**[parted的使用](http://blog.csdn.net/michaelwubo/article/details/47802877)**

2015-08-20 09:11 2672人阅读 [评论](http://blog.csdn.net/michaelwubo/article/details/47802877#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/michaelwubo/article/details/47802877#report)

本文章已收录于：

http://static.blog.csdn.net/images/category_icon.jpg分类：

Linux文件系统（12） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpghttp://static.blog.csdn.net/images/arrow_triangle_up.jpg

http://blog.csdn.net/michaelwubo/article/details/47802877

[作者同类文章](http://blog.csdn.net/Michaelwubo/article/category/2538365" \t "_blank)X

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录[(?)](http://blog.csdn.net/michaelwubo/article/details/47802877" \o "系统根据文章中H1到H6标签自动生成文章目录)

GNU Parted是一个用于对分区及其文件系统进行建立、修改、调整、检查、复制等操作的一个工具。对于安装新的系统来说，使用GNU Parted来为它进行分区是十分方便的。此外，你还可以用它来检查磁盘的使用状况，在不同的磁盘之间复制数据，甚至是“映象”磁盘——将一个磁盘的安装 完好地复制到另一个磁盘中。

我们假设读者您已经对文件系统和分区的概念有所认识。如果您想深入了解这方面的知识，那么我们推荐你读一下GNU Storage Guide[1](file:///\\\\data\\mynotes\\cmd_study\\parted_guide\\parted_guide.html" \l "foot25" \t "_blank)。

GNU Parted的设计宗旨之一就是尽量减少用户数据的损失的可能。例如，若对分区进行操作期间发生意外（如突如其来的断电），GNU Parted能确保你数据的损失的可能性减到最少。此外，GNU Parted还会进行许多安全检查。尽管如此，软件的bugs总会带来一些不稳定的因素。因此，我们建议您先将您的数据备份后再使用GNU Parted进行磁盘操作。另外请注意，libreiserfs提供了对reiserfs分区格式的支持，它可能没有达到上面所说的那样“将数据损失降到 最低”的设计宗旨。同样，ntfsresize等等第三方工具也不一定绝对可靠。因而，进行磁盘操作时请您慎重。

GNU Parted的主页是：[http://www.gnu.org/software/parted](http://www.gnu.org/software/parted" \t "_blank)。一些库文件和前端可以从这里下载：[ftp://ftp.gnu.org/gnu/parted](ftp://ftp.gnu.org/gnu/parted" \t "_blank)。

**使用 GNU Parted的 Static binaries**

**介绍**

如果你需要在一台没有安装GNU/[**Linux**](http://lib.csdn.net/base/linux)使用GNU Parted，或者你需要调整**[linux](http://lib.csdn.net/base/linux" \o "Linux知识库" \t "_blank)**系统的根分区或引导分区，那么你就需要使用一张引导磁盘。以前我们有专门的引导磁盘映象。但是现在已经没有什么必 要用到它了，因为大量包含GNU Parted 的live CD 和急救磁盘已经出现。然而，你要留意一下，可能上面的GNU Parted 的版本是过时的。为了弥补其不足，Static binariy格式的GNU Parted出现了。你可以轻易地得到最新版本的Static binariy GNU Parted，并按照一下方法使用之。

**建立 Parted 磁盘**

1. 引导你的系统
2. 从 <ftp://ftp.gnu.org/parted/static>中下载‘parted-static-VERSION.tgz’
3. 将压缩包解压，你会得到一个名叫parted.static的文件Unpack the tarball, resulting in a file called “parted.static".
4. 插入一张软盘 Insert a floppy.
5. 对插入的软盘进行低级格式化。（在GNU/Linux下可以使用util-linux包里面的fdformat工具）这样做是因为软盘中有可能有一些坏区，我们要将它检查出来。
6. 建立文件系统。例如： $ parted /dev/fd0 mklabel loop mkpartfs primary ext2 0 1.4
7. 挂载软盘： $ mount -t ext2 /dev/fd0 /mnt/floppy
8. 将 parted.static 复制到软盘上 $ cp parted.static /mnt/floppy
9. 卸载软盘 $ umount /mnt/floppy

