

# Linux操作系统及应用技术

磁盘管理



# 主要学习内容

1、磁盘的分区类型；

2、磁盘的分区结构；

3、磁盘的接口类型；

4、Linux系统对各个分区的命名；

quota ['kwəutə] 

n. 配额；限额；定额

5、Linux中查看分区、创建分区的命令 fdisk ；

6、Linux中的磁盘配额 Disk Quota 。

## ➤ 重点：

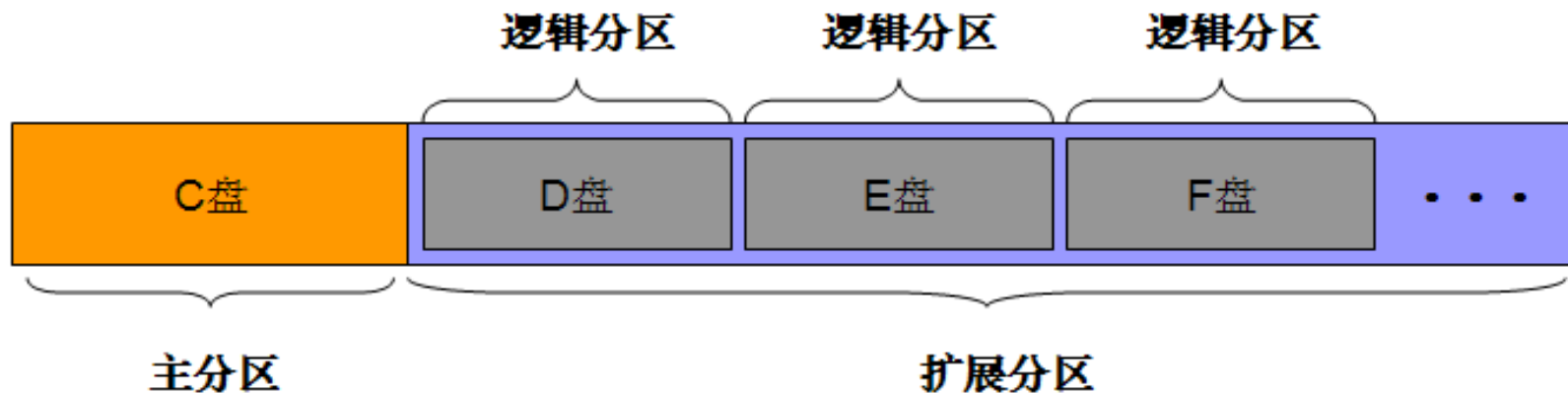
- (1) Linux中查看分区、创建分区的命令 `fdisk` ；
- (2) Linux中的磁盘配额 `Disk Quota` 。

## ➤ 难点：

- (1) Linux中的磁盘配额 `Disk Quota` 。

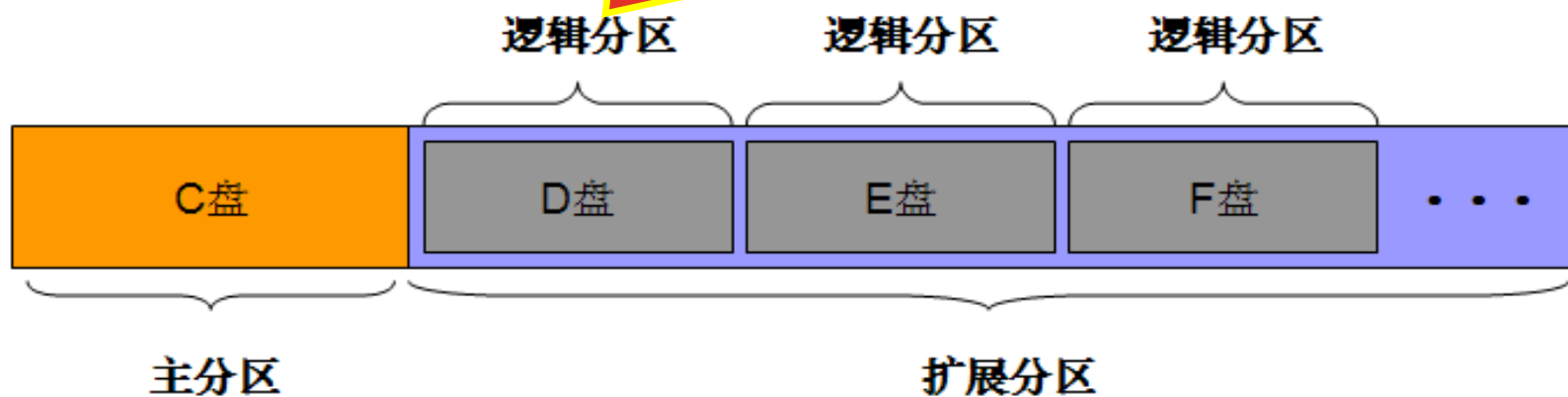
# 1、磁盘的分区类型

磁盘 1 基本 149.04 GB 联机				
	(C:)	DISK1_VOL2 (D:)	DISK1_VOL3 (E:)	DISK1_VOL4 (F:)
	9.32 GB NTFS 状态良好 (系统)	27.95 GB FAT32 状态良好	37.26 GB FAT32 状态良好	74.52 GB FAT32 状态良好



在扩展分区中继续创建的分区，称为**逻辑分区**：

- ✓ Windows中的D盘、E盘、F盘等，都是属于逻辑分区；
- ✓ 在逻辑分区中，可以保存文件。




**扩展分区**包含着Windows中的D盘、E盘、F盘等：

- ✓ 扩展分区并不能直接保存文件；
- ✓ 需要在扩展分区中继续创建**逻辑分区**，即创建D盘、E盘、F盘等，才能保存文件。

我们在Windows中常见的C盘，就是属于**主分区**的类型：

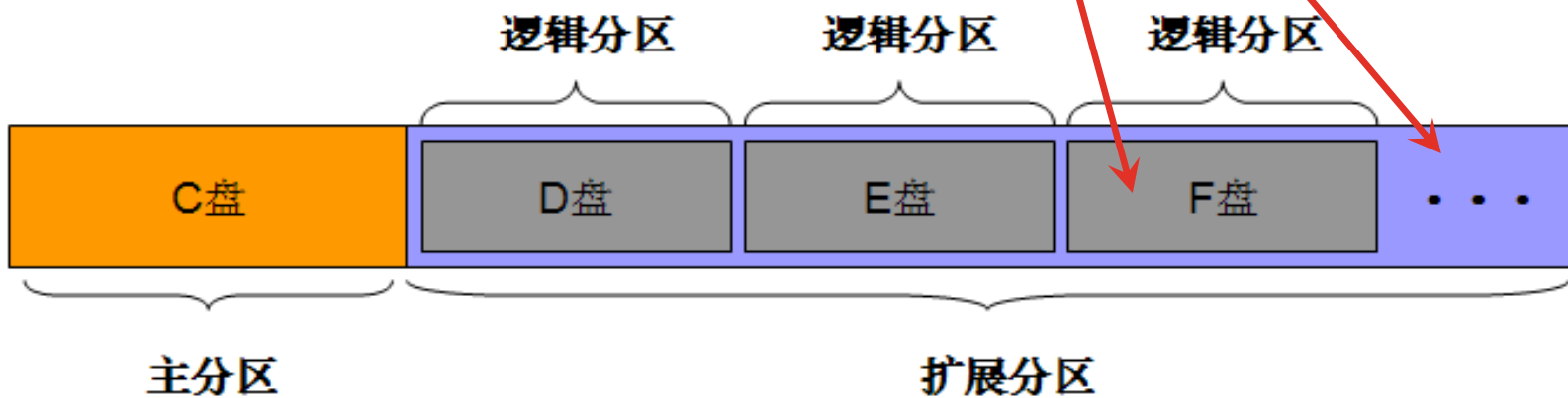
- ✓ 主分区用来安装操作系统的系统文件和启动文件；
- ✓ 主分区是不能够再继续划分分区的。



