

第9章 Linux磁盘管理

学习Linux请加QQ群： 群1(163262181) 群2(148412746) 群3(246401509) 群4(173884211)
跟阿铭学Linux邀请函 (<http://www.aminglinux.com>)，猿课已上线，请加微信aminglinux84索要配套视频教程。
在日常的Linux管理工作中，这部分内容使用还是比较多的。

查看磁盘或者目录的容量

命令：df

“df” 查看已挂载磁盘的总容量、使用容量、剩余容量等，可以不加任何参数，默认是按k为单位显示的。

| | | | | | |
|------------------------|----------|---------|----------|-----|----------|
| [root@localhost ~]# df | | | | | |
| 文件系统 | 1K-块 | 已用 | 可用 | 已用% | 挂载点 |
| /dev/sda3 | 14347632 | 1490876 | 12127924 | 11% | / |
| tmpfs | 163308 | 0 | 163308 | 0% | /dev/shm |
| /dev/sda1 | 99150 | 26808 | 67222 | 29% | /boot |

“df” 常用选项有 “-i” “-h” “-k” “-m”等

“-i” 查看inodes使用状况

| | | | | | |
|-----------|--------|--------|--------|----------|----------|
| 文件系统 | Inode | 已用 (I) | 可用 (I) | 已用 (I) % | 挂载点 |
| /dev/sda3 | 912128 | 66195 | 845933 | 8% | / |
| tmpfs | 40827 | 1 | 40826 | 1% | /dev/shm |
| /dev/sda1 | 25688 | 38 | 25650 | 1% | /boot |

“-h” 使用合适的单位显示，例如 ‘G’

| | | | | | |
|-----------|------|------|------|------|----------|
| 文件系统 | 容量 | 已用 | 可用 | 已用 % | 挂载点 |
| /dev/sda3 | 14G | 1.5G | 12G | 11% | / |
| tmpfs | 160M | 0 | 160M | 0% | /dev/shm |
| /dev/sda1 | 97M | 27M | 66M | 29% | /boot |

“-k”, “-m” 分别以K, M 为单位显示

| | | | | | |
|---------------------------|----------|---------|----------|-----|----------|
| [root@localhost ~]# df -k | | | | | |
| 文件系统 | 1K-块 | 已用 | 可用 | 已用% | 挂载点 |
| /dev/sda3 | 14347632 | 1490880 | 12127920 | 11% | / |
| tmpfs | 163308 | 0 | 163308 | 0% | /dev/shm |
| /dev/sda1 | 99150 | 26808 | 67222 | 29% | /boot |
| [root@localhost ~]# df -m | | | | | |
| 文件系统 | 1M-块 | 已用 | 可用 | 已用% | 挂载点 |
| /dev/sda3 | 14012 | 1456 | 11844 | 11% | / |
| tmpfs | 160 | 0 | 160 | 0% | /dev/shm |
| /dev/sda1 | 97 | 27 | 66 | 29% | /boot |

简单介绍一下各列所表示的含义，其实如果你的Linux和阿铭的虚拟机一样也是中文显示的话，那么不用说太多，看字面意思就明白了。第一列是分区的名字，第二列为该分区总共的容量，第三列为已经使用了多少，第四列为还剩下多少，第五列为已经使用百分比，如果这个数值到达90%以上，那么你就应该关注了，磁盘分区满了可不是什么好事情，会引起系统崩溃的。最后一列为挂载点，你是否还记得，阿铭在装系统的时候，有说到这个词，“/dev/shm” 为内存挂载点，如果你想把文件放到内存里，就可以放到/dev/shm/目录下。

命令：du

“du” 用来查看某个目录或文件所占空间大小。

语法：du [-abckmsh] [文件或者目录名] 常用的参数有：

“-a” 全部文件与目录大小都列出来。如果不加任何选项和参数只列出目录（包含子目录）大小。

| | |
|--------------------------------|--|
| [root@localhost ~]# du dirb | |
| 4 dirb/dirc | |
| 12 dirb | |
| [root@localhost ~]# du -a dirb | |
| 4 dirb/filee | |
| 4 dirb/dirc | |
| 12 dirb | |

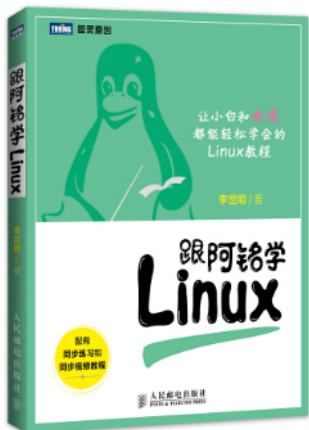
如果du不指定单位的话，默认显示单位为K。

“-b” 列出的值以bytes为单位输出。

目录列表

| |
|------------------------------|
| 第1章 前言 |
| 第2章 关于Linux的历史 |
| 第3章 对Linux系统管理员的建议 |
| 第4章 安装Linux操作系统 |
| 第5章 初步认识Linux |
| 第6章 Linux系统的远程登陆 |
| 第7章 Linux文件与目录管理 |
| 第8章 Linux系统用户及用户组管理 |
| 第9章 Linux磁盘管理 |
| 第10章 文本编辑工具vim |
| 第11章 文档的压缩与打包 |
| 第12章 安装RPM包或者安装源码包 |
| 第13章 学习 shell脚本之前的基础知识 |
| 第14章 正则表达式 |
| 第15章 shell脚本 |
| 第16章 linux系统日常管理 |
| 第17章 LAMP环境搭建 |
| 第18章 LNMP环境搭建 |
| 第19章 学会使用简单的MySQL操作 |
| 第20章 NFS服务配置 |
| 第21章 配置FTP服务 |
| 第22章 配置Squid服务 |
| 第23章 配置Tomcat |
| 第24章 配置Samba服务器 |
| 第25章 MySQL replication(主从)配置 |
| 结语 |

阿铭著作：



微信扫码获取最新版linux电子书和视频

"-k" 以KB为单位输出，和默认不加任何选项的输出值是一样的。

"-m" 以MB为单位输出

"-h" 系统自动调节单位，例如文件太小可能就几K，那么就以K为单位显示，如果大到几G，则就以G为单位显示。

```
[root@localhost ~]# du -b /etc/passwd
1181    /etc/passwd
[root@localhost ~]# du -k /etc/passwd
4      /etc/passwd
[root@localhost ~]# du -m /etc/passwd
1      /etc/passwd
[root@localhost ~]# du -h /etc/passwd
4.0K   /etc/passwd
```

"-c" 最后加总

```
[root@localhost ~]# du -c dirb
4      dirb/dirb
12     dirb
12     总用量
[root@localhost ~]# du dirb
4      dirb/dirb
12     dirb
```

