	---

## 2. 添加交换文件

【例9.15】 添加交换文件。

(1) 创建文件/swapfile。

将大小乘以1024来判定块的大小。例如，大小为68MB的交换文件的块大小为65536。

在Shell提示下以root用户身份输入以下命令，其中的count等于想要输入的块大小。

```
[root@PC-LINUX ~]# dd if=/dev/zero of=/swapfile bs=1024 count=66536
```

记录了66536+0 的读入

记录了66536+0 的写出

68132864字节(68 MB)已复制，0.447235 秒，152 MB/秒

```
[root@PC-LINUX ~]# ls /
```

```
bin  dev  home  lost+found  mnt  proc  run  srv      sys  usr  
boot  etc  lib   media      opt  root  sbin  swapfile  tmp  var
```

//在“/”目录下可以看到文件swapfile

## （2）创建交换文件。

使用以下命令创建交换文件。

```
[root@PC-LINUX ~]# mkswap /swapfile  
Setting up swapspace version 1, size = 66532 KiB  
no label, UUID=1587ac78-d291-4fac-ab1e-ac89a7d9c658
```

## （3）启用交换文件。

使用如下命令启用交换文件。

```
[root@PC-LINUX ~]# swapon /swapfile
```

（4）新添了交换分区并启用它之后，使用如下命令确保交换文件已被启用了。

```
[root@PC-LINUX ~]# free
```

	total	used	free	shared	buffers	cached
Mem:	1027848	943240	84608		0	73552
-/+ buffers/cache:		314164	713684			
Swap:	2163680	604	2163076			

```
[root@PC-LINUX ~]# cat /proc/swaps
```

Filename	Type	Size	Used	Priority
/dev/sda3	partition	2097148	604	0
/swapfile	file	66532	0	-1

（5）如果要在系统引导时启用交换文件，编辑**/etc/fstab**文件添加如下内容。然后在系统下次引导时，就会启用新建的交换文件。

/swapfile	swap	swap	defaults	0	0
-----------	------	------	----------	---	---

## 9.6.2 删除交换空间

当某个交换分区或交换文件不再需要时，可以使用如下步骤将其删除。

# 1. 删除交换分区

【例9.16】 删除交换分区。

(1) 在Shell提示下以root用户身份输入以下命令禁用交换分区（这里的/dev/sdb5是交换分区）。

```
[root@PC-LINUX ~]# swapoff /dev/sda5
```

```
[root@PC-LINUX ~]# free
```

	total	used	free	shared	buffers	cached
Mem:	1027848	873644	154204		0	73572
-/+ buffers/cache:		311084	716764			
Swap:	2097148	604	2096544			

(2) 如果要在系统引导时不启用交换分区，编辑/etc/fstab文件删除如下内容。然后在系统下次引导时，就不会启用交换分区。

/dev/sda5	swap	swap	defaults	0	0
-----------	------	------	----------	---	---

## 2. 删除交换文件

【例9.17】 删除交换文件。

(1) 在Shell提示下以root用户身份执行以下命令来禁用交换文件（这里的/swapfile是交换文件）。

```
[root@PC-LINUX ~]# swapoff /swapfile
```

```
[root@PC-LINUX ~]# free
```

	total	used	free	shared	buffers	cached
Mem:	1027848	943268	84580		0	73552
-/+ buffers/cache:		314192	713656			
Swap:	2097148	604	2096544			