从以上概念可以得知，在一个硬盘中，只有**主分区**和**逻辑分区**才能够保存文件或数据。

➤ 3种分区类型之间的关系：

很明确，在扩展分区中包含着逻辑分区。

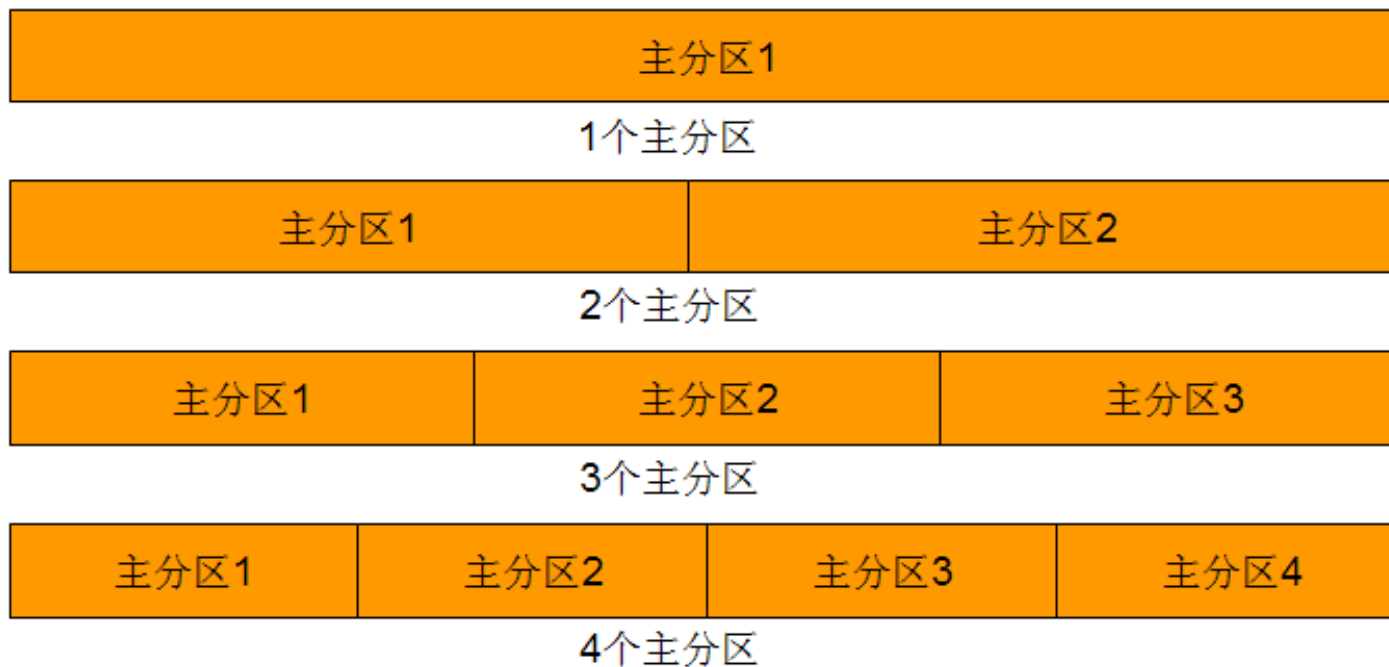


主分区与扩展分区属于**并行的、并列的、谁也不包含谁**的关系。

## 2、磁盘的分区结构

- 一块硬盘中，能够创建多少个**主分区**？多少个**扩展分区**？
  - ✓这是磁盘的分区结构问题。
  - ✓1—4

(1) 一块硬盘中，可以创建1个主分区、2个主分区、3个主分区、4个主分区。最多只能创建4个主分区，即：**主分区 $\leq$ 4**，如下图所示：

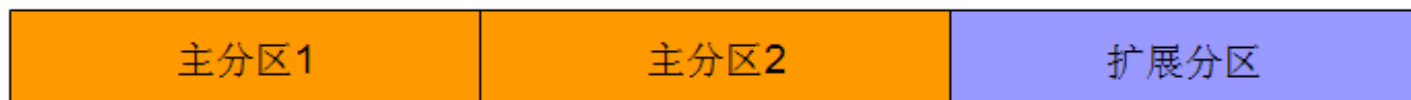


(2) 既创建主分区，又创建扩展分区，则：

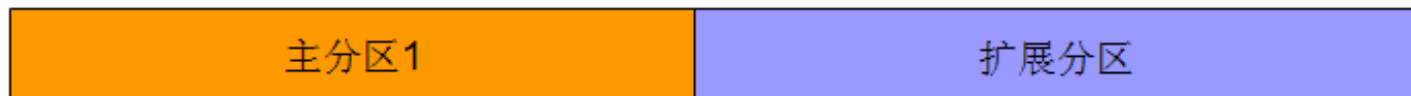
主分区 + 扩展分区  $\leq 4$ ，如下图所示：



3个主分区 + 1个扩展分区



2个主分区 + 1个扩展分区



1个主分区 + 1个扩展分区

这是最常见的分区情况

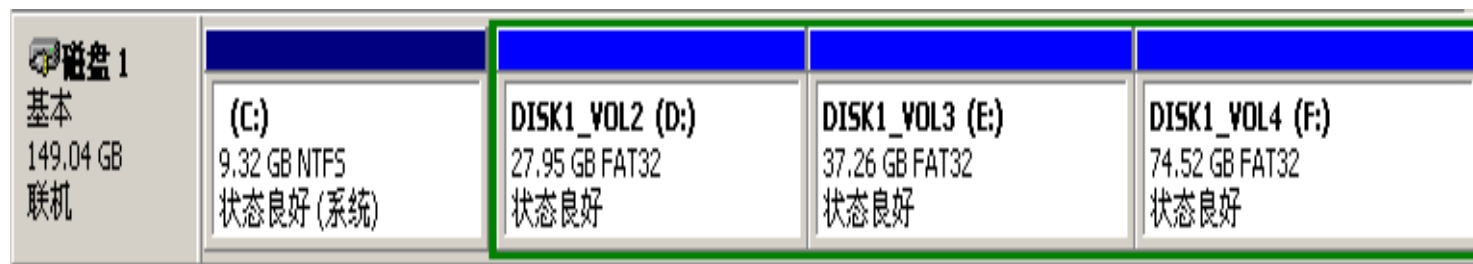
📖 在一个磁盘中，为什么只能够创建：

✓4个主分区（**主分区**≤4）？或者

✓4个主分区和扩展分区（**主分区** + **扩展分区**≤4）？

📖 这跟操作系统（Windows/Linux/Unix）没有关系。这是磁盘上分区表的限制，这是一种物理上的限制。

📖 1个主分区 + 1个扩展分区，是我们电脑中最常见的划分情况



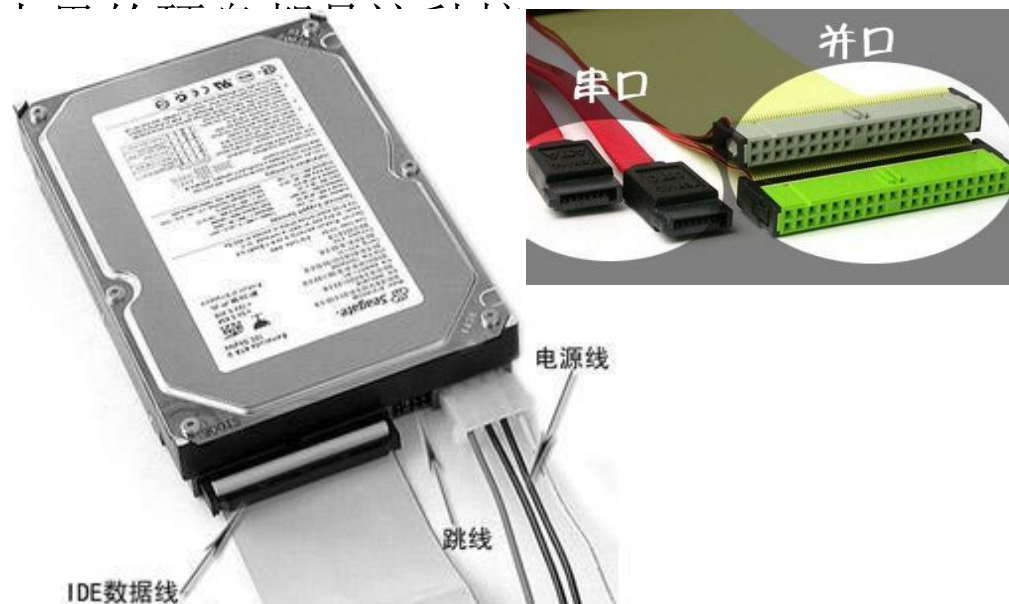
磁盘 1	(C:)	DISK1_VOL2 (D:)	DISK1_VOL3 (E:)	DISK1_VOL4 (F:)
基本 149.04 GB 联机	9.32 GB NTFS 状态良好 (系统)	27.95 GB FAT32 状态良好	37.26 GB FAT32 状态良好	74.52 GB FAT32 状态良好

### 3、磁盘的接口类型

📖 硬盘的接口类型，有3种：IDE接口、SATA接口、SCSI接口。

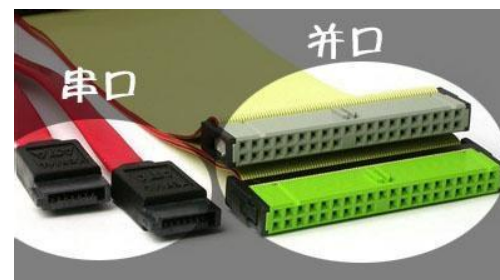
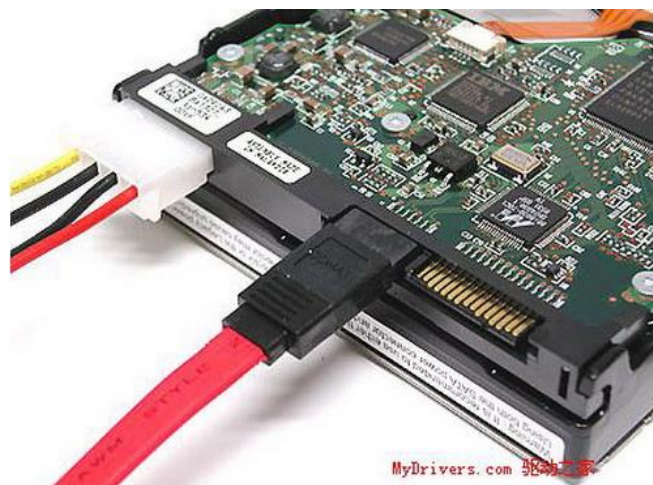
(1) IDE接口