"-s" 只列出总和

```
[root@localhost ~]# du -s dirb
12     dirb
```

阿铭习惯用 `du -sh filename` 这样的形式。

磁盘的分区和格式化

阿铭经常做的事情就是拿一个全新的磁盘来分区并格式化。这也说明了作为一个linux系统管理员，对于磁盘的操作必须要熟练。所以请你认真学习该部分内容。在正式介绍Linux下分区工具之前，阿铭需要先给虚拟机添加一块磁盘，以便于我们做后续的实验，如果你也是使用vmware 虚拟机，请跟着阿铭一起来做吧。

1. 先关闭正在运行的Linux系统 `init 0`。
2. 到vmware的Linux虚拟机界面，点 “Edit virtual machine settings”，点一下左侧靠下面的 “Add...” 按钮。
3. 在左侧选中 “Hard Disk” 默认就是这一行，点右下角的 “Next”，继续点 “Next”。
4. “Virtual disk type” 选择 IDE, 点 “Next”
5. 继续点 “Next”, “Disk size” 默认即可，最后点 “Finish”。

命令 : fdisk

fdisk 是Linux下硬盘的分区工具，是一个非常实用的命令，但是fdisk只能划分小于2T的分区。

语法：`fdisk [-l] [设备名称]` 选项只有一个。

"-l" 后边不跟设备名会直接列出系统中所有的磁盘设备以及分区表，加上设备名会列出该设备的分区表。

```
[root@localhost ~]# fdisk -l

Disk /dev/sda: 17.2 GB, 17179869184 bytes
255 heads, 63 sectors/track, 2088 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: 0x00018d63

   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         274       2097152    82  Linux swap / Solaris
Partition 2 does not end on cylinder boundary.
/dev/sda3          274        2089       14576640    83  Linux

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0x00000000

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

SEARCH

Go

Enter search terms or a module, class or function name.

```
Disk /dev/sda: 17.2 GB, 17179869184 bytes
255 heads, 63 sectors/track, 2088 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: 0x00018d63

    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          274       2097152   82  Linux swap / Solaris
Partition 2 does not end on cylinder boundary.
/dev/sda3          274         2089       14576640   83  Linux
```

可以看到刚才阿铭加的一块磁盘 **/dev/sdb** 的信息。

“fdisk” 如果不加 **“-l”** 则进入另一个模式，在该模式下，可以对磁盘进行分区操作。

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

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

Command (m for help):
```

如果你输入 **‘m’** 会列出常用的命令：

```
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)
```

如果你的英文好，我想你不理解这些字母的功能。阿铭常用的有**‘p’**、**‘n’**、**‘d’**、**‘w’**、**‘q’**。

“p” 打印当前磁盘的分区情况。

```
Command (m for help): p

Disk /dev/sda: 17.2 GB, 17179869184 bytes
255 heads, 63 sectors/track, 2088 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: 0x00018d63

    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          274       2097152   82  Linux swap / Solaris
Partition 2 does not end on cylinder boundary.
/dev/sda3          274         2089       14576640   83  Linux
```

- ‘n’** 建立一个新的分区。
- ‘w’** 保存操作。
- ‘q’** 退出。
- ‘d’** 删除一个分区

下面阿铭会把刚才增加的磁盘**/dev/sdb**进行分区操作。先使用 **‘p’** 命令看一下**/dev/sdb**的分区状况：

```
[root@localhost ~]# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0xf4121235.
```

```
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

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

Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0xf4121235

    Device Boot      Start         End      Blocks   Id  System

```

| Device | Boot | Start | End | Blocks | Id | System |
|-----------|------|-------|-----|----------|----|--------|
| /dev/sdb1 | | 1 | 128 | 1028128+ | 83 | Linux |

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

可以看到目前/dev/sdb没有任何分区，下面阿铭给它建立第一个分区：

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
```

使用`n`命令新建分区，它会提示是要`e`（扩展分区）还是`p`（主分区）[\[1\]](#) 阿铭的选择是`p`，于是输入`p`然后回车

```
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1044, default 1): 1
Last cylinder, +cylinders or +size{K,M,G} (1-1044, default 1044): +1000M
```

输入`p`后，会提示分区数，这里阿铭写`1`，因为是第一个分区，当然你也可以写`2`或`3`；如果你直接回车的话，会继续提示你必须输入一个数字，接着又提示第一个柱面从哪里开始，默认是`1`，你可以写一个其他的数字，不过这样就浪费了空间，所以还是写`1`吧，或者你直接回车也会按`1`处理，接着是让输入最后一个柱面的数值，也就是说你需要给这个分区分多大空间，关于柱面是多大阿铭不再细究，你只需要掌握阿铭教给你的方法即可，即写`+1000M`，这样即方便又不容易出错。用`p`查看已经多出了一个分区：

```
Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0x0600660a

    Device Boot      Start         End      Blocks   Id  System
/dev/sdb1           1         128     1028128+   83  Linux
```

继续上面的操作，一直创建主分区到4，然后再一次创建分区的时候则会提示：

```
Command (m for help): n
You must delete some partition and add an extended partition first
```

这是因为，在linux中最多只能创建4个主分区，那如果你想多创建几个分区如何做？很容易，在创建完第三个分区后，创建第四个分区时选择扩展分区。

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
e
Selected partition 4
First cylinder (385-1044, default 385):
Using default value 385
Last cylinder, +cylinders or +size{K,M,G} (385-1044, default 1044):
Using default value 1044
```

```
Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0xef267349
```

| Device | Boot | Start | End | Blocks | Id | System |
|-----------|------|-------|------|----------|----|----------|
| /dev/sdb1 | | 1 | 128 | 1028128+ | 83 | Linux |
| /dev/sdb2 | | 129 | 256 | 1028160 | 83 | Linux |
| /dev/sdb3 | | 257 | 384 | 1028160 | 83 | Linux |
| /dev/sdb4 | | 385 | 1044 | 5301450 | 5 | Extended |

扩展分区，在最后一列显示为 “Extended”，接下来继续创建分区：

```
Command (m for help): n
First cylinder (385-1044, default 385):
Using default value 385
Last cylinder, +cylinders or +size{K,M,G} (385-1044, default 1044): +1000M

Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0xef267349
```

| Device | Boot | Start | End | Blocks | Id | System |
|-----------|------|-------|------|----------|----|----------|
| /dev/sdb1 | | 1 | 128 | 1028128+ | 83 | Linux |
| /dev/sdb2 | | 129 | 256 | 1028160 | 83 | Linux |
| /dev/sdb3 | | 257 | 384 | 1028160 | 83 | Linux |
| /dev/sdb4 | | 385 | 1044 | 5301450 | 5 | Extended |
| /dev/sdb5 | | 385 | 512 | 1028128+ | 83 | Linux |