(2) 删除/swapfile文件。

输入以下命令删除/swapfile文件。

```
[root@PC-LINUX ~]# rm -rf /swapfile
```

(3) 如果要在系统引导时不启用交换文件，编辑/etc/fstab文件删除如下内容。然后在系统下次引导时，就不会启用交换文件。

/swapfile	swap	swap	defaults	0	0
-----------	------	------	----------	---	---

## 9.7

## 权限设置

为了有效地控制用户对文件和目录的访问，必须对其设置权限以实现安全控制，本节主要讲述如何设置权限以及更改文件和目录的所有权。



## 9.7.1 文件和目录权限

在Linux系统中，用户可以对每一个文件或目录都具有访问权限，这些访问权限决定了谁能访问，以及如何访问这些文件和目录。

# 1. 文件权限简介

通过设定权限可以限制或允许以下3种用户访问：文件的所有者（文件属主）、文件所有者所在组的同组用户（同组用户）、系统中的其他用户。

在Linux系统中，每种用户都有对文件或目录的读取、写入和执行权限。

# Linux中，每种用户都有三种权限

权限	对文件的影响	对目录的影响
r（读取）	可读取文件内容	可浏览目录
w（写入）	可修改文件内容	可删除、移动目录内文件
x（执行）	可执行文件	可进入目录

目录必须拥有x权限，否则无法进入目录

## 2. 一般权限

用“ls -l”命令可以显示文件的详细信息，其中包括权限，如下所示：

```
[root@PC-LINUX ~]# ls -l /root  
total 272
```

```
-rw-----. 1 root root 1816 Dec  7  2012 /root/anaconda-ks.cfg  
-rw-r--r--. 1 root root 44934 Dec 11  2012 /root/install.log  
-rw-r--r--. 1 root root 10151 Jul  2  2012 /root/install.log.syslog
```

<https://blog.csdn.net/yexiangCSDN/article/details/81011290>

ls命令

-rw-r--r--. 1 root root 44934 Dec 11 2012 install.log

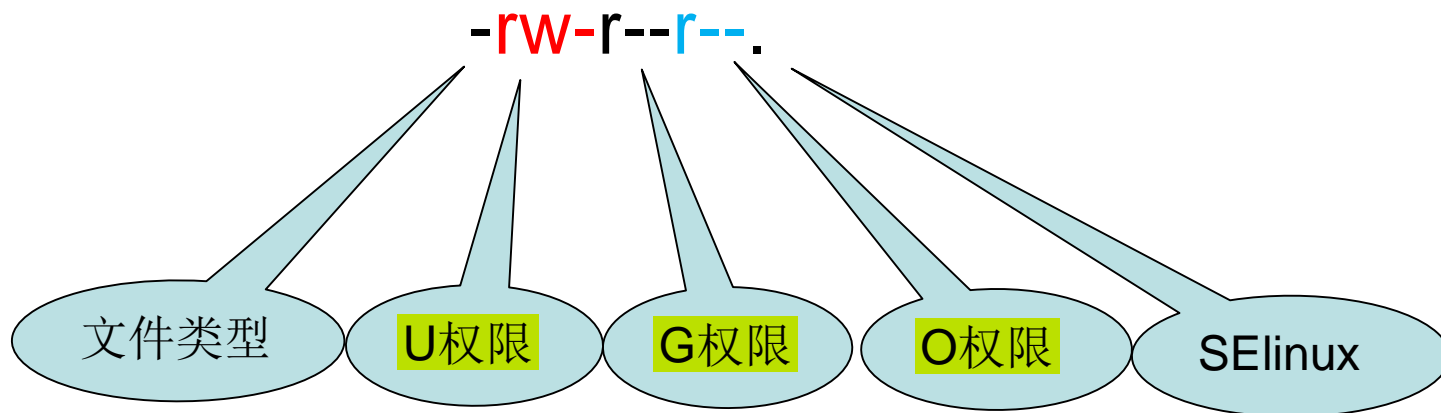
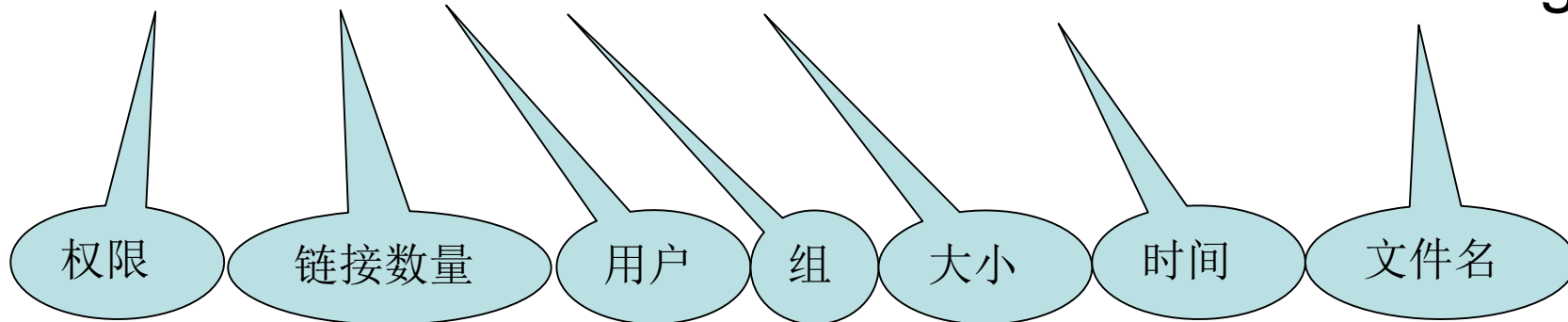