✓2005年及以前，台式机和笔记本电脑广泛使用IDE接口。



## (2) SATA接口

- ✓2005年以后，台式机和笔记本里的硬盘基本上都是这种接口；
- ✓在Linux中，标识为**sd**；
- ✓U盘、移动硬盘插入Linux中，也标识为**sd**。

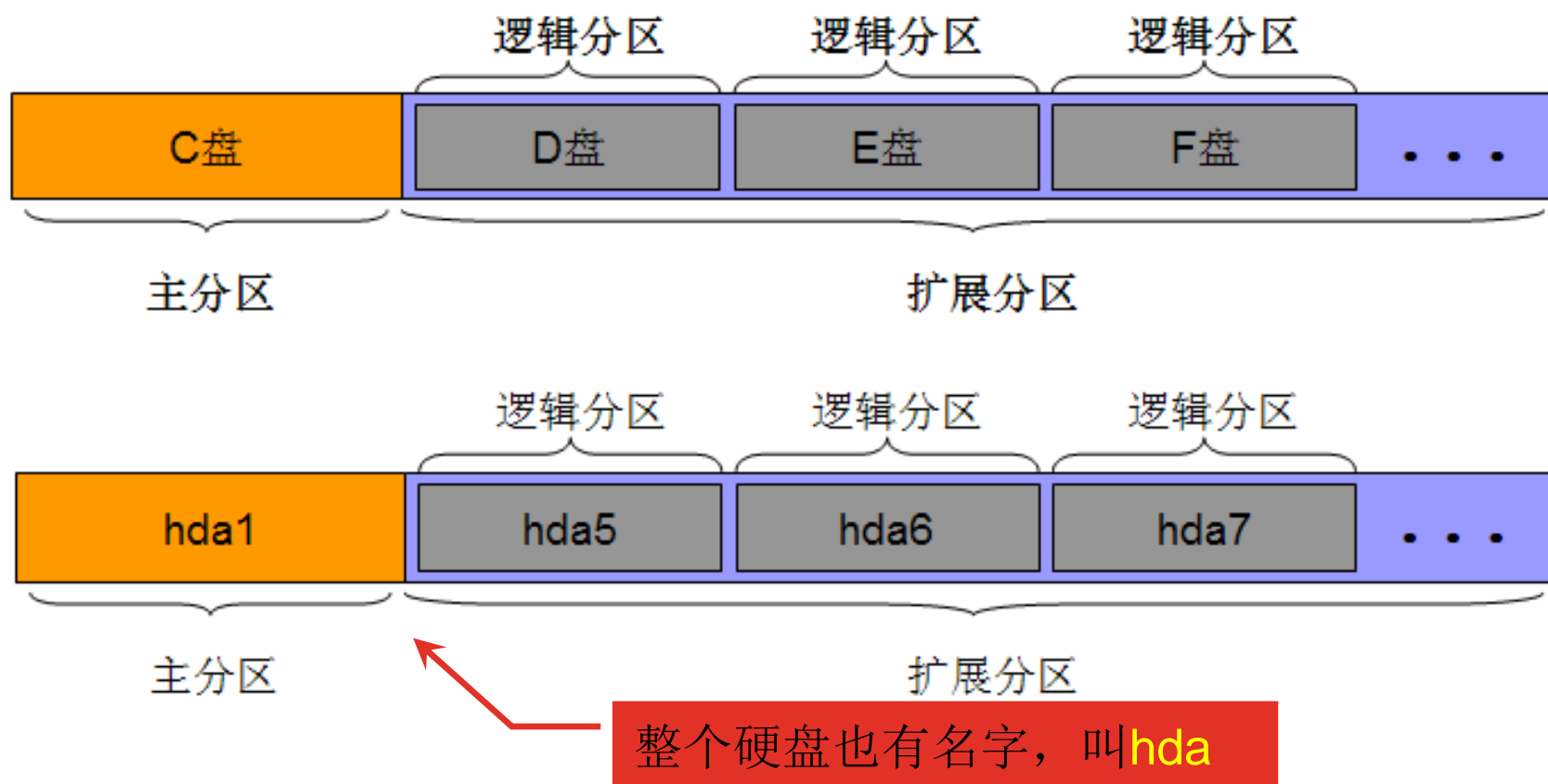


### (3) SCSI接口

- ✓【专业服务器】中的硬盘接口；
- ✓在Linux中，标识为sd。



## 4、Linux系统对各个分区的命名

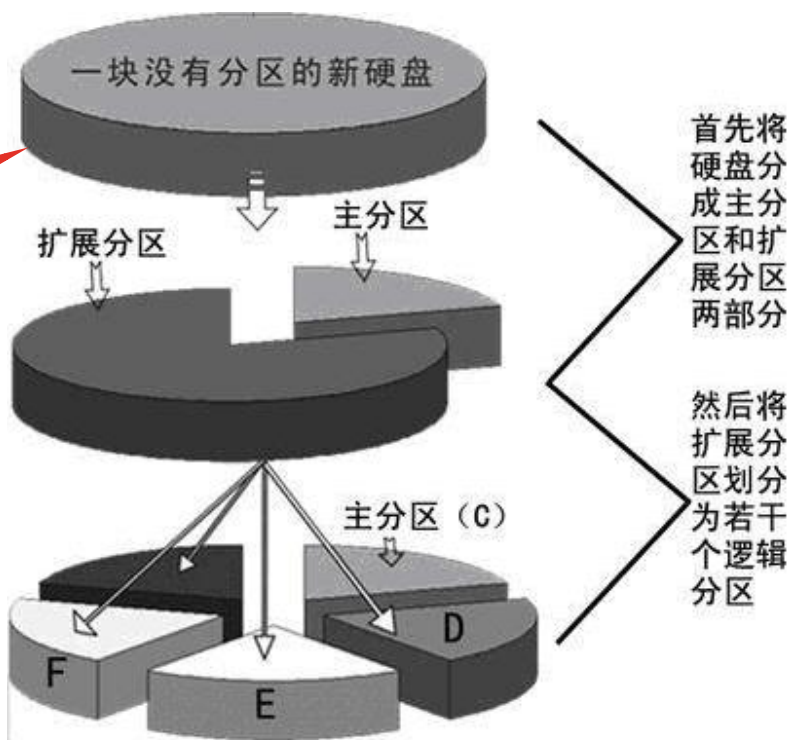


磁盘 1 基本 149.04 GB 联机	(C:) 9.32 GB NTFS 状态良好 (系统)	DISK1_VOL2 (D:) 27.95 GB FAT32 状态良好	DISK1_VOL3 (E:) 37.26 GB FAT32 状态良好	DISK1_VOL4 (F:) 74.52 GB FAT32 状态良好
	/dev/hda1	/dev/hda5	/dev/hda6	/dev/hda7

- hda1、hda5、hda6、hda7，都是设备文件，分别表示硬盘中的各个分区。这些设备文件，都保存在/dev目录下。
- 前2个字母表示硬盘的接口类型。
  - ✓ “hd”表示IDE接口；
  - ✓ “sd”表示SATA接口、SCSI接口、U盘、移动硬盘。
- 第3个字母，表示Linux系统中的第几块硬盘。
  - ✓ a表示第1块硬盘、b表示第2块硬盘、c表示第3块硬盘……。
  - ✓ hda或sda，表示整个硬盘的名字。
- 第4个数字，表示是硬盘中的第几个分区。

磁盘 1 基本 149.04 GB 联机	(C:)	DISK1_VOL2 (D:)	DISK1_VOL3 (E:)	DISK1_VOL4 (F:)
	9.32 GB NTFS	27.95 GB FAT32	37.26 GB FAT32	74.52 GB FAT32
	状态良好 (系统)	状态良好	状态良好	状态良好
	/dev/hda1	/dev/hda5	/dev/hda6	/dev/hda7

给一个新的硬盘  
创建分区的过程



## 5、Linux中查看分区与创建分区的命令 fdisk

### ➤ 查看分区操作：

```
[root@bogon ~]# fdisk -l
Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000020f2

   Device Boot      Start         End      Blocks   Id  System
  /dev/sda1  *           1          13        102400   83  Linux
  /dev/sda2                13        2563     2048000   83  Linux
  /dev/sda3                2563       2818       204800   82  Linux swap / Solaris
  /dev/sda4                2818       3917     8825856    5  Extended
  /dev/sda5                2818       2882       512000   83  Linux
[root@bogon ~]#
```

(2) 查看分区的挂载点，即查看分区挂载在哪个目录下：

# `df -h`

```
[root@bogon ~]# df -h
文件系统          容量  已用  可用  已用%%  挂载点
/dev/sda2          20G  2.5G  16G   14%    /
tmpfs              504M  88K  504M   1%    /dev/shm
/dev/sda1          97M  24M  69M   26%    /boot
/dev/sda5          485M  11M  449M   3%    /home
[root@bogon ~]#
```