这时候再分区和以前有区别了，不再选择是主分区还是扩展分区了，而是直接定义大小。有一点阿铭要讲一下，当分完三个主分区后，第四个扩展分区需要把剩余的磁盘空间全部划分给扩展分区，不然的话剩余的空间会浪费，因为分完扩展分区后，再划分新的分区时是在已经划分的扩展分区里来分的。其中/dev/sdb4为扩展分区，这个分区是不可以格式化的，你可以把它看成是一个空壳子，能使用的为/dev/sdb5，其中/dev/sdb5为/dev/sdb4的子分区，这个子分区叫做逻辑分区。如果你发现分区分的不合适，想删除掉某个分区怎么办？这就用到了 ‘d’ 命令：

```
Command (m for help): d
Partition number (1-5): 1

Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0x7b9a6af3
```

| Device | Boot | Start | End | Blocks | Id | System |
|-----------|------|-------|------|----------|----|----------|
| /dev/sdb2 | | 129 | 256 | 1028160 | 83 | Linux |
| /dev/sdb3 | | 257 | 384 | 1028160 | 83 | Linux |
| /dev/sdb4 | | 385 | 1044 | 5301450 | 5 | Extended |
| /dev/sdb5 | | 385 | 512 | 1028128+ | 83 | Linux |

```
Command (m for help): d
Partition number (1-5): 5

Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0x7b9a6af3
```

```

    Device Boot      Start   End  Blocks  Id System
/dev/sdb2            129     256    1028160  83  Linux
/dev/sdb3            257     384    1028160  83  Linux
/dev/sdb4            385    1044    5301450   5  Extended

Command (m for help): n
Command action
    l    logical (5 or over)
    p    primary partition (1-4)
l
First cylinder (385-1044, default 385):
Using default value 385
Last cylinder, +cylinders or +size{K,M,G} (385-1044, default 1044): +1000M
```

```

Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0x7b9a6af3
```

| Device | Boot | Start | End | Blocks | Id | System |
|-----------|------|-------|------|----------|----|----------|
| /dev/sdb2 | | 129 | 256 | 1028160 | 83 | Linux |
| /dev/sdb3 | | 257 | 384 | 1028160 | 83 | Linux |
| /dev/sdb4 | | 385 | 1044 | 5301450 | 5 | Extended |
| /dev/sdb5 | | 385 | 512 | 1028128+ | 83 | Linux |

```

Command (m for help): d
Partition number (1-5): 4
```

```

Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0x7b9a6af3
```

| Device | Boot | Start | End | Blocks | Id | System |
|-----------|------|-------|-----|---------|----|--------|
| /dev/sdb2 | | 129 | 256 | 1028160 | 83 | Linux |
| /dev/sdb3 | | 257 | 384 | 1028160 | 83 | Linux |

输入 'd' 会提示要删除哪个分区，可以选择从 1-5 其中1-3是主分区(sdb1, sdb2, sdb3)，4是扩展分区(sdb4)，5是逻辑分区 [\[1\]](#) (sdb5)，如果输入5，则直接把逻辑分区sdb5删除掉，但是如果输入4的话，会把整个扩展分区sdb4干掉，当然也包含扩展分区里面的逻辑分区sdb5。在刚才的分区界面直接 Ctrl + C 退出来，这样刚刚的分区全部都取消了，咱们重新来做分区：

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

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

Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0x7b9a6af3

    Device Boot      Start   End  Blocks  Id System

Command (m for help): n
Command action
    e    extended
    p    primary partition (1-4)
e
Partition number (1-4): 1
First cylinder (1-1044, default 1): 1
```

Last cylinder, +cylinders or +size{K,M,G} (1-1044, default 1044): 1044

Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0x7b9a6af3

| Device | Boot | Start | End | Blocks | Id | System |
|-----------|------|-------|------|----------|----|----------|
| /dev/sdb1 | | 1 | 1044 | 8385898+ | 5 | Extended |

Command (m for help): n

Command action

- l logical (5 or over)
- p primary partition (1-4)

如果把第一个分区分为扩展分区，并且把全部空间都分给扩展分区的话，再继续分区的话，会提示的分区类型为主分区还是逻辑分区(logical)，用 'l' 表示逻辑分区，逻辑分区的id是从5开始的，因为前四个id为主分区或者扩展分区。既然阿铭把所有磁盘空间都分为了扩展分区，如果你在这里选择 'p' 则会报错：

Command action

- l logical (5 or over)
- p primary partition (1-4)

p

Partition number (1-4): 2

No free sectors available

这是因为没有足够空间分给主分区了，那我们就分逻辑分区：

Command (m for help): n

Command action

- l logical (5 or over)
- p primary partition (1-4)

l

First cylinder (1-1044, default 1): 1

Last cylinder, +cylinders or +size{K,M,G} (1-1044, default 1044): +1000M

Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0x7b9a6af3

| Device | Boot | Start | End | Blocks | Id | System |
|-----------|------|-------|------|----------|----|----------|
| /dev/sdb1 | | 1 | 1044 | 8385898+ | 5 | Extended |
| /dev/sdb5 | | 1 | 128 | 1028097 | 83 | Linux |

Command (m for help): n

Command action

- l logical (5 or over)
- p primary partition (1-4)

l

First cylinder (129-1044, default 129): 129

Last cylinder, +cylinders or +size{K,M,G} (129-1044, default 1044): +1000M

Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0x7b9a6af3

| Device | Boot | Start | End | Blocks | Id | System |
|-----------|------|-------|------|----------|----|----------|
| /dev/sdb1 | | 1 | 1044 | 8385898+ | 5 | Extended |
| /dev/sdb5 | | 1 | 128 | 1028097 | 83 | Linux |
| /dev/sdb6 | | 129 | 256 | 1028128+ | 83 | Linux |

分区完后，需要输入 `w` 命令来保存我们的配置：

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

再使用 `fdisk -l /dev/sdb` 查看分区情况：

```
[root@localhost ~]# fdisk -l /dev/sdb

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0x7b9a6af3


   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1             1         1044     8385898+    5  Extended
/dev/sdb5             1          128     1028097    83   Linux
/dev/sdb6           129          256     1028128+    83   Linux
```

通过以上操作，相信你也学会了用**fdisk**来分区了吧。但阿铭要提醒你，不要闲着没事分区玩儿，这操作的危险性是很高的，一不留神就把服务器上的数据全部给分没有了。所以在你执行分区操作的时候，请保持百分之二百的细心，切记切记！

格式化磁盘分区

命令：mke2fs, mkfs.ext2, mkfs.ext3, mkfs.ext4

当用**man**查询这四个命令的帮助文档时，你会发现我们看到了同一个帮助文档，这说明四个命令是一样的。**mke2fs**常用的选项有：

`-b` 分区时设定每个数据区块占用空间大小，目前支持**1024, 2048** 以及**4096 bytes**每个块。

`-i` 设定**inode**的大小

`-N` 设定**inode**数量，有时使用默认的**inode**数不够用，所以要自定设定**inode**数量。

`-c` 在格式化前先检测一下磁盘是否有问题，加上这个选项后会非常慢

`-L` 预设该分区的标签**label**

`-j` 建立**ext3**格式的分区，如果使用**mkfs.ext3** 就不用加这个选项了

`-t` 用来指定什么类型的文件系统，可以是**ext2, ext3** 也可以是 **ext4**。