**使用Parted**

**关于分区**

很不幸，对磁盘分区是十分复杂的。因为各个**[操作系统](http://lib.csdn.net/base/operatingsystem" \o "操作系统知识库" \t "_blank)**对磁盘操作的差异，所以你需要考虑许多因素。因此你最好对这方面的知识有一个充分的了解。

**使用 GNU Parted**

GNU Parted有两种运行模式：命令行模式和交互模式。你必须用一下的这种方式来运行 GNU Parted

# parted device

device就是你想要编辑的硬盘设备名称。如果你是一个比较懒的人，省略了这个参数，不要紧， GNU Parted会尝试猜测一下你想要操作的设备。在命令行模式下，你需要将parted的命令一个接一个地输入到一行中。

而在交互模式下，每次只能输入一条parted命令。一旦你按回车键确认，命令就马上执行，对磁盘的更改就立刻生效。例如：

(parted) resize 1 52.0005Mb 104.5Mb

(parted) mkfs 2 fat16

在不被混淆的前提下，你可以对命令进行缩写。例如，你可以用p代替print。若软件被翻译成了你的本国语言，那么你甚至可以用本国语言来输入命令，而不 单是英语。不过，这也许会造成混乱，所以还是建议使用英语输入命令。注意， GNU Parted对大小写不敏感。

对于表示分区位置的数字可以使用整数和小数。而数字后面的字母就表示了你所使用的单位。关于 GNU Parted的单位的介绍请参考第节 [Unit]。 如果没有指明数字的单位，那么就会采用默认值。若数字为负数，那么就表示从磁盘的末尾开始计算。如“-1s”就表示硬盘最后一个扇区，就是硬盘的最末尾 处。对于你所给出的位置，parted会稍作计算使之合理。例如，当你使用GB作为单位时，分区位置的误差不会超过+-500M。当然，你可以使用代表扇 区的单位s来指定精确的位置。

对于一条命令，如果你没有给出一个参数，那么 GNU Parted会再次向你提问，例如：

(parted) resize 1

Start? 0Gb

End? 40Gb

当你的某些操作具有一定的危险性时，Parted 会警告你，除非是那些本来就是很危险的操作（ 像rm,mklabel和mkfs）。例如当你想将一个分区过分地缩小，缩小后的空间不足以存放该分区现有的数据，那么Parted 就会自动给出一个安全的分区缩小的方案，以免丢失数据。如果这个Parted 给出的缩小结果和你给出的相差太大，那么Parted 会询问你是否愿意继续。因为不同的分区系统都有复杂的限制，因此Parted 所实际的更改会和你给出的数字有一些微小的出入（例如，你要Parted 为你建立一个10.4MB的分区，但你得到的分区最终只有10.325MB）。如果误差太大，Parted 同样会让你确认一下是否继续。

**命令行模式的选项**

当你调用命令行模式时，Parted的语法是这样的：

# parted [option] device [command [argument]]

有几个选项可以供你使用。对于其他涉及分区细节的Parted命令，请参见下一节。选项以一个横杠开头，而Parted命令前则没有这个横杠。

选项:

‘-h’

‘-help’ 显示帮助信息

‘-i’

‘-interactive’

进入交互模式

‘-v’

‘-version’

显示软件的版本号

**Parted 交互模式下的命令**

GNU Parted 提供以下命令:

**check（检查）**

格式：check number

检查编号中所指定的文件系统是否有什么错误.。

例：

(parted) check 1

信息: ext2 文件系统通过基本检查。至于更广泛的检查，请使用 e2fsck 程序

|  |
| --- |
|  |

**cp（复制）**

格式：cp [from-device] from-number to-number

将分区 from-number上的文件系统完整地复制到分区to-number 中。分区to-number 上的数据会被全部删除。你可以再指定一个来源硬盘的设备名称 from-device来指定来源分区所在的硬盘。