📖 进入各个分区操作，即进入各个分区的挂载目录即可：

- ✓ # `cd /` //进入根分区，即/dev/sda2
- ✓ # `cd /boot` //进入boot分区，即/dev/sda1
- ✓ # `cd /home` //进入home分区，即/dev/sda5

(3) 查看分区的挂载在哪个目录下:

# `mount`

```
[root@bogon ~]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda1 on /boot type ext4 (rw)
/dev/sda5 on /home type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
[root@bogon ~]#
```

## ➤ 创建分区操作：

📖 例子：创建一个500M的分区，开机挂载在/backup目录下。

(1) 首先，查看Linux系统中：

- ✓ 有几个硬盘？
- ✓ 硬盘的名字？
- ✓ 硬盘总的大小？
- ✓ 已有的分区及类型？
- ✓ 硬盘中还有没有剩余的空闲空间来满足创建新的500M的分区？

```
# fdisk -l //小写字母l
```

- ✓ 硬盘的名字: /dev/sda
- ✓ 硬盘的大小: 32G

- ✓ 255个磁头
- ✓ 每个磁道有63个扇区
- ✓ 整个硬盘一共有3916个柱面

```
[root@localhost ~]# fdisk -l
Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000020f2
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	102400	83	Linux
Partition 1 does not end on cylinder boundary.						
/dev/sda2		13	2563	2048000	83	Linux
/dev/sda3		2563	2818	2048000	82	Linux swap / Solaris
/dev/sda4		2818	3917	8825856	5	Extended
/dev/sda5		2818	2882	512000	83	Linux

```
[root@localhost ~]#
```

- ✓ sda1、sda2、sda3是主分区
- ✓ sda4是扩展分区
- ✓ sda5是逻辑分区

- ✓ 每个分区的开始Start柱面、结束End柱面。
- ✓ 扩展分区结束柱面是3917，表示使用到了硬盘的最后一个柱面3916，也表示把硬盘的空间用完了。
- ✓ 逻辑分区sda5的结束柱面是2882，还没到3917表示在扩展分区里还有剩余空间1035\*8M=8280M。

(2) 接着，进入fdisk命令的界面。输入子命令m，查看帮助，即查看可以用哪些子命令来创建分区。

```
# fdisk /dev/sda //硬盘的名字
```

```
[root@localhost ~]# fdisk /dev/sda
```

```
WARNING: DOS-compatible mode is deprecated. It is recommended to  
switch off the mode (command 'c') and change display units to  
sectors (command 'u').
```

```
Command (m for help): m
```

```
Command action
```

```
 a toggle a bootable flag  
 b edit bsd disklabel  
 c toggle the dos compatibility flag  
 d delete a partition  
 l list known partition types  
 m print this menu  
 n add a new partition  
 o create a new empty DOS partition table  
 p print the partition table  
 q quit without saving changes  
 s create a new empty Sun disklabel  
 t change a partition's system id  
 u change display/entry units  
 v verify the partition table  
 w write table to disk and exit  
 x extra functionality (experts only)
```

```
Command (m for help):
```

对硬盘/dev/sda进行分区

m: 获得帮助信息

d: 删除一个分区

m: 获得帮助信息

n: 新建一个分区

p: 显示分区表信息

q: 退出不存盘

w: 存盘退出

(3) 输入子命令p，查看已有分区的情况。子命令p显示出来的信息，和 fdisk -l 显示的信息一样。

```
Command (m for help): p
Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000020f2

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1          13        102400   83  Linux
Partition 1 does not end on cylinder boundary
/dev/sda2                13         2563        204800   83  Linux
/dev/sda3                2563        2818        204800   83  Linux
/dev/sda4                2818        3917        882000   83  Linux
/dev/sda5                2818        2882        512000   83  Linux

Command (m for help):
```

硬盘总的柱面数有3916

扩展分区用到了硬盘的最后一个柱面，表示整个硬盘都没有剩余空间了

但是，在扩展分区里还有剩余空间，因为它里面的逻辑分区/dev/sda5才用到了第2882个柱面，还剩下1035\*8M=8280M的空间

(4) 然后，用子命令n来创建一个新的分区。

```
Command (m for help): n
First cylinder (2882-3917, default 2882):
Using default value 2882
Last cylinder, +cylinders or +size{K,M,G} (2882-3917, default 3917): +500M
```

输入新分区的第1个柱面，默认从第2882个柱面开始

```
Command (m for help): p
```

```
Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000020f2
```

输入新分区的最后1个柱面，但我们直接新建分区要求的大小，注意格式：**+500M**

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1				
Partition 1 does not end on cylinder boundary						
/dev/sda2		13				
/dev/sda3		2563				
/dev/sda4		2818	3917	8825856	5	Extended
/dev/sda5		2818	2882	512000	83	Linux
/dev/sda6		2882	2946	518241+	83	Linux

查看新建立的分区/dev/sda6，其大小为(2946-2882)\*8M=512M，这是一个正常的误差。

```
Command (m for help):
```

(5) 若没有问题，则输入子命令w存盘退出。若有问题，则输入子命令q不存盘退出。

输入子命令w，存盘退出。

```
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
WARNING: Re-reading the partition table failed with error 16: 设备或资源忙.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
[root@localhost ~]# ls /dev/sda*
/dev/sda /dev/sda1 /dev/sda2 /dev/sda3 /dev/sda4 /dev/sda5
[root@localhost ~]#
```

根据提示信息及查询的结果来看，新建分区的设备文件/dev/sda6还没有产生，新建分区暂时还没有生效，需重启Linux才能生效。

(6) 不重启，让新创建的分区生效的办法。

# `partx -a /dev/sda`

```
[root@localhost ~]# partx -a /dev/sda
-a: No such file or directory
[root@localhost ~]# partx -a /dev/sda
BLKPG: Device or resource busy
error adding partition 1
BLKPG: Device or resource busy
error adding partition 2
BLKPG: Device or resource busy
error adding partition 3
BLKPG: Device or resource busy
error adding partition 4
BLKPG: Device or resource busy
error adding partition 5
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# ls /dev/sda*
/dev/sda /dev/sda1 /dev/sda2 /dev/sda3 /dev/sda4 /dev/sda5 /dev/sda6
```

执行该命令，让新建的分区立即生效。

能看到新建分区的设备文件/dev/sda6，表示新建的分区生效了。

(7) 格式化、创建挂载的目录/backup、挂载。

```
# mkfs.ext4 /dev/sda6
```

```
# mkdir /backup
```

✓ 将分区/dev/sda6与目录/backup关联起来，称为挂载。

```
# mount /dev/sda6 /backup
```

✓ 检查挂载的情况。

```
# mount
```

```
# df -h
```

分区 主分区 扩展分区 逻辑分区

格式化 ext4

新建两目录 作为挂载点 d e 挂载

```
[root@bogon ~]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda1 on /boot type ext4 (rw)
/dev/sda5 on /home type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/sda6 on /backup type ext4 (rw)
[root@bogon ~]# df -h
文件系统      容量  已用  可用  已用%%  挂载点
/dev/sda2      20G  2.4G  16G   13%  /
tmpfs          504M  88K  504M   1%  /dev/shm
/dev/sda1      97M  24M  69M   26%  /boot
/dev/sda5      485M  11M  449M   3%  /home
/dev/sda6      491M  11M  455M   3%  /backup
[root@bogon ~]#
```

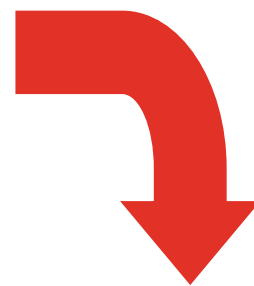
查看挂载的情况

查看挂载的情况

(8) 完成**开机自动挂载**的设置。

- ✓ 第6步操作中的**mount /dev/sda6 /backup**，是手动挂载。以后每次重新启动Linux，都需要再进行一次手动挂载。
- ✓ 能不能让Linux在开机的时候，**自动挂载该分区**呢？
- ✓ 编辑**/etc/fstab**文件。写在该文件中的所有分区，都会在Linux启动的过程中，自动挂载里面所有的分区。
- ✓ # **vi /etc/fstab**

```
[root@bogon ~]# vi /etc/fstab
[root@bogon ~]#
```



```
 9 UUID=91235379-f035-4458-9f9a-db81730ae284 / ext4 defaults 1 1
10 UUID=5c5f03a8-952a-4445-84da-fc8ee68556ca /boot ext4 defaults 1 2
11 UUID=b1b7ec48-2804-4326-b811-d2f705733a05 /home ext4 defaults 1 2
12 UUID=d3b4d7d0-be5d-43b8-9477-89d7003017c3 swap swap defaults 0 0
13 tmpfs /dev/shm tmpfs defaults 0 0
14 devpts /dev/pts devpts gid=5,mode=620 0 0
15 sysfs /sys sysfs defaults 0 0
16 proc /proc proc defaults 0 0
17 /dev/sda6 /backup ext4 defaults 0 0
```

- ✓ 记住了：这个文件相当重要。添加自动挂载的分区信息，千万不要写错了。也千万不要误改其他行的信息，错一个字母都不行。
- ✓ 轻则导致分区挂载失败，重则导致Linux启动不了。

(9) 检查刚才编辑的 `/etc/fstab` 文件是否有错误。

✓ 卸载刚才手动挂载的分区 `/dev/sda6`:

```
# umount /dev/sda6
```

✓ `mount` 一下，查看是否将分区 `/dev/sda6` 卸载掉了。

```
# mount
```

✓ 执行命令 `mount -a`，该命令会把 `/etc/fstab` 文件中所有的分区自动再挂载一次。若没有出错提示信息，再执行 `mount` 命令，查看到分区 `/dev/sda6` 是否再次挂载。若挂载，那说明 `/etc/fstab` 文件设置的没错。

```
# mount -a
```

```
# mount
```



```
[root@bogon ~]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda1 on /boot type ext4 (rw)
/dev/sda5 on /home type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/sda6 on /backup type ext4 (rw)
[root@bogon ~]# umount /dev/sda6
[root@bogon ~]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda1 on /boot type ext4 (rw)
/dev/sda5 on /home type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
[root@bogon ~]# mount -a
[root@bogon ~]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda1 on /boot type ext4 (rw)
/dev/sda5 on /home type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/sda6 on /backup type ext4 (rw)
```

查看挂载的情况

卸载分区/dev/sda6

查看卸载的情况

该命令会把/etc/fstab文件中所有的分区自动再挂载一次

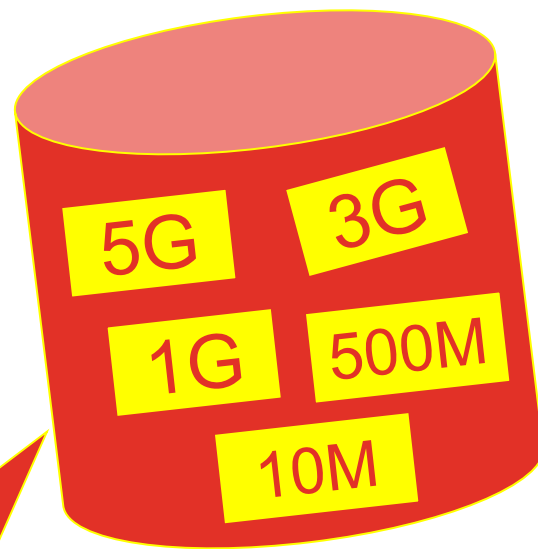
查看挂载的情况

quota ['kwɒtə] 

n. 配额; 限额; 定额

## 6、Linux中的磁盘配额Disk Quota

- ✓ 你在网上申请的网盘，有大小限制吗？
- ✓ 你在网上申请的邮箱，有大小限制吗？
- ✓ 你在网上申请的网站，有大小限制吗？



这就是**磁盘配额**

## ➤ 磁盘配额:

- ✓就是给你划分或规定你能够使用多大的磁盘空间;
- ✓既可以从**空间大小**上来进行限制, 又可以从可以创建**多少个文件的数量**上来进行限制。
- ✓磁盘配额, 是针对**某个分区**做的一种设置, 而不是针对某个目录做的设置。

```
[root@bogon ~]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=620)
tmpfs on /dev/shm type tmpfs (rw,root=ext="system_u:object_r
/dev/sda1 on /boot type ext4 (rw)
/dev/sda5 on /home type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
gvfs-fuse-daemon on /root/.gvfs type fuse.gvfs-fuse-daemon (rw,r
/dev/sr0 on /media/RHEL_6.0 i386 Disc 1 type iso9660 (ro,nosuid,
isks,uid=0,gid=0,icharset=utf8,mode=0400,dmode=0500)
[root@bogon ~]#
```

✓用mount命令查看一下各个分区及其挂载的参数。

✓拟针对分区/dev/sda5设置磁盘配额

➤ 设置磁盘配额的方法或步骤:

(1) 修改分区/dev/sda5的挂载参数, 即增加以下两个参数:

```
# mount -o remount,usrquota,grpquota /home
```

✓-o remount: 重新挂载

✓usrquota: 设置该分区的**用户**磁盘配额 (User Quota) 参数

✓grpquota: 设置该分区的**用户组**磁盘配额 (Group Quota) 参数

✓/home分区 或 /dev/sda5分区, 这2种写法是等效的。

```
[root@bogon ~]# mount -o remount,usrquota,grpquota /home
[root@bogon ~]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object
```

/dev/sda5 on /home type ext4 (rw,usrquota,grpquota)

```
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
gvfs-fuse-daemon on /root/.gvfs type fuse.gvfs-fuse-daemon (
/dev/sr0 on /media/RHEL_6.0 type iso9660 (ro,nos
```

isks,uid=0,gid=0,icharset=

```
[root@bogon ~]#
```

增加usrquota、grpquota这两个参数

(2) 创建/dev/sda5分区的磁盘配额数据库:

```
# quotacheck -ugc /dev/sda5
```

✓-u: user, 建立用户的磁盘配额数据库[aquota.user](#)

✓-g: group, 建立用户组的磁盘配额数据库[aquota.group](#)

✓-c: create, 创建以上两个数据库文件。

```
[root@bogon ~]# quotacheck -ugc /home
[root@bogon ~]# cd /home
[root@bogon home]# ls
aquota.group  jack      student  user01  user04  user06  user08
aquota.user   lost+found tom       user02  user05  user07  user09
[root@bogon home]
```

以下两个参数, 都行, 是等效的  
✓ /home  
✓ /dev/sda5

必须要在/dev/sda5分区的根目录中看到这两个文件

(3) 启用（或启动）磁盘配额功能：

# **quotaon** [-p] [-a] [某个分区]

✓-p: 仅显示磁盘的配额功能是否启用；

✓-a: 启用所有已设置usrquota、grpquota参数的分区的磁盘配额功能；

✓某个分区，只启用指定的分区的磁盘配额功能

```
[root@bogon home]# quotaon -p /dev/sda5  
group quota on /home (/dev/sda5) is off  
user quota on /home (/dev/sda5) is off  
[root@bogon home]#
```

查看磁盘配额的状态，默认是关闭的。

启用磁盘配额

```
[root@bogon home]# quotaon -a  
[root@bogon home]# quotaon -p /home  
group quota on /home (/dev/sda5) is on  
user quota on /home (/dev/sda5) is on  
[root@bogon home]#
```

查看磁盘配额的状态，启用了。

(4) 设置某个用户或某个用户组能够使用**多大的磁盘空间**:

# **edquota** [-u | -g] **用户名或用户组名**

✓-u: 设置某个用户的磁盘配额大小时, 可以不用加该参数;

✓-g: 设置某个用户组的磁盘配额大小时, 一定要加该参数;

✓注意, 以超级用户root的身份来做这个设置

```
[root@bogon home]# edquota student  
[root@bogon home]#
```

以**用户student**为例, 设置该用户能够使用多大的磁盘空间





实际上，这是vi的界面

设置用户student的磁盘配额

```
Disk quotas for user student (uid 500):  
Filesystem      blocks  soft  hard  inodes  soft  hard  
/dev/sda5      18      0    0     9      0    0
```

设置可以使用的空间大小，以K为单位：  
**blocks**：已使用的空间大小；  
**soft**：软限制，即允许超过的界限，但会报警；  
**hard**：硬限制，即不允许超过的界限，达到这个界限之后，就没有空间可用了。

设置用户student在/dev/sda5分区中的磁盘配额

设置可以创建的文件或目录的数量：  
**inodes**：已有的文件或目录的数量；  
**soft**：软限制，即允许超过的界限，但会报警；  
**hard**：硬限制，即不允许超过的界限，达到这个界限之后，就无法再创建文件或目录了。

**soft**: 软限制, 2048K, 即2M

注意: 此处设置的大小, 不能小于已用空间的大小, 即不能小于**blocks**的值。

```
Disk quotas for user student (uid 500):  
Filesystem      blocks  soft  hard  inodes  soft  hard  
/dev/sda5       18     2048 5120   9        0     0  
~
```

**hard**: 硬限制, 5120K, 即5M

注意: 此处设置的大小, 要大于**soft**的值。

设置完毕, 输入:**wq**, 存盘退出。

(5) 以student用户的身份来测试对其设置的磁盘配额（测试其能使用的空间大小）：

```
# dd if=/dev/zero of=file1 bs=1K count=512
```

- ✓利用该命令，可以人为地创建一个指定大小的文件
- ✓if: input file, 表示指定输入文件
- ✓of: output file, 表示指定输出文件
- ✓bs: blocks, 指定每次读取的块的大小，单位可以是字节、K、M
- ✓count:指定读取多少块。