```
[root@localhost ~]# mke2fs -t ext4 /dev/sdb5
mke2fs 1.41.12 (17-May-2010)
文件系统标签=
操作系统:Linux
块大小=4096 (log=2)
分块大小=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
64256 inodes, 257024 blocks
12851 blocks (5.00%) reserved for the super user
第一个数据块=0
Maximum filesystem blocks=264241152
8 block groups
32768 blocks per group, 32768 fragments per group
8032 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

正在写入inode表: 完成
Creating journal (4096 blocks): 完成
Writing superblocks and filesystem accounting information: 完成

This filesystem will be automatically checked every 24 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

指定文件系统格式为**ext4**，该命令等同于 `mkfs.ext4 /dev/sdb5`。目前**CentOS 6** 默认文件系统格式为**ext4**，所以以后你遇到需要格式磁盘分区的时候，直接指定格式为**ext4**即可，但早期的版本**CentOS 5** 是使用**ext3**作为默认的文件系统的，所以你可以根据操作系统的版本来决定格式化什么格式的文件系统。在上面的例子中，你是否有注意到一些指标呢？其中一个指标是“块大小=4096”这里涉及到一个“块”的概念，磁盘在被格式化的时候会预先规定好每一个块的大小，然后再把所有的空间分割成一个小块，存数据的时候也是一个块一个块的去写入。所以如果你的磁盘存的都是特别小特别小的文件，比如说**1k**或者**2k**，那么建议在格式化磁盘的时候指定块数值小一点。**ext**文件系统默认块大小为**4096**也就是**4k**。 在格式化的时候，可以指定

块大小为**1024**，**2048**，**4096**(它们是成倍增加的)，虽然格式化的时候可以指定块大小超过**4096**，但是一旦超过**4096**则不能正常挂载，如何指定块大小？

```
[root@localhost ~]# mke2fs -t ext4 -b 8192 /dev/sdb5
Warning: blocksize 8192 not usable on most systems.
mke2fs 1.41.12 (17-May-2010)
mke2fs: 8192-byte blocks too big for system (max 4096)
无论如何也要继续? (y,n) y
Warning: 8192-byte blocks too big for system (max 4096), forced to continue
文件系统标签=
操作系统:Linux
块大小=8192 (log=3)
分块大小=8192 (log=3)
Stride=0 blocks, Stripe width=0 blocks
64256 inodes, 128512 blocks
6425 blocks (5.00%) reserved for the super user
第一个数据块=0
Maximum filesystem blocks=134201344
2 block groups
65528 blocks per group, 65528 fragments per group
32128 inodes per group
Superblock backups stored on blocks:
    65528

正在写入inode表: 完成
Creating journal (4096 blocks): 完成
Writing superblocks and filesystem accounting information: 完成

This filesystem will be automatically checked every 28 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

指定块大小为**8192**会提示，块值设置太大了，我们直接输入 **‘y’** 强制格式化，你还可以尝试指定更大的数字。

```
[root@localhost ~]# mke2fs -t ext4 -L TEST -b 8192 /dev/sdb5
```

可以使用 **‘-L’** 来指定标签。标签会在挂载磁盘的时候使用，另外也可以写到配置文件里，稍后阿铭介绍。关于格式化的这一部分，阿铭建议你除非有需求，否则不需要指定块大小，也就是说，你只需要记住这两个选项：**‘-t’** 和 **‘-L’** 即可。

命令：e2label

用来查看或修改分区的标签，阿铭很少使用，你只要了解一下即可。

```
[root@localhost ~]# e2label /dev/sdb5
TEST
[root@localhost ~]# e2label /dev/sdb5 TEST123
[root@localhost ~]# e2label /dev/sdb5
TEST123
```

挂载/卸载磁盘

在上面的内容中讲到了磁盘的分区和格式化，那么格式化完了后，如何去用它呢？这就涉及到了挂载这块磁盘。格式化后的磁盘其实是一个块设备文件，类型为**b**，也许你会想，既然这个块文件就是那个分区，那么直接在那个文件中写数据不就写到了那个分区中么？当然不行。

在挂载某个分区前需要先建立一个挂载点，这个挂载点是以目录的形式出现的。一旦把某一个分区挂载到了这个挂载点（目录）下，那么再往这个目录写数据使，则都会写到该分区中。这就需要你注意一下，在挂载该分区前，挂载点（目录）下必须是个空目录。其实目录不为空并不影响所挂载分区的使用，但是一旦挂载上了，那么该目录下以前的东西就不能看到了。只有卸载掉该分区后才能看到。

命令：mount

如果不加任何选项，直接运行**“mount”** 命令，会显示如下信息：

```
[root@localhost ~]# mount
/dev/sda3 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)
/dev/sda1 on /boot type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
```

这个命令可以查看当前系统已经挂载的所有分区，以及分区文件系统的类型，挂载点和一些选项等信息，所以你如果想知道某个分区的文件系统类型直接用该命令查看即可。下面我们先建立一个空目录，然后在目录里建一个空白文档。

```
[root@localhost ~]# mkdir /newdir
[root@localhost ~]# touch /newdir/newfile.txt
[root@localhost ~]# ls /newdir/newfile.txt
/newdir/newfile.txt
```

然后把刚才格式化的 `/dev/sdb5` 挂载到 `/newdir` 上。

```
[root@localhost ~]# mount /dev/sdb5 /newdir/
mount: wrong fs type, bad option, bad superblock on /dev/sdb5,
       missing codepage or helper program, or other error
       In some cases useful info is found in syslog - try
       dmesg | tail  or so
```

不能完成挂载，根据提示可以查看一下错误信息：

```
[root@localhost ~]# dmesg |tail
eth0: no IPv6 routers present
sdb: sdb1 < sdb5 >
sdb:
sdb: sdb1 < sdb5 sdb6 >
EXT4-fs (sdb5): bad block size 8192
EXT4-fs (sdb5): bad block size 8192
EXT4-fs (sdb5): bad block size 8192
EXT4-fs (sdb5): bad block size 8192
EXT4-fs (sdb5): mounted filesystem with ordered data mode. Opts:
EXT4-fs (sdb5): bad block size 8192
```

可以看到，我的 `/dev/sdb5` 指定的块值 **8192** 不合法，所以只能重新格式化磁盘。