该命令支持以下文件系统：

* ext2, ext3 （所提供的目标分区必须比源分区要大）
* fat16, fat32
* linux-swap (相当于再目标分区执行mkswap 命令)
* reiserfs (如果安装了libreiserfs库)

例：

/\*最开始第一个分区拷贝了一个归档文件，而第二个分区什么都没有\*/

(parted) print

磁盘 /dev/sdb: 3146MB

数字 开始： End Size 类型 文件系统 标志

1 31.7kB 302MB 302MB 主分区 ext2

2 302MB 705MB 403MB 主分区 ext2

(parted) cp /dev/sdb 1 2

cp /dev/sdb 1 2

原始块复制... 94%

(parted)

/\*查看第二个分区，第一个分区的归档文件已经从拷贝过来，同时验证文件没有问题\*/

如果是ext3文件系统，则会给出下面的错误：

(parted) cp /dev/sdb 1 2

错误: File system has an incompatible feature enabled.

注：我在**[测试](http://lib.csdn.net/base/softwaretest" \o "软件测试知识库" \t "_blank)**的时候，发现了这样的一个现象：首先parted目前对CP的功能支持EXT2但不支持EXT3,其他文件系统，我没有测试。如果你用 mkfs.ext2命令来对分区做文件系统，做CP的时候，也会给出``File system has an incompatible feature enabled. ``的提示，你需要使用parted自己的mkfs子命创建文件系统才能执行CP功能，我不太清楚parted自身的mkfs和系统自带的 mkfs.ext2在做文件系统的时候有什么差别，毕竟ext2的定义是一样的呀，那么在组织文件结构的时候应该是一致的，这个需要有时间在去看看了。

所以，为了以后能够用到这个功能，分区的时候的可以使用parted，也可以使用fdisk，但是做文件系统的时候，只能使用parted自己的mkfs。

不过，现在，还有谁使用ext2吗？当然我们变态的使用这个功能，需要cp的时候，先降级为ext2，完成后，再升级为ext3。

**help（帮助）**

格式：help [command]

输出一个命令的帮助信息

例：

(parted) help resize

resize NUMBER START END resize partition NUMBER and its file system

NUMBER 是 Linux 使用的分区编号。在 msdos 磁盘标签中，主分区编号为 1 - 4，逻辑分区从 5 开始。

START and END are disk locations, such as 4GB or 10%. Negative values count from the end of the disk. For example,

-1s specifies exactly the last sector.

分区必须具有以下的其中一个文件系统类型： ext3, ext2, fat32, fat16, hfsx, hfs+, hfs, linux-swap

**mklabel（建立磁盘分区表类型）**

格式：mklabel label-type

建 立一个label-type类型的磁盘分区表类型。新的分区表类型不会有任何分区操作。这个命令通常不会破坏你的数据，但是它有可能让你 的数据变得不可读（你可以尝试将你的磁盘分区类型-默认是msdos-改成gpt试试）。这是你就需要用到rescue命令来恢复分区。关于rescue 命令的用法请参见第。Parted 能在各种分区表上工作。

label-type 必须是一下这些类型：

* bsd
* loop (raw disk access)
* gpt
* mac
* msdos
* pc98
* sun

例：

(parted) mklabel msdos

建立一个MS-DOS格式的分区表。这对于PC而言是一个非常常见的操作

**mkfs（建立文件系统）**

格式：mkfs number fs-type

在指定的分区上建立指定的文件系统。这个命令会破坏该分区上的所有数据。

该命令支持以下文件系统：

* ext2
* fat16, fat32
* linux-swap
* reiserfs (如果安装了libreiserfs库)