```
# quota //查看该用户磁盘配额使用的情况
```

```
[root@bogon home]# su - student
[student@bogon ~]$ pwd
/home/student
[student@bogon ~]$ ls
[student@bogon ~]$ quota
Disk quotas for user student (uid 500):
  Filesystem  blocks   quota  limit  grace  files  quota  limit  grace
    /dev/sda5     18   2048   5120      0      9      0      0      0
[student@bogon ~]$ dd if=/dev/zero of=file1 bs=1M count=2
记录了2+0 的读入
记录了2+0 的写出
2097152字节(2.1 MB) 已复制, 0.0224317 秒, 93.5 MB/秒
[student@bogon ~]$ ls -lh
总用量 2.1M
-rw-rw-r--. 1 student student 2.0M 1月 18 02:51 file1
[student@bogon ~]$ quota
Disk quotas for user student (uid 500):
  Filesystem  blocks   quota  limit  grace  files  quota  limit  grace
    /dev/sda5  2067*   2048   5120   7days    10      0      0      0
[student@bogon ~]$
```

切换到student用户的身份

查看该用户磁盘配额使用的情况

18<2048, 没有超出软限制, 还没有出现过期grace的天数

创建第1个文件file1, 2M

2067\*>2048, 超出软限制, 报警! 出现过期grace的天数: 7天

创建第2个文件file2, 3M

```
[student@bogon ~]$ dd if=/dev/zero of=file2 bs=1M count=3
```

```
dd: 正在写入"file2": 超出磁盘限额
```

提示: 超出硬限制

```
记录了3+0 的读入
```

```
记录了2+0 的写出
```

```
3125248字节(3.1 MB)已复制, 0.0357778 秒, 87.4 MB/秒
```

```
[student@bogon ~]$ quota
```

```
Disk quotas for user student (uid 500):
```

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/sda5	5120*	2048	5120	6days	11	0	0	

5120\*>5120, 超出硬限制, 报警!  
出现过期grace的天数: 6天

```
[student@bogon ~]$ cp /etc/passwd .
```

```
cp: 无法创建普通文件"./passwd": 超出磁盘限额
```

拷贝一个文件

```
[student@bogon ~]$ touch file3
```

```
touch: 无法创建"file3": 超出磁盘限额
```

```
[student@bogon ~]$
```

(6) 设置某个用户或某个用户组能够创建**多少个文件或目录**:

- ✓能够使用多大的磁盘空间，或能够创建多少个文件或目录，这两个条件是 OR 的关系，哪个条件先达到，就以哪个为准来限制用户对磁盘的使用。即以先到为准。
- ✓所以，为便于测试这个设置，需要**先删除在前面的测试中创建的那些文件**。

# **edquota [-u | -g] 用户名或用户组名**

- ✓-u: 设置某个用户的磁盘配额大小时，可以不用加该参数;

设置某个用户组的磁盘配额大小时，一定要加该参数

```
[root@bogon home]# edquota student
[root@bogon home]#
```

以**用户student**为例，设置该用户能够创建多少个文件或目录  
注意：以**超级用户root**的身份来做这个设置





**soft:** 软限制, 12个文件或目录

注意: 此处设置的大小, 不能小于已创建的文件或目录的数量, 即不能小于**inodes**的值。

```
Disk quotas for user student (uid 500):
```

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/sda5	18	2048	5120	9	12	15

**hard:** 硬限制, 15个文件或目录

注意: 此处设置的大小, 要大于**soft**的值。

(7) 以student用户的身份来测试对其设置的磁盘配额（测试其能够创建的文件或目录的数量）：

```
# touch file1
```

```
# quota //查看该用户磁盘配额使用的情况
```

```
[root@bogon home]# su - student
[student@bogon ~]$ pwd
/home/student
```

查看该用户磁盘配额使用的情况

```
[student@bogon ~]$ ls
[student@bogon ~]$ quota
Disk quotas for user student (uid 500):
```

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/sda5	18	2048	5120		9	12	15	

```
[student@bogon ~]$ touch file1
[student@bogon ~]$ touch file2
[student@bogon ~]$ touch file3
[student@bogon ~]$ quota
```

9<12, 没有超出软限制, 还没有出现过期grace的天数

```
Disk quotas for user student (uid 500):
Filesystem blocks quota limit grace files quota limit grace
/dev/sda5 21 2048 5120 12 12 15
```

```
[student@bogon ~]$ touch file4
[student@bogon ~]$ quota
```

12=12, 没有超出软限制, 还没有出现过期grace的天数

```
Disk quotas for user student (uid 500):
Filesystem blocks quota limit grace files quota limit grace
/dev/sda5 22 2048 5120 13* 12 15 7days
```

```
[student@bogon ~]$ touch file5
[student@bogon ~]$ touch file6
[student@bogon ~]$ touch file7
```

13\*>12, 超出软限制, 报警! 出现过期grace的天数: 7天

touch: 无法创建"file7": 超出磁盘限额

```
[student@bogon ~]$ quota
Disk quotas for user student (uid 500):
Filesystem blocks quota limit grace files quota limit grace
/dev/sda5 24 2048 5120 15* 12 15 6days
```

提示: 超出硬限制

15\*>15, 超出硬限制, 报警! 出现过期grace的天数: 6天

# Linux操作系统及应用技术

网络配置



Linux的网络配置可以通过桌面图形界面或相关命令来进行设置，对于熟练的使用者往往通过直接修改相应的配置文件来进行网络配置。本章将详细讲述Linux的网络配置命令及相关配置文件。



## 本章要点

**7.1 网络配置命令**

7.2 常用网络配置文件

## 1. ifconfig命令

使用权限:

root



说明

第1块以太网卡的设备名为eth0，其对应的配置文件名为ifcfg-eth0；第2块以太网卡的设备名为eth1，对应配置文件名则为ifcfg-eth1，其余依此类推。所有网卡设备的配置文件位于/etc/sysconfig/network-scripts目录中。

### 功能1 ▶ 显示网卡的配置信息

用法

- » ifconfig 显示当前已激活（未禁用）网卡配置信息
- » ifconfig -a 显示当前系统中所有网卡的配置信息
- » ifconfig 网卡设备名 显示指定网卡设备的配置信息



**【例7-1】**若要显示当前系统所有网络设备接口信息，则运行命令

```
[root@localhost ~]# ifconfig -a
```

## 命令运行后返回结果如下:

eno16777736: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500

ether 00:0c:29:fa:55:89 txqueuelen 1000 (Ethernet)

RX packets 0 bytes 0 (0.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 0 bytes 0 (0.0 B)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536

inet 127.0.0.1 netmask 255.0.0.0

inet6 ::1 prefixlen 128 scopeid 0x10<host>

loop txqueuelen 0 (Local Loopback)

RX packets 458 bytes 38840 (37.9 KiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 458 bytes 38840 (37.9 KiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

## 命令运行后返回结果如下:

```
virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
  inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
  ether 52:54:00:32:95:52 txqueuelen 0 (Ethernet)
  RX packets 0 bytes 0 (0.0 B)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 0 bytes 0 (0.0 B)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0-nic: flags=4098<BROADCAST,MULTICAST> mtu 1500
  ether 52:54:00:32:95:52 txqueuelen 500 (Ethernet)
  RX packets 0 bytes 0 (0.0 B)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 0 bytes 0 (0.0 B)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

其中eth0表示**第一块网卡设备**；lo表示**环路地址信息**。通过ifconfig命令可以查看到网卡的IP地址、子网掩码、MTU（最大传输单元）、MAC地址、跳数、发送数据包的个数、接收数据包的个数及其错误个数、丢弃个数等信息。

## 1. ifconfig命令

### 功能2 ▶ 为网卡指定临时IP地址

#### 用法

» ifconfig 网卡设备名 IP地址 netmask 子网掩码



#### 说明

通过ifconfig命令添加或修改ip地址只是临时有效。当重启network服务或重启计算机后，之前用ifconfig创建的ip地址将无效。

Linux也支持**IP别名**，即可以在一个网络接口上配置多个IP地址。Linux通过创建虚拟网卡来实现创建IP别名，虚拟网卡的设备名为ethN:M，对应的配置文件为ifcfg-ethN:M。比如：在第一块网卡上创建的第一块虚拟网卡，其设备名为eth0:0，对应的配置文件名为ifcfg-eth0:0；在第一块网卡上创建的第二块虚拟网卡的设备名为eth0:1，其余依次类推，Linux最多支持255个IP别名。虚拟网卡的配置文件可通过更名复制ifcfg-eth0配置文件来获得，然后通过修改配置文件来指定该虚拟网卡要设置的IP地址。

**【例7-2】** 将第一块网卡的ip地址设为192.168.122.200，同时将其子网掩码设为255.255.255.0。  
则使用命令：

```
# ifconfig eno16777736 192.168.122.200 netmask 255.255.255.0
```

运行后再使用ifconfig eno16777736命令验证，结果如下：

```
inet 192.168.122.200 netmask 255.255.255.0 broadcast 192.168.122.255  
ether 00:0c:29:fa:55:89 txqueuelen 1000 (Ethernet)  
RX packets 0 bytes 0 (0.0 B)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 0 bytes 0 (0.0 B)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

## 1. ifconfig命令

### 功能3 ▶ 禁用网卡

#### 用法

» ifconfig 网卡设备名 down

» 或 ifdown 网卡设备名



#### 说明

以上两条命令功能一致，后者用法简洁，使用较多。



**【例7-3】**若要禁用eth0网卡，则实现命令为：