```
[root@localhost ~]# mke2fs -t ext4 -L TEST /dev/sdb5
```

使用默认块值即可。我们继续挂载 `sdb5`：

```
[root@localhost ~]# mount /dev/sdb5 /newdir/
[root@localhost ~]# ls /newdir/
lost+found
[root@localhost ~]# df -h
文件系统          容量  已用  可用  已用%% 挂载点
/dev/sda3          14G   1.5G   12G   11% /
tmpfs              160M    0   160M    0% /dev/shm
/dev/sda1           97M   27M   66M   29% /boot
/dev/sdb5          989M   18M   921M    2% /newdir
```

把 `/dev/sdb5` 挂载到 `/newdir` 后，原来在 `/newdir` 下的 `newfile.txt` 被覆盖了，通过 `df -h` 可以看到刚刚挂载的分区，我们也可以使用 **LABEL** 的方式挂载分区：

```
[root@localhost ~]# umount /newdir/
[root@localhost ~]# df -h
文件系统          容量  已用  可用  已用%% 挂载点
/dev/sda3          14G   1.5G   12G   11% /
tmpfs              160M    0   160M    0% /dev/shm
/dev/sda1           97M   27M   66M   29% /boot
[root@localhost ~]# mount LABEL=TEST /newdir
[root@localhost ~]# df -h
文件系统          容量  已用  可用  已用%% 挂载点
/dev/sda3          14G   1.5G   12G   11% /
tmpfs              160M    0   160M    0% /dev/shm
/dev/sda1           97M   27M   66M   29% /boot
/dev/sdb5          989M   18M   921M    2% /newdir
```

本例中用到了 “**umount**” 命令，这个是用来卸载磁盘分区的，稍后阿铭介绍。**mount** 命令常用的选项有：‘**-a**’，‘**-t**’，‘**-o**’。在讲 ‘**-a**’ 选项前，我们有必要先了解一下这个文件 `/etc/fstab`。

```
[root@localhost ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Tue May  7 17:51:27 2013
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=95297b81-538d-4d96-870a-de90255b74f5 /          ext4    defaults    1 1
UUID=a593ff68-2db7-4371-8d8c-d936898e9ac9 /boot      ext4    defaults    1 2
UUID=ff042a91-b68f-4d64-9759-050c51dc9e8b swap       swap    defaults    0 0
tmpfs /dev/shm tmpfs    defaults    0 0
```

| | | | | |
|--------|----------|--------|----------------|-----|
| devpts | /dev/pts | devpts | gid=5,mode=620 | 0 0 |
| sysfs | /sys | sysfs | defaults | 0 0 |
| proc | /proc | proc | defaults | 0 0 |

这个文件是系统启动时，需要挂载的各个分区。第一列就是分区的标识，可以写分区的LABEL，也可以写分区的UUID(等会阿铭会着重讲一下这个概念)，当然也可以写分区名(/dev/sda1)；第二列是挂载点；第三列是分区的格式；第四列则是mount的一些挂载参数，等下会详细介绍一下有哪些参数，一般情况下，直接写defaults即可；第五列的数字表示是否被dump备份，是的话这里就是1，否则就是0；第六列是开机时是否自检磁盘。1，2都表示检测，0表示不检测，在Redhat/CentOS中，这个1，2还有个说法，/分区必须设为1，而且整个fstab中只允许出现一个1，这里有一个优先级的说法。1比2优先级高，所以先检测1，然后再检测2，如果有多个分区需要开机检测那么都设置成2吧，1检测完了后会同时去检测2。下面该说说第四列中常用到的参数了。

“async/sync”：async表示和磁盘和内存不同步，系统每隔一段时间把内存数据写入磁盘中，而sync则会时时同步内存和磁盘中数据；

“auto/noauto”：开机自动挂载/不自动挂载；

“default”：按照大多数永久文件系统的缺省值设置挂载定义，它包含了rw, suid, dev, exec, auto, nouser, async

“ro”：按只读权限挂载；

“rw”：按可读可写权限挂载；

“exec/noexec”：允许/不允许可执行文件执行，但千万不要把根分区挂载为noexec，那就无法使用系统了，连mount命令都无法使用了，这时只有重新做系统了；

“user/nouser”：允许/不允许root外的其他用户挂载分区，为了安全考虑，请用nouser；

“suid/nosuid”：允许/不允许分区有suid属性，一般设置nosuid；

“usrquota”：启动使用者磁盘配额模式，磁盘配额相关内容在后续章节会做介绍；

“grquota”：启动群组磁盘配额模式；

学完这个/etc/fstab后，我们就可以自己修改这个文件，增加一行来挂载新增分区。例如，阿铭增加了这样一行：

| | | | | |
|------------|---------|------|----------|-----|
| LABEL=TEST | /newdir | ext4 | defaults | 0 0 |
|------------|---------|------|----------|-----|

然后卸载掉刚才我们已经挂载的/dev/sdb5

```
[root@localhost ~]# umount /dev/sdb5
[root@localhost ~]# df -h
文件系统      容量  已用  可用  已用%% 挂载点
/dev/sda3      14G   1.5G   12G   11% /
tmpfs          160M    0   160M    0% /dev/shm
/dev/sda1       97M   27M   66M   29% /boot
```

使用 df -h 查看已经成功卸载 /dev/sdb5 下面执行命令 mount -a

```
[root@localhost ~]# mount -a
[root@localhost ~]# df -h
文件系统      容量  已用  可用  已用%% 挂载点
/dev/sda3      14G   1.5G   12G   11% /
tmpfs          160M    0   160M    0% /dev/shm
/dev/sda1       97M   27M   66M   29% /boot
/dev/sdb5      989M   18M   921M    2% /newdir
```

此时，多出来一个 /dev/sdb5 挂载到了 /newfir 下。这就是 mount -a 命令执行的结果，这个 ‘-a’ 选项会把/etc/fstab中出现的所有磁盘分区挂载上。

```
[root@localhost ~]# umount /newdir
[root@localhost ~]# mount -t ext4 /dev/sdb5 /newdir
[root@localhost ~]# df -h
文件系统      容量  已用  可用  已用%% 挂载点
/dev/sda3      14G   1.5G   12G   11% /
tmpfs          160M    0   160M    0% /dev/shm
/dev/sda1       97M   27M   66M   29% /boot
/dev/sdb5      989M   18M   921M    2% /newdir
```

‘-t’ 选项用来指定挂载的分区类型，默认不指定会自动识别。

‘-o’ 选项用来指定挂载的分区有哪些特性，即上面 “/etc/fatab” 配置文件中第四列的那些。阿铭经常这样使用这个 ‘-o’ 选项：