例：

(parted) mkfs 2 fat32

在第二个分区上建立FAT32格式的文件系统

**mkpart（建立新分区）**

格式：mkpart part-type [fs-type] start end

建立一个新的分区。若要建立非扩展分区，就最好用fs-type来指定文件系统。start和end是新分区开始和结束的具体位置。part-type是 以下类型之一: primary（主分区）, extended（扩展分区）, logical（逻辑分区）. 扩展分区和逻辑分区只对msdos 和dvh分区表有效。

fs-type必须是以下文件系统：

* ext2
* fat16, fat32
* hfs, hfs+, hfsx
* linux-swap
* NTFS
* reiserfs
* ufs

例：

(parted) mkpart logical 0.0 692.1

建立一个包含ext2文件系统的逻辑分区，其位置是从磁盘的开始位置（0.0）到692.1M之间

**mkpartfs（建立分区及其文件系统）**

格式：mkpartfs part-type fs-type start end

建立一个新的分区，并在上面建立文件系统。与mkpart相似，分区的位置处于begin和end之间，默认的单位是MB.。请勿使用这个命令来恢复一个被删除了的分区（应使用mkpart）。part-type是以下类型之一:

primary（主分区）, extended（扩展分区）, logical（逻辑分区）. 扩展分区和逻辑分区只对msdos和dvh分区表类型有效。 fs-type必须是以下文件系统：

* ext2 (不支持ext3)
* fat16, fat32
* linux-swap
* reiserfs (如果安装了libreiserfs库)

例：

(parted) mkpartfs logical ext2 440 670

建立一个包含ex2格式文件系统的逻辑分区。其开始位置是440 MB，结束位置是670MB

**move（移动）**

格式：move number start end

该命令能将选定的分区在硬盘中移动。start可以指定一个新的分区的起始位置。不过，你不能将一个分区移到另一个分区上。也就是说，你只能在空闲的地方中移动分区。如果你想改变分区的大小，那么你所需要的命令是resize,见第节。另外，虽然分区被移动了，但它的分区编号是不会改变的。

该命令支持以下文件系统：

* ext2, ext3 （所提供的目标分区必须比源分区要大）
* fat16, fat32
* linux-swap
* reiserfs (如果安装了libreiserfs库)

例：

(parted)p

........

5 705MB 1207MB 502MB 逻辑分区 ext2

........

(parted) move

分区编号？ 5

起始点？ 1210

结束点？ [1712MB]?

原始块复制... 13% (剩余时间 00:54)

(parted)p

.......

5 1210MB 1712MB 502MB 逻辑分区 ext2

........

注：主分区和逻辑分区之间不能互相移动

**name （命名）**

格式：name number name

为编号所指定的分区命名（只支持GPT, Mac, MIPS 和PC98 格式的分区）。名字可以用引号括起来。

例：

(parted) name 2 ’Secret Documents’

将第2个分区命名为 ’Secret Documents’。

**print （输出信息）**

格式：print [number]

输出parted正在操作的硬盘的分区表的信息，或者是某一个具体分区的信息。

例：

(parted) print

磁盘 /dev/sdb: 3146MB

Sector size (logical/physical): 512B/512B

分区表：msdos

数字 开始： End Size 类型 文件系统 标志

2 302MB 705MB 403MB 主分区 ext2

1 705MB 1008MB 303MB 主分区 ext2

3 2000MB 3146MB 1146MB 扩展分区 lba

5 2000MB 2100MB 100MB 逻辑分区 ext2

(parted) print 1

Minor：1

标志：

文件系统：ext2

大小: 303MB (9.62%)

最小尺寸: 19.5MB (0.62%)

最大尺寸: 3146MB (100%)

**quit （退出）**

格式：quit

退出Parted.

