# 磁盘分区

1. 磁盘安装好之后,利用命令查看磁盘是否被系统识别

[root@test100 ~]# fdisk -l

[root@test100 ~]# fdisk -l |grep 'sd[a-c]' (也可使用grep指定一个范围过滤)

Disk /dev/sda: 53.7 GB, 53687091200 bytes

/dev/sda1 \* 1 26 204800 83 Linux

/dev/sda2 26 217 1536000 82 Linux swap / Solaris

/dev/sda3 217 6528 50686976 83 Linux

Disk /dev/sdb: 107.4 GB, 107374182400 bytes #新添加的磁盘未分区

2. 根据磁盘的容量大小选择分区工具

1)大于或等于2TB 选择使用parteb命令

2)小于2TB 选择使用fdisk命令

## fdisk命令分区(小于2TB)

### 选择要分区的磁盘

(这里是新添加的第二块sdb磁盘100GB)

**分区参数说明**

|  |  |
| --- | --- |
| m | 显示帮助信息（了解） |
| l | 显示分区的类型（了解） |
| n | 增加一个新的分区 |
| d | 删除一个分区 |
| p | 输出分区表信息 |
| q | 退出不保存 |
| w | 修改分区表信息后，进行保存退出 |

[root@test100 ~]# fdisk -cu /dev/sdb #选择要分区的磁盘,此处为sdb

Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel

Building a new DOS disklabel with disk identifier 0x8c4da869.

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)

Command (m for help): n #增加一个分区

Command action

e extended(扩展分区)

p primary partition (1-4)(主分区)

p #先创建主分区

Partition number (1-4): 1 #选择序号,第一个主分区

First sector (2048-209715199, default 2048):

Using default value 2048

Last sector, +sectors or +size{K,M,G} (2048-209715199, default 209715199): +1G #选择大小

Command (m for help): p #查看一下是否分区成功

Disk /dev/sdb: 107.4 GB, 107374182400 bytes

255 heads, 63 sectors/track, 13054 cylinders, total 209715200 sectors

Units = sectors of 1 \* 512 = 512 bytes

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

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0xe48dbc0e

Device Boot Start End Blocks Id System

/dev/sdb1 2048 2099199 1048576 83 Linux

**至此.主分区已创建完毕!**

Command (m for help): n #继续增加一个分区

Command action

e extended

p primary partition (1-4)

e #选择创建扩展分区

Partition number (1-4): 2 #选择序号

First sector (2099200-209715199, default 2099200):

Using default value 2099200

Last sector, +sectors or +size{K,M,G} (2099200-209715199, default 209715199): #(不填写,则默认大小全部分配给扩展分区)

Using default value 209715199

Command (m for help): p #查看

Disk /dev/sdb: 107.4 GB, 107374182400 bytes

255 heads, 63 sectors/track, 13054 cylinders, total 209715200 sectors

Units = sectors of 1 \* 512 = 512 bytes

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

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0xe48dbc0e

Device Boot Start End Blocks Id System

/dev/sdb1 2048 2099199 1048576 83 Linux

/dev/sdb2 2099200 209715199 103808000 5 Extended

**至此.扩展分区,已创建完成!**

Command (m for help): n #增加分区

Command action

l logical (5 or over)

p primary partition (1-4)

l #选择创建逻辑分区

First sector (2101248-209715199, default 2101248):

Using default value 2101248

Last sector, +sectors or +size{K,M,G} (2101248-209715199, default 209715199): +20G #选择大小

Command (m for help): p #检查

Disk /dev/sdb: 107.4 GB, 107374182400 bytes

255 heads, 63 sectors/track, 13054 cylinders, total 209715200 sectors

Units = sectors of 1 \* 512 = 512 bytes

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

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0xe48dbc0e

Device Boot Start End Blocks Id System

/dev/sdb1 2048 2099199 1048576 83 Linux

/dev/sdb2 2099200 209715199 103808000 5 Extended

/dev/sdb5 2101248 44044287 20971520 83 Linux

**至此,逻辑分区已创建完成!**

Command (m for help): w #保存分区

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

**说明: 企业中服务器新增硬盘,直接创建一个扩展分区即可;分别创建逻辑分区,然后在执行以下操作(告知内核,格式化,取消自检,完成挂载)**

### 将分区情况告知内核

[root@test100 ~]# partprobe /dev/sdb

### 格式化创建文件系统

(centos6常见文件系统ext4 ext3 ;centos7常见文件系统xfs)

[root@test100 ~]# mkfs.ext4 /dev/sdb1 #格式化sdb1操作

mke2fs 1.41.12 (17-May-2010)

Filesystem label=

OS type: Linux

Block size=4096 (log=2)

Fragment size=4096 (log=2)

Stride=0 blocks, Stripe width=0 blocks

65536 inodes, 262144 blocks

13107 blocks (5.00%) reserved for the super user

First data block=0

Maximum filesystem blocks=268435456

8 block groups

32768 blocks per group, 32768 fragments per group

8192 inodes per group

Superblock backups stored on blocks:

32768, 98304, 163840, 229376

Writing inode tables: done

Creating journal (8192 blocks): done

Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 32 mounts or 提示:该文件系统将每32次挂载或180天自动检查一次

180 days, whichever comes first. Use tune2fs -c or -i to override. 无论哪个先来。可使用tune2fs -c或-i来覆盖