表9-2                      文件类型

代 表 字 符	文 件 类 型
d	表示目录文件，目录是一个特殊的文件
—	表示普通的文件
l	表示链接文件，实际上它指向另一个文件
b	表示块设备文件
c	表示字符设备文件
p	表示管道文件

### 3. 特殊权限

除了一般权限以外，还有所谓的特殊权限。用户若无特殊需求，不要启用这些权限，避免出现安全漏洞。

权限	对文件的影响	对目录的影响
SUID	以文件的所属用户身份执行，而非执行文件的用户	无
SGID	以文件所属组身份执行	在该目录中创建的任意新文件的所属组与该目录的所属组相同
Sticky	无	对目录拥有写入权限的用户仅可以删除其拥有的文件，无法删除其他用户所属的文件

## 9.7.2 权限设置

只有系统管理员和文件的所有者才可以更改文件或目录的权限，更改文件或目录权限的方法一般有3种。



# 1. 文件管理器更改权限



图9-1 设置文件权限

## 2. 文字设定法

通过文字设定法更改权限需要使用chmod命令，chmod的命令格式如下：

```
chmod [ugoa][+ -=][rwx][文件或目录名]
```

u表示该文件的所有者，g表示与该文件的所有者属于同一个组的用户，o表示其他用户，a表示以上三者；

+表示增加指定权限，-表示取消指定权限，=表示设定权限等于指定权限；

r表示可读取，w表示可写入，x表示表示文件可执行或目录可进入。

通过文字设定法更改特殊权限需要使用  
chmod命令，chmod的命令格式如下：

权限	命令	模式
SUID	chmod u+s	s=x+SUID S=--+SUID
SGID	chmod g+s	s=x+SGID S=--+SGID
Sticky	chmod o+t	t=x+Sticky T=--+Sticky

【例9.18】 添加所有者对a文件的写入权限。

```
[root@PC-LINUX ~]# ls -l a
-r--r--r--. 1 root root 6 6月  3 09:33 a
//可以看到a文件现在所有者的权限是读取
[root@PC-LINUX ~]# chmod u+w a
[root@PC-LINUX ~]# ls -l a
-rw-r--r--. 1 root root 6 6月  3 09:33 a
//更改权限以后，所有者对a文件多了写入的权限
```

【例9.19】 取消所有者对a文件的读取权限。

```
[root@PC-LINUX ~]# chmod u-r a
```

```
[root@PC-LINUX ~]# ls -l a
```

```
--w-r--r--. 1 root root 6 6月  3 09:33 a
```

//查看文件权限，可以看到文件a的所有者权限已经没有读取了

【例9.20】 重新分配同组用户对a文件有写入的权限。

```
[root@PC-LINUX ~]# chmod g=w a
```

```
[root@PC-LINUX ~]# ls -l a
```

```
--w--w-r--. 1 root root 6 6月  3 09:33 a
```