只有当Parted退出后，Linux内核才会知道磁盘的设置已经被改变了。但是实际上只要你一执行一条命令，更改马上就会写入磁盘。不过，磁盘缓存有可能延缓写入的操作。

**rescue （恢复）**

格式：rescue start end

如果你不小心用Parted的rm命令删除了一个分区，那么这个命令可以帮你恢复。你需要给出所误删的分区的大概的开始和结束的位置。Parted 就会在你给出的磁盘区域内去寻找，如果找到这个分区，那么Parted 就会询问你是否重新建立这个分区。

例：

/\*未删除时的分区列表\*/

(parted) p

磁盘 /dev/sdb: 3146MB

Sector size (logical/physical): 512B/512B

分区表：msdos

数字 开始： End Size 类型 文件系统 标志

2 302MB 705MB 403MB 主分区 ext2

1 705MB 1008MB 303MB 主分区 ext2

3 2000MB 3146MB 1146MB 扩展分区 lba

5 2000MB 2100MB 100MB 逻辑分区 ext2

(parted) rm 1 /\*删除第一个分区 \*/

/\*删除后的分区列表\*/

(parted) p

磁盘 /dev/sdb: 3146MB

Sector size (logical/physical): 512B/512B

分区表：msdos

数字 开始： End Size 类型 文件系统 标志

2 302MB 705MB 403MB 主分区 ext2

3 2000MB 3146MB 1146MB 扩展分区 lba

5 2000MB 2100MB 100MB 逻辑分区 ext2

/\*开始找回\*/

(parted) rescue 700 1100

(parted) p

磁盘 /dev/sdb: 3146MB

Sector size (logical/physical): 512B/512B

分区表：msdos

数字 开始： End Size 类型 文件系统 标志

2 302MB 705MB 403MB 主分区 ext2

1 705MB 1008MB 303MB 主分区 ext2

3 2000MB 3146MB 1146MB 扩展分区 lba

5 2000MB 2100MB 100MB 逻辑分区 ext2

/\*由此可见分区已经找到了\*/

**resize （调整大小）**

格式：resize number start end

将编号所指定的分区调整大小。分区的开始位置和结束位置由start和end决定。resize不会改变分区的编号。注意，如果更改扩展分区的大小，那么你必须保证新的分区大小能够容纳里面的逻辑分区。另外，在调整大小前你无须对分区进行碎片整理。

该命令支持以下文件系统：

* ext2, ext3 - 限制：新的分区开始位置必须和原来的开始位置相同。也就是说，你只能向后扩展分区的大小。
* fat16, fat32
* hfs, hfs+, hfsx - 限制：新的分区开始位置必须和原来的开始位置相同。而且新的end必须小于原来的end。也就是说，你只能够缩小分区。
* linux-swap
* reiserfs (如果安装了libreiserfs库)

例：

(parted) p

数字 开始： End Size 类型 文件系统 标志

1 705MB 1008MB 303MB 主分区 ext2

(parted) resize 1 705 1200

(parted) p

数字 开始： End Size 类型 文件系统 标志

1 705MB 1201MB 496MB 主分区 ext2

**rm （删除分区）**

格式：rm number

将编号所指定的分区删除。如果你误删了某个分区，那么你还可以用mkpart来挽救它（而不是用mkpartds）。注意，如果你使用的是MSDOS格式 的分区表，那么假如你删除了一个逻辑分区，那么该分区后面的逻辑分区的编号都会相应地向前改变。例如，你删除了编号为6的逻辑分区，那么后面的7、8号逻 辑分区的编号将会改变，分别变成6、7号逻辑分区。若在Linux下。这意味着你可能要改变/etc/fstab文件。