```
[root@localhost ~]#ifconfig eth0 down
```

或

```
[root@localhost ~]#ifdown eth0
```

## 1. ifconfig命令

### 功能4 ▶ 重新启用网卡

#### 用法

» ifconfig 网卡设备名 up

» 或 ifup 网卡设备名



#### 说明

以上两条命令功能一致，后者用法简捷，使用较多。



**【例7-4】**若要启用eth0网卡，则实现命令为：

```
[root@localhost ~]# ifconfig eth0 up
```

或

```
[root@localhost ~]# ifup eth0
```

## 2. netconfig命令

netconfig命令以对话框的方式，配置一个永久ip地址、网关及DNS服务器信息。RHEL7.2默认安装好之后是没有自动开启网络连接的。通过netconfig命令设置的ip信息会直接写到**网卡配置文件** (/etc/sysconfig/network-scripts/ifcfg-eth0) 和**DNS客户端配置文件** (/etc/resolv.conf) 中。但不会立即生效，设置完成后还需重启network服务或重启计算机系统。

**用法** >> **ifconfig 网卡设备名 down**

**【例7-5】**使用netconfig命令以对话框方式为eth0网卡配置ip地址为192.168.3.3，子网掩码为255.255.255.0，网关为192.168.3.1，DNS1为8.8.8.8，DNS2为8.8.4.4。操作命令如下：

```
# netconfig --device eth0
```

设置eth0网卡ip地址，此时的--device可以省略。

## 2. netconfig命令

通过以下步骤完成设置：

### STEP 1

编辑配置文件ifcfg-eno16777736，如图7-1所示添加修改内容。

```
IPADDR0="192.168.3.3"  
PREFIX0="24"  
GATEWAY0="192.168.3.1"  
DNS1="8.8.8.8"  
DNS2="8.8.4.4"
```

图7-1 更改网卡配置文件

### STEP 2

输入service network restart命令重启网络。

可查看/etc/sysconfig/network-scripts/ifcfg-eth0配置文件，验证是否修改成功。

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
```

## 2. netconfig命令

显示结果如图7-2所示。

```
TYPE=Ethernet
BOOTPROTO=dhcp
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6_INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=eno16777736
UUID=27a0fb32-fab5-434d-997b-0ecde3e30682
DEVICE=eno16777736
ONBOOT=no
IPADDR0="192.168.3.3"
PREFIX0="24"
GATEWAY0="192.168.3.1"
DNS1="8.8.8.8"
DNS2="8.8.4.4"
```

图7-2 网卡配置文件

通过以下命令查看/etc/resolv.conf配置文件：

```
# cat /etc/resolv.conf
```

显示结果如下：

```
nameserver 172.16.100.1
```

通过以下命令重启network服务：

```
# service network restart
```

重启network服务后，刚刚设置的静态永久ip地址已生效。可采用ifconfig eth0或ping 172.16.100.98进行查看或测试ip地址的有效性。

### 3. ethtool命令

ethtool是一个查询和设置网卡参数的命令。通常用来查看和修改网卡传输速率、半双工、全双工、网卡连接状态、自适应工作方式等。

**用法** >> **ethtool [参数选项] 网卡设备名**

参数

-l

查询网卡的相关信息

-d

查询网卡网口注册性信息

-r

重置网卡网口到自适应模式

-S

查询网卡网口收发包统计

-s [speed 10|100|1000]

设置网卡速率10/100/1000M

### 【例7-6】查看网卡eth1的相关参数信息。

```
# ethtool eno16777736
Settings for eno16777736:
  Supported ports: [ MII ]
  Supported link modes:  10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Half 1000baseT/Full

  Supports auto-negotiation: Yes
  Advertised link modes:  10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Half 1000baseT/Full

  Advertised auto-negotiation: Yes
  Speed: 100Mb/s           //传输速率为100Mb/s
  Duplex: Full             //工作方式为全双工
  Port: Twisted Pair
  PHYAD: 1
  Transceiver: internal
  Auto-negotiation: on     //自适应打开
  Supports Wake-on: g
  Wake-on: d
  Current message level: 0x000000ff (255)
  Link detected: yes      //网线连接正常，即有网线连接
```

又如, 设置eno167777737网卡, 自适应关闭, 传输速率为10Mb/s, 工作模式为半双工。

```
# ethtool -s eno167777737 autoneg off speed 10 duplex half
# ethtool eno167777737
Settings for eno167777737:
  Supported ports: [ MII ]
  Supported link modes:  10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Half 1000baseT/Full
  Supports auto-negotiation: Yes
  Advertised link modes:  Not reported
  Advertised auto-negotiation: No //自适应关闭
  Speed: 10Mb/s //传输速率为10Mb/s
  Duplex: Half //工作模式为半双工
  Port: Twisted Pair
  PHYAD: 1
  Transceiver: internal
  Auto-negotiation: off
  Supports Wake-on: g
  Wake-on: d
  Current message level: 0x000000ff (255)
  Link detected: yes
```



### 说明

如果为no时, 表示此网卡没有激活、启用或网线没有连接。

## 4. mii-tool命令

mii-tool命令用于查看、管理网卡的网络接口的状态，如网卡的传输速率、半双工、全双工、自适应、网卡是否连接正确等。

**用法** >> **mii-tool [参数选项] 网卡设备名**

参  
数

- V 显示mii-tool命令的版本
- v 显示网络接口的信息
- R 重新设置mii到开启状态
- r 重启自动协商模式
- w 查看网络接口连接的状态变化
- l 写入事件到系统日志
- A 指令特定的网络接口
- F 更改网络接口协商方式，常用的协商方式有：100baseT4，100baseTx-FD，100baseTx-HD，10baseT-FD，10baseT-HD等。

### 【例7-7】查看网络接口的协商状态。

```
# mii-tool -v
eth0: negotiated 100baseTx-FD, link ok
product info: vendor 00:08:18, model 24 rev 0
basic mode: autonegotiation enabled
basic status: autonegotiation complete, link ok
capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
eth1: negotiated 100baseTx-FD flow-control, link ok
product info: vendor 00:08:18, model 24 rev 0
basic mode: autonegotiation enabled
basic status: autonegotiation complete, link ok
capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
# mii-tool
eth0: negotiated 100baseTx-FD, link ok
eth1: negotiated 100baseTx-FD flow-control, link ok
```

又如，更改网络接口的协商方式，在进行更改时，不要超过网卡所支持的协商方式。

```
# mii-tool -v eth1
```

```
eth1: negotiated 100baseTx-FD flow-control, link ok
```

```
product info: vendor 00:08:18, model 24 rev 0
```

```
basic mode: autonegotiation enabled
```

```
basic status: autonegotiation complete, link ok
```

```
capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
```

```
advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
```

```
link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
```

```
# mii-tool -F 100baseTx-FD eth1
```

```
# mii-tool -v eth1
```

```
eth1: 100 Mbit, full duplex, link ok
```

```
product info: vendor 00:08:18, model 24 rev 0
```

```
basic mode: 100 Mbit, full duplex
```

```
basic status: link ok //表示已将网卡的协商方式从自适应设置为100baseTX-FD方式
```

```
capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
```

```
advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
```

## 5. netstat命令

### 用法 » netstat [参数选项]

netstat命令用于数据监听及连接的情况。包括正在监听的服务、已连接的会话、正在结束的会话、服务器的socket连接情况等。

#### 参数

-a

显示所有协议和socket连接信息

-n

显示对应的源ip地址和目的ip地址

-l

显示正在监听的信息

-u

显示UDP协议信息

-t

显示TCP协议信息

-p

显示协议的服务器名称

-r

显示路由表信息

**【例7-8】** 要显示所有正在监听的tcp和udp的服务信息并显示对应的服务器名称。可运行如下命令：

```
# netstat -antupl
```

显示结果如下：

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
tcp    0      0 0.0.0.0:111      0.0.0.0:*    LISTEN 5342/portmap
tcp    0      0 :::22           :::*        LISTEN 3542/sshd
tcp    0 148 ::ffff:172.16.100.98:22 ::ffff:172.16.100.204:1325 ESTABLISHED 6081/1
udp    0      0 0.0.0.0:111      0.0.0.0:*          5342/portmap
```

- **第一列**：显示对应的协议，TCP还是UDP协议。
- **第二列**：接收的数据包。
- **第三列**：发送的数据包。
- **第四列**：本地ip地址及开放的端口。
- **第五列**：远程ip地址及连接的端口。
- **第六列**：现在连接状态，监听（LISTEN）、回应包（ESTABLISHED）、结束包（FIN）、等待包（WAIT）等。
- **第七列**：服务进程ID号（PID）和服务名称（服务守护进程）。

## 6. chkconfig命令

功能1 ▶ 添加、删除对应的系统服务

### 用法

» `chkconfig --add {name}`

» `chkconfig --del {name}`

功能3 ▶ 显示某个服务在各运行级别中的状态。

### 用法

» `chkconfig --list [name]`

功能2 ▶ 设置某个服务在特定的级别中是否随系统启动时启动或停止。

### 用法

» `chkconfig --level {0-6} {name} {on/off/reset}`

也可以采用 `ntsysv --level {12345}` 命令实现此功能。

### 参数

- `--add` 添加某个自定义服务到系统服务中，开机时随系统一起启动。
- `--del` 删除某个自定义服务或系统服务。
- `--list` 显示某个服务，在特定级别的启动状态。
- `--level` 设置某个服务在特定的级别启动或停止。
- `On` 开机时自动运行此服务。
- `Off` 开机时不自动运行此服务。
- `Reset` 恢复此服务在各运行级别的默认状态。

**【例7-9】** 设置sendmail服务在2345启动级别中，不自动运行。