//可以看到，同组用户原先的权限没有了，现在重新分配的是写入权限

【例9.21】更改a文件权限，添加所有者为读取、写入权限，同组用户为读取权限，其他用户读取、写入和执行的权限。

```
[root@PC-LINUX ~]# chmod u+rw,g+r,o+rw a  
[root@PC-LINUX ~]# ls -l a  
-rw-rw-rwx. 1 root root 6 6月 3 09:33 a
```

【例9.22】取消所有用户的读取、写入和执行权限。

```
[root@PC-LINUX ~]# chmod a-rwx a  
[root@PC-LINUX ~]# ls -l a  
-----. 1 root root 6 6月 3 09:33 a
```

【例9.23】 添加a文件的特殊权限为SUID。

```
[root@PC-LINUX ~]# ls -l a
-----. 1 root root 6 6月  3 09:33 a
[root@PC-LINUX ~]# chmod u+s a
[root@PC-LINUX ~]# ls -l a
---S-----. 1 root root 6 6月  3 09:33 a
```

【例9.24】 添加a文件的特殊权限为SGID。

```
[root@PC-LINUX ~]# chmod g+s a
[root@PC-LINUX ~]# ls -l a
---S--S---. 1 root root 6 6月  3 09:33 a
```

【例9.25】 添加a文件的特殊权限为Sticky。

```
[root@PC-LINUX ~]# chmod o+t a
[root@PC-LINUX ~]# ls -l a
---S--S--T. 1 root root 6 6月  3 09:33 a
```

### 3. 数字设定法

文件和目录的权限表中用r, w及x这3个字符来为所有者、同组用户和其他用户设置权限。有时候, 字符似乎过于麻烦, 因此还有另外一种方法是以数字来表示权限, 而且仅需3个数字。



使用数字设定法更改文件权限，首先必须了解数字表示的含义：0表示没有权限，1表示可执行权限，2表示写入权限，4表示读取权限，然后将其相加。所以数字属性的格式应为3个0~7的8进制数，其顺序是（u），（g），（o）。

- (1)  $r$ : 对应数值4。
- (2)  $w$ : 对应数值2。
- (3)  $x$ : 对应数值1。
- (4)  $-$ : 对应数值0。

使用数字设定法更改文件权限，chmod的命令格式如下：

`chmod [n1n2n3] [文件或目录名]`

【例9.26】 设置a文件权限，所有者拥有读取、写入和执行的权限。

```
[root@PC-LINUX ~]# ls -l a
-r--r--r--. 1 root root 6 6月  3 09:33 a
[root@PC-LINUX ~]# chmod 700 a
[root@PC-LINUX ~]# ls -l a
-rwx-----. 1 root root 6 6月  3 09:33 a
```

【例9.27】 设置a文件权限，所有者拥有读取，同组用户有读取、写入和执行的权限。

```
[root@PC-LINUX ~]# chmod 470 a
[root@PC-LINUX ~]# ls -l a
-r--rwx---. 1 root root 6 6月  3 09:33 a
```

【例9.28】 设置a文件权限，所有用户拥有读取、写入和执行的权限。

```
[root@PC-LINUX ~]# chmod 777 a
[root@PC-LINUX ~]# ls -l a
-rwxrwxrwx. 1 root root 6 6月  3 09:33 a
```

【例9.29】 设置a文件权限，其他用户拥有读取、写入和执行的权限。

```
[root@PC-LINUX ~]# chmod 7 a
[root@PC-LINUX ~]# ls -l a
-----rwx. 1 root root 6 6月  3 09:33 a
//在这里和用chmod 007 a是一样的效果
```

如果要加上特殊权限，就必须使用4位数字才能表示。特殊权限的对应数值如下。

- (1) SUID: 对应数值4。
- (2) SGID: 对应数值2。
- (3) Sticky: 对应数值1。

【例9.30】 设置文件a具有SUID权限。

```
[root@PC-LINUX ~]# ls -l a
-r--r--r--. 1 root root 6 6月  3 09:33 a
[root@PC-LINUX ~]# chmod 4000 a
[root@PC-LINUX ~]# ls -l a
---S-----. 1 root root 6 6月  3 09:33 a
```