例：

(parted) rm 3

将编号为3的分区删除。

**select （选择设备）**

格式：select device

选择Parted将要编辑的磁盘设备。这个磁盘设备可以是一个Linux硬盘设备，一个软盘、一个软RAID磁盘阵或者是LVM逻辑卷。

例：

(parted) select /dev/sdb

使用 /dev/sdb

**set （设置标记）**

格式：set number flag state

更改编号所指定的分区的标记。哪些标记可用是由你的分区表决定的。标记有两种状态，要么是on（生效），要么是off（失效）。

* ‘boot’(用 于Mac,MS-DOS,PC98分区表) - 如果你想从某个分区引导，那么你应该将这个分区的boot标记设置为on。对于MS－DOS分区表，一个硬盘上只能有一个分区是可引导的。如果你将 LILO装到某个分区上，那么这个分区一定是可引导的（也就是说你应该将安装了LILO的分区的boot标记设为on）。对于PC98分区表，所有 ext3分区都必须是设置为可引导的。（Parted会对此强行操作）
* ‘lba’ (用于MS-DOS分区表)-这个标记可以告诉MS DOS、MS Windows 9x和 MS Windows ME等系统使用LBA（Logic Block Addressing）模式。
* ‘root’ (用于Mac分区表) - 如果某分区是Linux系统的根分区，那么就要对该分区启用这个标记。
* ‘swap’ (用于Mac分区表) - 如果某分区是Linux系统的交换分区，那么就要对该分区启用这个标记。
* ‘hidden’ (用于MS-DOS, PC98分区表) - 启用这个标记能使一个分区‘隐藏’起来。（仅对微软的操作系统有效）
* ‘raid’ （用于MS-DOS分区表）-启用这个标记能够告诉Linux系统这是一个软RAID磁盘阵。
* ‘LVM’（用于MS-DOS分区表）-启用这个标记能够告诉Linux系统这是一个LVM逻辑。

使用print命令能够将所有能用的标记显示出来。

例：

(parted) set 1 boot on

将编号为1的分区的boot标记设定为on（生效）

**unit （设置单位）**

格式：unit unit

为Parted选择一个表示磁盘大小的单位。之后，凡是关于磁盘的位置（如分区的开始位置等等）的数字都是使用这个单位，你就可以不用给出单位后 缀。（如：如果你选择了MB作为单位，那么512就代表512MB，你就不用额外打多个MB之类的东西）你可以使用下列单位：

* ‘s’ ：扇区 （通常一个扇区的大小是512个字节）
* ‘B’ byte
* ‘kB’ kilobyte (1000 bytes)
* ‘MB’ megabyte (1000000 bytes)
* ‘GB’ gigabyte (1000000000 bytes)
* ‘TB’ terabyte (1000000000000 bytes)
* ‘%’ 表示占整个磁盘设备的百分之多少（显而易见，这个数字必须在0和100之间）
* ‘cyl’ 柱面（这个具体的位置和BIOS的CHS地址有关）
* ‘chs’ 柱面（cylinders）,磁头（heads）,扇区（sectors）的地址（这个具体的位置和BIOS的CHS地址有关）
* ‘compact’ 这是一个比较奇怪的单位。输入时是使用MB作为单位，输出时的单位有可能不同，但是它确保输出时的单位能使用户一眼就看得出来 磁盘的大小的，而不用心里默默地进行复杂的换算。（例如。它会使用1GB而不是使用1024MB，使用2MB而不是2048KB）

一般来说，Parted只会使用默认的单位进行输出，除非输入的时候你没有给出数字的后缀。你可以在输入的数字后面加上一个单位来代替默认的单位（单位和 数字之间是没有空格的）。不过若你使用CHS地址作为单位时，你就不需要给定后缀。使用chs地址的格式是：“柱面:磁头:扇区”。对你所给出的数字地 址，Parted可能进行一些调整。例如，当你使用GB作为单位时，分区位置可能会有+-500M的误差。不过Parted总会尽量地使误差尽量地小。当 然，你可以使用单位s（也就是扇区）来指定一个精确的位置。若无法做到精确，Parted会让你给出一个最接近的解决方案。另外，若你给出的数字为负数， 那么就表示从磁盘的末尾开始计算。如“-1s”就表示硬盘最后一个扇区，就是硬盘的最末尾处。