```
# chkconfig --level 2345 sendmail off
```

显示sendmail服务在2345启动级别中的状态。

```
# chkconfig --list sendmail
```

显示结果如下：

```
sendmail 0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

通过chkconfig命令设置完成后。系统启动时，Sendmail服务不会随系统启动时自动运行。其中在设置时，一般不需要对级别0、1、6进行配置。默认情况这3个级别都不会有服务运行。

## 7. nslookup命令

nslookup命令用于检测域名对应的ip地址或者根据ip检测出对应的域名等。

### ● 用法

» nslookup 域名|ip 地址

**【例7-10】 nslookup命令使用。 启动DNS服务：**

```
# service named start
```

**修改DNS客户端配置，具体内容如下：**

```
; generated by /sbin/dhclient-script
```

```
search domain
```

```
nameserver 172.16.100.98
```

//本机ip地址或DNS服务器的ip地址

```
nameserver 202.96.128.166
```

**采用nslookup命令检测localhost域名的对应的ip地址：**

```
# nslookup
```

显示结果如下 (粗体为用户输入内容) :

```
> set type=a //设置A记录信息
> localhost //检测localhost域名的A记录信息
Server: 172.16.100.98
Address: 172.16.100.98#53

Name: localhost
Address: 127.0.0.1
> set type=soa //设置SOA记录信息。
> localhost //检测localhost域名的SOA记录信息。
Server: 172.16.100.98
Address: 172.16.100.98#53
```

## 显示结果如下 (粗体为用户输入内容) :

```
localhost
  origin = localhost
  mail addr = root.localhost
  serial = 42
  refresh = 10800
  retry = 900
  expire = 604800
  minimum = 86400
> set type=ns                //设置NS记录信息。
> localhost                  //检测localhost域名的NS记录信息。
Server:                172.16.100.98
Address:               172.16.100.98#53
localhost nameserver = localhost.
> exit                      //退出nslookup检测
```

以上通过**nslookup**命令检测了localhost的A记录、SOA记录、NS记录信息。其他对应类型有MX（邮件交换器）、CNAME（别名记录）、PTR（指针记录）等。也可采用host或dig命令进行域名检测，具体参数请查阅man手册。

## 本章要点

7.1 网络配置命令

7.2 常用网络配置文件

1

## /etc/nsswitch.conf

**文件功能：**名称服务交换配置 (/etc/nsswitch.conf) 文件。



它是由**Sun公司**开发并用于管理系统中多个配置文件的查找顺序的，是配置文件 /etc/host.conf的**增强版**。它控制了数据库搜寻的工作，包括主机、使用者、群组等。此外，还定义了所要搜寻的**数据库**。如 “hosts: files dns:” 表示指明主机数据库来自两个地方，一个files文件 (/etc/hosts) ，另一个为DNS服务器。



### 说明

/etc/host.conf文件由版本5的libc库所使用，而配置文件/etc/nsswitch.conf由版本6的glibc所使用。在Linux高版本中一般用的是/etc/nsswitch.conf文件。

## 1

**/etc/nsswitch.conf**

/etc/nsswitch.conf每一个关键字都是在/etc/目录可以被/etc/nsswitch.conf控制的文件名称。下面为可以被包含的关键字。

关键字	含义	关键字	含义
<b>aliases:</b>	邮件别名;	<b>shadow:</b>	加密口令;
<b>group:</b>	用户工作组;	<b>networks:</b>	网络名;
<b>hosts:</b>	主机名;	<b>services:</b>	服务名称;
<b>protocols:</b>	网络协议;	<b>rpc:</b>	远程进程调用的名;
<b>ethers:</b>	以太网;	<b>dns:</b>	只对hosts有意义;
<b>netgroup:</b>	网内工作组;	<b>files:</b>	除了netgroup, 对其他关键字都有效。
<b>passwd:</b>	系统用户;		

**1**

## /etc/nsswitch.conf

### /etc/nsswitch.conf配置文件内容如下:

```
passwd:          files
shadow:         files
group:          files
//域名的解析顺序, 先查找/etc/hosts文件, 若没有找到, 再询问DNS服务器
hosts:          files dns
bootparams:     files
ethers:         files
netmasks:       files
networks:       files
protocols:      files
rpc:            files
services:       files
netgroup:       files
publickey:      files
automount:      files
aliases:        files
```

设置DNS的解析顺序。支持的查询有HOSTS文件、BIND服务、NIS服务。默认情况先查询HOSTS文件,再查询DNS服务器。若用户需添加NIS服务查询,可采用“hosts:files,dns,nis”方式添加。

## 2

## /etc/services

### 文件功能:

网络服务配置文件，将网络服务名转换为端口号/协议。文件中的每一行对应一种服务，由4个字段组成。分别表示“服务名称”“使用端口”“协议名称”和“服务说明”。

**/etc/services**配置文件内容如下。

//服务	端口	协议类型	服务说明
.....			
ftp-data	20/tcp		
ftp-data	20/udp		
# 21 is registered to ftp, but also		used by fsp	
ftp		21/tcp	
ftp		21/udp	fsp fspd
ssh		22/tcp	
		# SSH Remote Login Protocol	
ssh		22/udp	
		# SSH Remote Login Protocol	
telnet	23/tcp		
telnet	23/udp		

## 2

## /etc/services

**/etc/services**配置文件内容如下。

```
# 24 - private mail system
lmtptcp 24/tcp
# LMTP Mail Delivery
lmtputp 24/udp
# LMTP Mail Delivery
smtptcp 25/tcp mail
smtpudp 25/udp mail
timetcp 37/tcp timserver
timeudp 37/udp timserver
rlptcp 39/tcp resource
# resource location
rlputp 39/udp resource
.....
```

通过以上内容可知。ftp协议对应着20和21号端口，ssh协议对应着22号端口。若用户将ssh协议对应的22号端口改成其他端口。系统将访问本机22端口，将系统错误或访问不成功。

### 3

## /etc/resolv.conf

### 文件功能:

/etc/resolv.conf为DNS客户端配置文件，当DNS客户端用户需要进行DNS解析时，系统会将查询请求发送到此文件中**nameserver**语句定义的IP地址（DNS服务器的IP地址）上。要求此IP地址帮忙解析。若解析成功。DNS服务器会返回一个值给DNS客户端。同时可以设置要求查询的域名等。

### /etc/resolv.conf文件格式如下:

```
; generated by /sbin/dhclient-script  
search gdLinux.com  
nameserver 192.168.1.1  
domain gdLinux.com
```

### 相关指令解释如下:

“search gdLinux.com”：将搜索域设置为gdLinux.com域。  
“nameserver 192.168.1.1”：设置DNS服务器的IP地址为192.168.1.1。  
“domain gdLinux.com”：定义本地域名

4

`/etc/sysconfig/network`

**文件功能：**

主要用于设置Linux系统的主机名及系统启动时是否加载网卡信息。

**`/etc/sysconfig/network`配置文件内容如下：**

```
NETWORKING=yes
```

```
HOSTNAME=localhost.localdomain
```

**以上相关指令解释如下：**

NETWORKING=yes：表示系统启动时加载network服务。

HOSTNAME=localhost.localdomain：表示设置系统的主机名为localhost.localdomain。

# 5

## /etc/sysconfig/network-scripts/ifcfg-eth0

### 文件功能:

网卡信息配置文件，包括网络接口设备、协议类型（静态、动态）、IP地址、子网掩码、网关、DNS服务器等。可以通过netconfig命令进行设置，也可直接编辑此配置文件进行更改。

**/etc/sysconfig/network-scripts/ifcfg-eth0 (静态IP设置) 内容如下:**

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=172.16.100.98
NETMASK=255.255.255.0
GATEWAY=172.16.100.254
```

**/etc/sysconfig/network-scripts/ifcfg-eth0配置  
文件 (动态IP设置) 内容如下:**

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
```

## 5

# /etc/sysconfig/network-scripts/ifcfg-eth0

### 以上相关语句解释如下：

DEVICE=eth0：设置网络接口设备为eth0。若为第二块网卡设置成eth1、第三块网卡设置成eth2、设备子接口设置成DEVICE=eth0:1等。

ONBOOT：系统开机时是否加载本网卡信息。

BOOTPROTO：引导协议类型。其中static表示采用静态IP地址；dhcp表示采用动态IP地址。由dhcp服务器进行动态IP地址分配。

IPADDR：设置网卡的IP地址。

NETMASK：设置IP地址的子网掩码。

GATEWAY：设置网关IP地址。



A nighttime cityscape featuring various buildings, including a prominent one with a tall spire and a star on top. The scene is dimly lit with some lights from the buildings. A large, bright blue speech bubble shape is positioned at the bottom of the image, containing the text.

谢谢观看