```
[root@localhost ~]# mkdir /newdir/dir1
[root@localhost ~]# mount -o remount,ro,sync,noauto /dev/sdb5 /newdir
[root@localhost ~]# mkdir /newdir/dir2
mkdir: 无法创建目录 "/newdir/dir2": 只读文件系统
```

由于指定了 ‘ro’ 参数，所以该分区只读了。通过 mount 命令也可以看到 /dev/sdb5 有 ‘ro’ 选项

```
[root@localhost ~]# mount
/dev/sda3 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)
/dev/sda1 on /boot type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
/dev/sdb5 on /newdir type ext4 (ro,async)
```

下面阿铭重新挂载，让它恢复读写。

```
[root@localhost ~]# mount -o remount /dev/sdb5 /newdir
[root@localhost ~]# mkdir /newdir/dir2
[root@localhost ~]# ls /newdir/
dir1 dir2 lost+found
```

命令：**blkid**

阿铭在日常的运维工作中遇到过这样的情况，一台服务器上新装了两块磁盘，磁盘**a**（在服务器上显示为**sdc**）和磁盘**b**（在服务器上显示为**sdd**），有一次把这两块磁盘都拔掉了，然后再重新插上，重启机器，结果磁盘编号调换了，**a**变成了**sdd**，**b**变成了**sdc**（这是因为把磁盘插错了插槽），问题来了。通过上边的学习，你挂载磁盘是通过**/dev/hdb1** 这样的分区名字来挂载的，如果先前加入到了**/etc/fstab** 中，结果系统启动后则会挂载错分区。那么怎么样避免这样的情况发生？

这就用到了**UUID**，可以通过 **blkid** 命令获取各分区的**UUID**：

```
/dev/sda1: UUID="a593ff68-2db7-4371-8d8c-d936898e9ac9" TYPE="ext4"
/dev/sda2: UUID="ff042a91-b68f-4d64-9759-050c51dc9e8b" TYPE="swap"
/dev/sda3: UUID="95297b81-538d-4d96-870a-de90255b74f5" TYPE="ext4"
/dev/sdb5: LABEL="TEST" UUID="c61117ca-9176-4d0b-be4d-1b0f434359a7" TYPE="ext4"
/dev/sdb6: UUID="c271cb5a-cb46-42f4-9eb4-d2b1a5028e18" SEC_TYPE="ext2" TYPE="ext3"
```

这样可以获得全部磁盘分区的**UUID**，如果格式化的时候指定了 **LABEL** 则该命令也会显示**LABEL**值，甚至连文件系统类型也会显示。当然这个命令后面也可以指定哪个分区：

```
[root@localhost ~]# blkid /dev/sdb5
/dev/sdb5: LABEL="TEST" UUID="c61117ca-9176-4d0b-be4d-1b0f434359a7" TYPE="ext4"
```

获得**UUID**后，如何使用它呢？

```
[root@localhost ~]# umount /newdir
[root@localhost ~]# mount UUID="c61117ca-9176-4d0b-be4d-1b0f434359a7" /newdir
[root@localhost ~]# df -h
文件系统          容量  已用  可用  已用%% 挂载点
/dev/sda3          14G   1.5G   12G   11%   /
tmpfs              160M    0   160M    0%   /dev/shm
/dev/sda1          97M   27M   66M   29%   /boot
/dev/sdb5          989M   18M  921M    2%   /newdir
```

也可以把下面这行写到 **/etc/fstab** 中

| | | | |
|---|---------|------|----------|
| UUID=c61117ca-9176-4d0b-be4d-1b0f434359a7 | /newdir | ext4 | defaults |
|---|---------|------|----------|

0 0

如果想让某个分区开机后就自动挂载，有两个办法可以实现：

1. 在 **/etc/fstab** 中添加一行，如上例中那行；
2. 把挂载命令写到 **/etc/rc.d/rc.local** 文件中，阿铭会经常把想要开机启动的命令加到这个文件中。系统启动完后会执行这个文件中的命令，所以只要你想开机后运行什么命令统统写入到这个文件下面吧，直接放到最后面即可，阿铭把挂载的命令放到该文件的最后一行了：

```
[root@localhost ~]# cat /etc/rc.d/rc.local
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
mount UUID="c61117ca-9176-4d0b-be4d-1b0f434359a7" /newdir
```

以上两种方法，任选其一，阿铭介绍第二种方法其实也是教给你一个小知识，如何让一些操作行为随系统启动而自动执行。另外，阿铭需要给你一个小建议，那就是挂载磁盘分区的时候，尽量使用**UUID**或者**LABEL**这两种方法。

命令：**umount**

在上面的小实验中，阿铭多次用到这个命令，这个命令也简单的很，后边可以跟挂载点，也可以跟分区名

(/dev/hdb1), 但是不可以跟LABEL和UUID.

```
[root@localhost ~]# umount /dev/sdb5
[root@localhost ~]# mount UUID="c61117ca-9176-4d0b-be4d-1b0f434359a7" /newdir
[root@localhost ~]# umount /newdir
[root@localhost ~]# mount UUID="c61117ca-9176-4d0b-be4d-1b0f434359a7" /newdir
```

umount 命令有一个非常有用的选项那就是 '-l', 有时候你会遇到不能卸载的情况:

```
[root@localhost newdir]# umount /newdir
umount: /newdir: device is busy.
(In some cases useful info about processes that use
the device is found by lsof(8) or fuser(1))
```

这是因为当前目录为要卸载的分区上, 解决办法有两种, 一是到其他目录, 二是使用 '-l' 选项:

```
[root@localhost newdir]# umount -l /newdir
[root@localhost newdir]# df -h
文件系统          容量  已用  可用  已用%% 挂载点
/dev/sda3          14G   1.5G   12G   11% /
tmpfs              160M    0   160M    0% /dev/shm
/dev/sda1          97M   27M   66M   29% /boot
```

建立一个swap文件增加虚拟内存

从装系统时就接触过这个**swap**了, 它类似与**windows**的虚拟内存, 分区的时候一般大小为内存的**2**倍, 如果你的内存超过**8G**, 那么你分**16G**似乎是没有必要了。分**16G**足够日常交换了。然而, 还会有虚拟内存不够用的情况发生。如果真遇到了, 莫非还要重新给磁盘分区? 当然不能, 那我们就增加一个虚拟的磁盘出来。基本的思路就是: 建立**swapfile** -> 格式化为**swap**格式 -> 启用该虚拟磁盘。