【例9.31】 设置文件a具有SGID权限。

```
[root@PC-LINUX ~]# chmod 2000 a
[root@PC-LINUX ~]# ls -l a
-----S---. 1 root root 6 6月  3 09:33 a
```

【例9.32】 设置文件a具有Sticky权限。

```
[root@PC-LINUX ~]# chmod 1000 a  
[root@PC-LINUX ~]# ls -l a  
-----T. 1 root root 6 6月  3 09:33 a
```

【例9.33】 设置文件a具有SUID，SGID和Sticky权限。

```
[root@PC-LINUX ~]# chmod 7000 a  
[root@PC-LINUX ~]# ls -l a  
---S--S--T. 1 root root 6 6月  3 09:33 a
```



【例9.34】 设置/home/user目录连同他的子文件夹的权限为777。

```
[root@PC-LINUX ~]# mkdir /home/user
```

```
[root@PC-LINUX ~]# touch /home/user/abc
```

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

//表示将整个/home/user目录及其中的文件和子目录的权限都设置读取、写入和执行

```
[root@PC-LINUX ~]# ls -l /home|grep user
```

```
drwxrwxrwx. 2 root  root  4096 6月  3 09:44 user
```

```
[root@PC-LINUX ~]# ls -l /home/user/abc
```

```
-rwxrwxrwx. 1 root root 0 6月  3 09:44 /home/user/abc
```

## 9.7.3 更改文件和目录的所有权

文件和目录的创建者默认就具有所有权，他们对该文件和目录具有任何权限，可以进行任何操作。他们也可以将所有权交给别的用户，使别的用户对该文件和目录具有任何操作权限。文件和目录的所有者及所属用户组也能修改，用户可以通过图形界面来设置，也可以通过其他命令来修改。

# 1. chown命令

使用chown命令可以更改文件和目录的所有者和用户组。

命令语法:

```
chown [-R][用户.组][文件|目录]
```

【例9.35】 将文件a的所有者改成newuser。

```
[root@PC-LINUX ~]# ls -l a
-r--r--r--. 1 root root 6 6月  3 09:33 a
//目前文件a的所有者和用户组是root
[root@PC-LINUX ~]# chown newuser a
[root@PC-LINUX ~]# ls -l a
-r--r--r--. 1 newuser root 6 6月  3 09:33 a
//更改所有者后，可以看到当前文件a所有者为newuser
```

【例9.36】 将文件a的用户组改成newuser。

```
[root@PC-LINUX ~]# chown :newuser a
[root@PC-LINUX ~]# ls -l a
-r--r--r--. 1 newuser newuser 6 6月  3 09:33 a
//更改用户组后，可以看到当前用户组为newuser
```

【例9.37】 将文件a的所有者和用户组一起改成root。

```
[root@PC-LINUX ~]# chown root.root a  
[root@PC-LINUX ~]# ls -l a  
-r--r--r--. 1 root root 6 6月  3 09:33 a
```

【例9.38】 将文件a的用户组改成newuser。

```
[root@PC-LINUX ~]# chown .newuser a  
[root@PC-LINUX ~]# ls -l a  
-r--r--r--. 1 root newuser 6 6月  3 09:33 a
```

【例9.39】 将目录/root/b连同它的下级文件/root/b/ccc的所有者和用户组一起更改为newuser。

```
[root@PC-LINUX ~]# ls -l /root|grep b
drwxr-xr-x. 2 root root    4096 6月  3 09:51 b
[root@PC-LINUX ~]# ls -l /root/b/ccc
-rw-r--r--. 1 root root 0 6月  3 09:51 /root/b/ccc
//查看目录/root/b和文件/root/b/cc所有者和用户组，当前为root
[root@PC-LINUX ~]# chown -R newuser.newuser /root/b
[root@PC-LINUX ~]# ls -l /root|grep b
drwxr-xr-x. 2 newuser newuser 4096 6月  3 09:51 b
[root@PC-LINUX ~]# ls -l /root/b/ccc
-rw-r--r--. 1 newuser newuser 0 6月  3 09:51 /root/b/ccc
//查看目录/root/b和文件/root/b/cc所有者和用户组，当前为newuser
```