例：

(parted) unit compact

(parted) p

磁盘 /dev/sdb: 3146MB

Sector size (logical/physical): 512B/512B

分区表：msdos

数字 开始： End Size 类型 文件系统 标志

2 302MB 705MB 403MB 主分区 ext2

1 705MB 1201MB 496MB 主分区 ext2 启动

3 2000MB 3146MB 1146MB 扩展分区 lba

5 2000MB 2100MB 100MB 逻辑分区 ext2

(parted) unit chs

(parted) print

磁盘 /dev/sdb: 1021,59,47

Sector size (logical/physical): 512B/512B

BIOS 柱面、磁头、簇 的结构为：1021，97，62。每一个柱面是 3079kB。

分区表：msdos

数字 开始： End 类型 文件系统 标志

2 98,0,0 228,96,61 主分区 ext2

1 229,0,0 389,96,61 主分区 ext2 启动

3 649,51,2 1021,59,47 扩展分区 lba

5 649,51,3 682,0,14 逻辑分区 ext2

(parted) unit mb

(parted) print

磁盘 /dev/sdb: 3146MB

Sector size (logical/physical): 512B/512B

分区表：msdos

数字 开始： End Size 类型 文件系统 标志

2 302MB 705MB 403MB 主分区 ext2

1 705MB 1201MB 496MB 主分区 ext2 启动

3 2000MB 3146MB 1146MB 扩展分区 lba

5 2000MB 2100MB 100MB 逻辑分区 ext2

parted命令可以划分单个分区大于2T的GPT格式的分区，也可以划分普通的MBR分区，fdisk命令对于大于2T的分区无法划分，所以用fdisk无法看到parted划分的GPT格式的分区。

       Parted 命令分为两种模式：命令行模式和交互模式。

       1、命令行模式： parted [option] device [command] ,该模式可以直接在命令行下对磁盘进行分区操作，比较适合编程应用。

       2、交互模式：parted [option] device 类似于使用fdisk /dev/xxx

       MBR：MBR分区表(即主引导记录)大家都很熟悉。所支持的最大卷：2T，而且对分区有限制：最多4个主分区或3个主分区加一个扩展分区

       GPT： GPT（即GUID分区表）。是源自EFI标准的一种较新的磁盘分区表结构的标准，是未来磁盘分区的主要形式。与MBR分区方式相比，具有如下优点。突破MBR 4个主分区限制，每个磁盘最多支持128个分区。支持大于2T的分区，最大卷可达18EB。

       parted是一个可以分区并进行分区调整的工具，他可以创建，破坏，移动，复制，调整ext2 linux-swap fat fat32 reiserfs类型的分区，可以创建，调整，移动Macintosh的HFS分区，检测jfs，ntfs，ufs，xfs分区。

    使用方法：parted [options] [device [command [options...]...]]

    options

    -h  显示帮助信息

    -l  显示所有块设备上的分区

    device

    对哪个块设备进行操作，如果没有指定则使用第一个块设备

    command [options...]

    check partition

    对分区做一个简单的检测

    cp [source-device] source dest

    复制source-device设备上的source分区到当前设备的dest分区

    mklabel label-type

    创建新分区表类型，label-type可以是："bsd", "dvh", "gpt",  "loop","mac", "msdos", "pc98", or "sun" 一般的pc机都是msdos格式，如果分区大于2T则需要选用gpt格式的分区表。

    mkfs partition fs-type

    在partition分区上创建一个fs-type文件系统，fs-type可以是："fat16", "fat32", "ext2", "linux-swap","reiserfs" 注意不支持ext3格式的文件系统，只能先分区然后用专有命令进行格式化。

    mkpart part-type [fs-type] start end

    创建一个part-type类型的分区，part-type可以是："primary", "logical", or "extended" 如果指定fs-type则在创建分区的同时进行格式化。start和end指的是分区的起始位置，单位默认是M。

    eg：mkpart  primary  0  -1   0表示分区的开始  -1表示分区的结尾  意思是划分整个硬盘空间为主分区

    mkpartfs part-type fs-type start end

    创建一个fs-type类型的part-type分区，不推荐使用，最好是使用mkpart分区完成后使用mke2fs进行格式化。

    name partition name

    给分区设置一个名字，这种设置只能用在Mac, PC98, and GPT类型的分区表，设置时名字用引号括起来

    select device

    在机器上有多个硬盘时，选择操作那个硬盘

    resize partition start end

    调整分区大小

    rm partition

    删除一个分区

    rescue start end

    拯救一个位于stat和end之间的分区

    unit unit

    在前面分区时，默认分区时数值的单位是M，这个参数卡伊改变默认单位，"kB", "MB",  "GB",  "TB"