```
[root@localhost ~]# dd if=/dev/zero of=/tmp/newdisk bs=4k count=102400
记录了102400+0 的读入
记录了102400+0 的写出
419430400字节 (419 MB) 已复制, 2.59193 秒, 162 MB/秒
```

“**dd**” 这个命令阿铭经常用到, 所以请你也要掌握它的使用方法, 其实也不难, 用 “**if**” 指定源, 基本上除了 “**/dev/zero**” 外基本上不会写别的, 而**/dev/zero** 是**UNIX**系统特有的一个文件, 它可以提供源源不断的 “**0**”, 关于它的其他信息请你在网上查一下资料。 “**of**” 指定目标文件, “**bs**” 定义块的大小, “**count**” 定义块的数量, 这两个参数的多少决定了目标文件的大小, 目标文件大小 = **bs x count**. 阿铭用**dd**建了一个大小为**400M**的文件, 然后格式化**成swap**格式:

```
[root@localhost ~]# mkswap -f /tmp/newdisk
Setting up swapspace version 1, size = 409596 KiB
no label, UUID=29832cab-04b9-4083-a667-9a5795a5d490
```

格式化完后, 就可以挂载上使用了:

```
[root@localhost ~]# free -m
              total        used         free       shared    buffers     cached
Mem:           318         314            4           0           5         278
-/+ buffers/cache:          30          288
Swap:          2047            0          2047
[root@localhost ~]# swapon /tmp/newdisk
[root@localhost ~]# free -m
              total        used         free       shared    buffers     cached
Mem:           318         314            4           0           5         278
-/+ buffers/cache:          31          287
Swap:          2447            0          2447
```

前后对比**swap**分区多了**400M**空间。其中 “**free**” 这个命令用来查看内存使用情况, “**-m**” 表示以**M**为单位显示, 阿铭会在后面介绍该命令。

磁盘配额

磁盘配合其实就是给每个用户分配一定的磁盘额度, 只允许他使用这个额度范围内的磁盘空间。在**linux**系统中, 是多用户多任务的环境, 所以会有很多人共用一个磁盘的情况。针对每个用户去限定一定量的磁盘空间是有必要的, 这样才显得公平。随着硬件成本的降低, 服务器上的磁盘资源似乎不再刻意的去限制了, 所以磁盘配额也就可有可无了, 但是你也需要了解一下这部分内容, 用到时必须会操作。

在**linux**中, 用来管理磁盘配额的东西就是**quota**了。如果你的**linux**上没有**quota**, 则需要你安装这个软件包 **quota-3.13-5.el5.RPM** (其实版本是多少无所谓了, 关键是这个软件包)。 **quota**在实际应用中是针对整个分区进行限制的。比如, 如果我们限制了**/dev/sdb1**这个分区, 而**/dev/sdb1** 是挂载在**/home** 目录下的, 那么**/home** 所有目录都会受到限制。

quota 这个模块主要分为**quota** **quotacheck** **quotaoff** **quotaon** **quotastats** **edquota** **setquota** **warnquota** **repquota**这几个命令, 下面就分别介绍这些命令。

命令：**quota**

“quota” 用来显示某个组或者某个使用者的限额。

语法：`quota [-guvs] [user,group]`

“-g” 显示某个组的限额

“-u” 显示某个用户的限额

“-v” 显示的意思

“-s” 选择inode或硬盘空间来显示

命令：**quotacheck**

“quotacheck” 用来扫描某一个磁盘的quota空间。

语法：`quotacheck [-auvg] /path`

“-a” 扫描所有已经mount的具有quota支持的磁盘

“-u” 扫描某个使用者的文件以及目录

“-g” 扫描某个组的文件以及目录

“-v” 显示扫描过程

“-m” 强制进行扫描

命令：**edquota**

“edquota” 用来编辑某个用户或者组的quota值。

语法：`edquota [-u user] [-g group] [-t]`

“-u” 编辑某个用户的quota

“-g” 编辑某个组的quota

“-t” 编辑宽限时间

“-p” 拷贝某个用户或组的quota到另一个用户或组

当运行 `edquota -u user` 时，系统会打开一个文件，你会看到这个文件中有7列，它们分别代表的含义是：

“Filesystem” 磁盘分区，如/dev/sdb5

“blocks” 当前用户在当前的Filesystem中所占用的磁盘容量，单位是Kb。该值请不要修改。

“soft/hard” 当前用户在该Filesystem内的quota值，soft指的是最低限额，可以超过这个值，但必须要在宽限时间内将磁盘容量降低到这个值以下。hard指的是最高限额，即不能超过这个值。当用户的磁盘使用量高于soft值时，系统会警告用户，提示其要在宽限时间内把使用空间降低到soft值之下。

“inodes” 目前使用掉的inode的状态，不用修改。

命令：**quotaon**

“quotaon” 用来启动quota，在编辑好quota后，需要启动才能是quota生效

语法：`quotaon [-a] [-uvg directory]`

“-a” 全部设定的quota启动

“-u” 启动某个用户的quota

“-g” 启动某个组的quota

“-s” 显示相关信息

命令：**quotaoff**

“quotaoff” 用来关闭quota，该命令常用只有一种情况 `quotaoff -a` 关闭全部的quota。

以上讲了很多quota的相关命令，那么接下来阿铭教你如何在实践应用中去这个磁盘配额。整个执行过程如下：

首先先确认一下，你的/home目录是不是单独的挂载在一个分区下，用df 查看即可。如果不是则需要你跟我一起做。否则这一步即可省略。

| 文件系统 | 1K-块 | 已用 | 可用 | 已用% | 挂载点 |
|-----------|----------|---------|----------|-----|----------|
| /dev/sda3 | 14347632 | 1899376 | 11719424 | 14% | / |
| tmpfs | 163308 | 0 | 163308 | 0% | /dev/shm |
| /dev/sda1 | 99150 | 26808 | 67222 | 29% | /boot |

阿铭的linux系统中，/home并没有单独占用一个分区。所以需要把/home目录挂载在一个单独的分区下，因为quota是针对分区来限额的。下面阿铭把 /dev/sdb5 挂载到/home 目录下， 编辑 /etc/fstab 把刚才添加的那行修改为：

| | | | |
|---|-------|------|----------|
| UUID=c61117ca-9176-4d0b-be4d-1b0f434359a7 | /home | ext4 | defaults |
|---|-------|------|----------|

保存 /etc/fstab 后，运行 `mount -a` 命令挂载全部的分区。

```
[root@localhost ~]# mount -a
[root@localhost ~]# df -h
```

| 文件系统 | 容量 | 已用 | 可用 | 已用%% | 挂载点 |
|-----------|------|------|------|------|----------|
| /dev/sda3 | 14G | 1.9G | 12G | 14% | / |
| tmpfs | 160M | 0 | 160M | 0% | /dev/shm |