## 2. chgrp命令

使用chgrp命令可以更改文件或目录所属的组。

命令语法:

chgrp [选项][用户组][文件|目录]

【例9.40】 更改文件a的用户组为newuser。

```
[root@PC-LINUX ~]# ls -l a
-r---w-r--. 1 root root 6 6月  3 09:33 a
//文件a当前的用户组是root
[root@PC-LINUX ~]# chgrp newuser a
[root@PC-LINUX ~]# ls -l a
-r---w-r--. 1 root newuser 6 6月  3 09:33 a
```



# 小 结

分区就是硬盘的“段落”，如果用户希望在计算机上安装多个操作系统，将需要更多的分区。在Linux系统下，它本身又有更多的分区，比如，根分区“/”和交换分区“SWAP”。在Linux系统下使用fdisk命令能将磁盘划分成为若干个分区，同时也能为每个分区指定分区的文件系统。

## 小 结

文件系统通过为每个文件分配文件块的方式把数据存储在存储设备中，这样就要维护每一个文件的文件块的分配信息，而分配信息本身也要存在磁盘上，不同的文件系统用不同的方法分配和读取文件块。Linux系统内核可以支持几十种文件系统类型，如JFS，ReiserFS，ext，ext2，ext3，ext4，IS09660，XFS，Minix，MSDOS，VFAT，NTFS，HPFS，NFS，SMB及SysV等。

## 小 结

对一个新的硬盘进行分区以后，还要对这些分区进行格式化并创建文件系统，一个分区只有建立了某种文件系统后，这个分区才能使用。以Fedora 17系统为例，常用的创建文件系统的命令有mkfs, mkfs.ext4, mkfs.ext3, mkfs.msdos, mkfs.vfat和mkswap等。

## 小 结

使用mount和umount命令可以实现挂载和卸载功能，这样用户可以将某个分区、光盘、软盘或是U盘进行挂载或卸载。要查看Linux系统分区挂载情况，除了可以使用df命令之外，还可以通过mount -s命令以及/查看etc/mtab文件获取信息。

## 小 结

将某个分区或是设备挂载了以后才能使用，但是当计算机重新启动以后，又需要重新挂载，这个时候可以通过修改/etc/fstab文件实现开机自动挂载文件系统。/etc/fstab文件是一个配置文件，它包含了所有分区以及存储设备的信息。其中包含了磁盘分区和存储设备如何挂载，以及挂载在什么地方的信息。

## 小 结

如果系统需要更多的内存资源，而物理内存已经用完，内存中不活跃的页就会被转移到交换空间中。虽然交换空间可以为带有少量内存的计算机提供帮助，但是这种方法不应该被当作是对内存的取代。交换空间位于硬盘驱动器上，它比进入物理内存要慢。交换空间可以是一个专用的交换分区，也可以是一个交换文件，或是两者的结合。

# 小 结

在Linux系统中，用户可以对每一个文件或目录都具有读取、写入和执行访问权限，这些访问权限决定了谁能访问，以及如何访问这些文件和目录。只有系统管理员和文件的所有者才可以更改文件或目录的权限，更改文件或目录权限的方法一般有使用文件管理器、文字设定法、数字设定法3种。

## 小 结

文件和目录的创建者默认就具有所有权，他们对该文件和目录具有任何权限，可以进行任何操作。他们也可以将所有权交给别的用户，使别的用户对该文件和目录具有任何操作权限。文件和目录的所有者及所属用户组也能修改，用户可以通过图形界面来设置，也可以通过 `chown` 和 `chgrp` 命令来修改。