    move partition start end

    移动partition分区

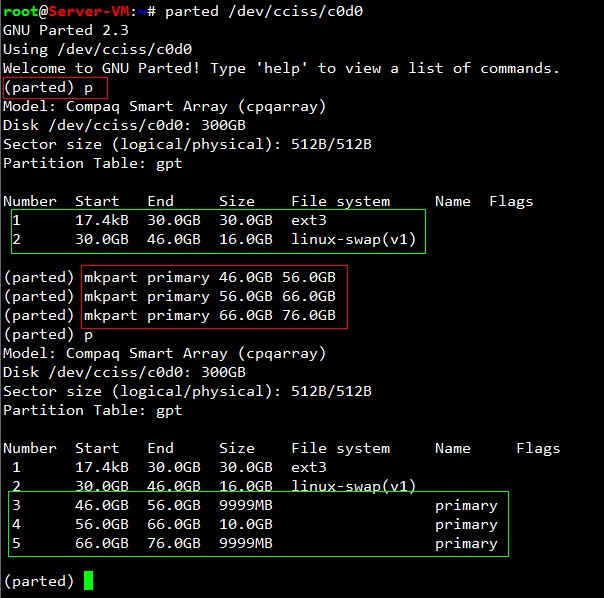
    print  显示分区表信息  quit 退出parted

    实战：

    1  初始信息

    注意绿框可以看到fdisk并不支持GPT格式的分区，然后输入命令p(print的简写)可以看到系统目前的所有分区

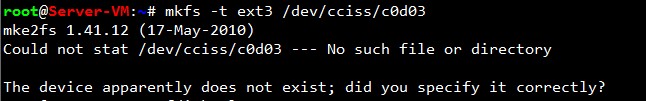
    2  连续划分三个大小为10G的主分区，不指定分区的文件系统类型，parted是划分后直接生效，没有fdisk中类似的w命令。

[](http://static.oschina.net/uploads/space/2012/0606/175716_4bH1_123777.jpg)

    可以看到新分区3 4 5 都是主分区 大小10G

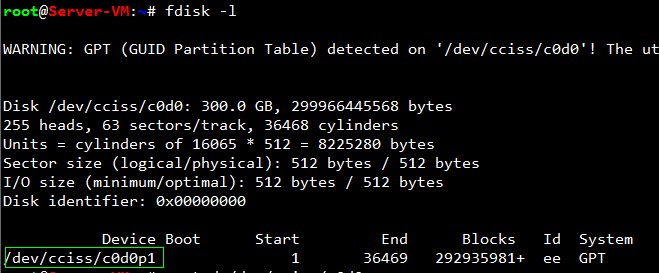
    3  也可以用非交互模式查看分区状况

    4  准备格式化新分区为ext3类型的文件系统

[](http://static.oschina.net/uploads/space/2012/0606/180011_rt9r_123777.jpg)

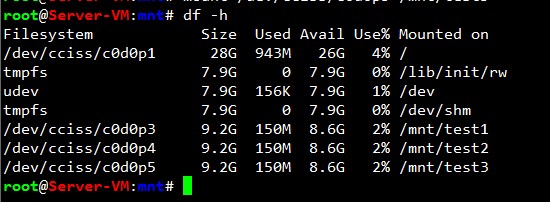
    突然发现找不到此分区，很奇怪

    5  查看已有的分区名称

[](http://static.oschina.net/uploads/space/2012/0606/180104_xYOv_123777.jpg)

    使用blkid查看已有的分区是什么name

    6  格式化后挂载新分区

[](http://static.oschina.net/uploads/space/2012/0606/180255_bjQr_123777.jpg)