```
/dev/sda1          97M   27M   66M   29% /boot
/dev/sdb5          989M   18M  921M    2% /home
```

此时的 `/home` 为一个单独分区了。

1. 建立测试账户

首先建立一个**test**用户，则同时建立了一个**test**组。其中**uid**和**gid**都为**511**，然后又建立一个**test1**账号，使其加入**test**组，查看**/etc/passwd**文件发现**test**和**test1**用户的**gid**都为**511**。

```
[root@localhost ~]# useradd test
[root@localhost ~]# grep test /etc/passwd
test:x:511:511::/home/test:/bin/bash
[root@localhost ~]# useradd -g 511 test1
[root@localhost ~]# grep test1 /etc/passwd
test1:x:512:511::/home/test1:/bin/bash
```

2. 打开磁盘的quota功能

默认linux并没有对任何分区做**quota**的支持，所以需要我们手动打开磁盘的**quota**功能，你是否记得，在前面内容中分析**/etc/fstab**文件的第四列时讲过这个**quota**选项（**usrquota**, **grpquota**），没错，要想打开这个磁盘的**quota**支持就是需要修改这个第四列的。用**vi**编辑**/etc/fstab** 编辑刚才加的那一行，如下：

```
UUID=c61117ca-9176-4d0b-be4d-1b0f434359a7    /home    ext4    defaults,usrquota,grpquota    0 0
```

保存 **/etc/fstab** 后，重新挂载**/home**分区。

```
[root@localhost ~]# umount /home/
[root@localhost ~]# mount -a
[root@localhost ~]# mount
/dev/sda3 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)
/dev/sda1 on /boot type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
/dev/sdb5 on /home type ext4 (rw,usrquota,grpquota)
```

使用 **mount** 命令可以查看到 **/home** 分区已经加上了“**usrquota,grpquota**”两个配额相关的参数。

3. 扫描磁盘的使用者使用状况，并产生重要的aquota.group与aquota.user

这一步就需要用到**quotacheck**了，**aquota.group**与**aquota.user**分别是组以及用户磁盘配额需要的配置文件。如果没有这两个文件，则磁盘配额是不会生效的。

```
[root@localhost ~]# quotacheck -augv
```

可能会有些错误信息，不要管它。看一看你的**/home**分区下是否多了两个文件(**aquota.group**, **aquota.user**)

```
[root@localhost ~]# ll /home/
总用量 44
-rw----- 1 root  root  7168 5月  12 02:07 aquota.group
-rw----- 1 root  root  8192 5月  12 02:07 aquota.user
drwxr-xr-x 2 root  root  4096 5月  12 00:11 dir1
drwx----- 2 root  root 16384 5月  11 23:18 lost+found
drwx----- 3 test  test  4096 5月  12 01:59 test
drwx----- 3 test1 test  4096 5月  12 02:00 test1
```

如果有了，则可以进入下一步了。

4. 启动quota配额

```
[root@localhost ~]# quotaon -av
/dev/sdb5 [/home]: group quotas turned on
/dev/sdb5 [/home]: user quotas turned on
```

5. 编辑用户磁盘配额

先来设定**test**账户的配额，然后直接把**test**的配额拷贝给**test1**即可。这里就需要用到**edquota**了。

```
[root@localhost ~]# edquota -u test
```

将下面内容

| | | | | | | |
|-----------|----|---|---|---|---|---|
| /dev/sdb5 | 20 | 0 | 0 | 5 | 0 | 0 |
|-----------|----|---|---|---|---|---|

修改为:

| | | | | | | |
|-----------|----|-------|-------|---|---|---|
| /dev/sdb5 | 20 | 20000 | 30000 | 5 | 0 | 0 |
|-----------|----|-------|-------|---|---|---|

其中单位是Kb, 所以**soft** 值大约为20Mb, **hard**值为30Mb, 保存这个文件, 保存的方式跟vi一个文件的方式一样的。下面将**test**的配额复制给**test1**。

```
[root@localhost ~]# edquota -p test test1
```

下面继续设定宽限时间:

```
[root@localhost ~]# edquota -t
```

将**7days** 改为 **1days**

| | | |
|-----------|-------|-------|
| /dev/sdb5 | 1days | 1days |
|-----------|-------|-------|

下面查看一下**test**以及**test1**用户的配额吧。

```
[root@localhost ~]# quota -uv test test1
Disk quotas for user test (uid 511):
    Filesystem blocks quota limit grace files quota limit grace
    /dev/sdb5   20   20000 30000          5      0      0
Disk quotas for user test1 (uid 512):
    Filesystem blocks quota limit grace files quota limit grace
    /dev/sdb5   20   20000 30000          5      0      0
```

5. 编辑组磁盘配额

```
[root@localhost ~]# edquota -g test
```

修改为:

| | | | | | | |
|-----------|----|-------|-------|----|---|---|
| /dev/sdb5 | 40 | 40000 | 50000 | 10 | 0 | 0 |
|-----------|----|-------|-------|----|---|---|

设定组**test**的**soft**配额值为40M, **hard**值为50M。下面查看组**test**的配额。

```
[root@localhost ~]# quota -gv test
Disk quotas for group test (gid 511):
    Filesystem blocks quota limit grace files quota limit grace
    /dev/sdb5   40   40000 50000         10      0      0
```

7. 设定开机启动

前面已经讲到启动磁盘配额的命令是 `quotaon -aug` 所以要想开机启动, 只需将这条命令加入到 `/etc/rc.d/rc.local`文件即可。

```
[root@localhost ~]# echo "quotaon -aug" >> /etc/rc.d/rc.local
```

阿铭建议你最好再扩展学习一下: <http://www.aminglinux.com/bbs/thread-5424-1-1.html>

教程答疑: [请移步这里](#).

欢迎你加入 [阿铭学院](#) 和阿铭一起学习Linux, 让阿铭成为你Linux生涯中永远的朋友吧!

[1](1, 2) 磁盘分区有三种形式: 主分区、扩展分区和逻辑分区。主分区最多可以有四个, 如果想再多分分区, 则需要先分三个主分区, 然后第四个分为扩展分区, 然后再把扩展分区分成若干个逻辑分区, 逻辑分区最多可以分多少个? 之前阿铭使用ide接口的磁盘尝试过(hda, hdb这样的磁盘), 最多可以分10个, 至于scsi接口的磁盘(sda, sdb)最多可以分多少个, 阿铭没有做实验, 这留给你来做吧。

[PREVIOUS](#) | [NEXT](#) | [INDEX](#)

© Copyright 2013, lishiming.net. Created using [Sphinx](#) 1.3b1[网站统计](#) .