### 4) 取消自检操作

[root@oldgirl ~]# tune2fs -c 0 -i 0 /dev/sdb1 #取消自检

tune2fs 1.41.12 (17-May-2010)

Setting maximal mount count to -1

Setting interval between checks to 0 seconds

### 5) 完成磁盘挂载操作

[root@oldgirl ~]# mkdir /fdisk\_dir

[root@oldgirl ~]# mount /dev/sdb1 /fdisk\_dir/

[root@test100 ~]# df -h

Filesystem Size Used Avail Use% Mounted on

/dev/sda3 48G 1.9G 44G 5% /

tmpfs 491M 0 491M 0% /dev/shm

/dev/sda1 190M 35M 146M 19% /boot

/dev/sdb1 976M 1.3M 924M 1% /fdisk\_dir

**注意:记得设置开机自动挂载**

**方法一:将挂载命令放入到/etc/rc.local 文件中即可**

**方法二: vim /etc/fstab 编辑开机自动挂载文件(建议使用UUID进行挂载 (blkid命令,查看所有磁盘UUID))**

* 对于一个新的硬盘，为什么要进行分区操作？

　　这里，举个例子，假如你买了一套新房，是不是买的时候啥都没有，有的情况下，连厨房和厕所的空间都没有给你明确划分出来。这时，你是不是要自己找人来给你划分一下区域，这就与磁盘分区操作一样，需要人为进行分区操作。

* 分区后，为什么要进行格式化操作？

　　分区之后的硬盘，里面是不是什么都没有？这就与你的新房做好了划分一样，里面啥都没有。格式化的作用就是，往分区里面写一点点东西，便于支持文件操作(文件的创建，删除，修改等)。

* 格式化后的分区，为什么要挂载到目录下？

　　这里举个比较形象的例子。将格式化后的分区（我们接下来称之为源）比作水桶,而挂载操作就相当于给源接了一根水管，之后，我们要往源中写入数据，就是通过这跟水管进行的。如果没有这根水管，数据就存不到源中，也就是数据无法保存到硬盘里面。因此，格式化操作后的硬盘，必须挂载在一个文件目录下，才能对此分区进行数据的输写。这里，我们会有这样一种想法。既然一个源能够能够通过一根水管向其中写入数据，那么我们可不可以在这个水桶上接入多根水管，通过不同水管对同一段数据进行访问呢？答案是肯定的，因为，流入水桶中的数据最终肯定都在水桶里面，而连接到水桶的水管是不是都接到水桶上了，既然源都是一样的，当然我们可以通过不同水管源中的数据进行访问。

## parted磁盘分区(大于2TB)

随着生产环境中数据量的增大，我们对硬盘的容量也有很大的需求，当硬盘的容量大于2T（工业上的最大磁盘2.2TB）的时候，MSDOS分区(又叫Master Boot Record，简称MBR分区)就无法识别更大的容量空间了，这是时候我们就不得不用GPT （GUID Partition Table）分区，他最大支持18EB的容量,如果对这个EB单位不是很了解的话，对GB或者TB应该在熟悉不过了吧，我们可以用一个公式来换算：1 EB = 1024 PB = 1024 x 1024 TB = 1024 x 1024 x 1024GB，好了，只要知道很大就行。所以我们用GPT分区基本上不存在只有4个主分区这么一说了。

* fdisk 【只支持MSDOS分区布局】
* parted【支持MSDOS、GPT分区布局】分区操作实时生效,比较危险;因此最好在测试机操作,测试一下命令.

**分区参数说明**

|  |  |
| --- | --- |
| mklabel (mklabel msdos ;mklabel gpt) | 指定分区的引导记录信息 |
| mkpart | 制造一个分区 |
| print | 显示分区信息 |
| rm | 删除一个分区 |
| quit | 退出分区操作 |

### 对/dev/sdc分区进行操作

[root@localhost ~]# parted /dev/sdc

GNU Parted 3.1

Using /dev/sdb

Welcome to GNU Parted! Type 'help' to view a list of commands.

### 制定磁盘格式为gpt

(parted) mklabel

New disk label type? gpt

Warning: The existing disk label on /dev/sdc will be destroyed and all data on this diskwill be lost. Do you want to continue? #警告:/dev/sdc上的现有磁盘标签和磁盘上的所有数据将被销毁将丢失。你想继续吗?

Yes/No? y

### 查看磁盘分区信息

(parted) print

Model: VMware, VMware Virtual S (scsi)

Disk /dev/sdc: 3221GB

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

Partition Table: gpt

Number Start End Size File system Name Flags

### 制作分区名为primary分区

(parted) mkpart

Partition name? []? primary #自定义名称; 分区类型为主分区,这里没有扩展分区一说，都是主分区.

File system type? [ext2]? ext4 #指定文件系统格式为ext4

Start? 0 #设置起始分配位置

End? 1000GB #设置结束分配位置

Warning: The resulting partition is not properly aligned for best performance.

Ignore/Cancel? i #输入i，忽略此提示即可

### 查看新增分区后的状态

(parted) print

Model: VMware, VMware Virtual S (scsi)

Disk /dev/sdc: 3221GB

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

Partition Table: gpt

Number Start End Size File system Name Flags

1 17.4kB 1000GB 1000GB primary

### 删除分区1（Number号）

(parted) rm 1

### 继续新增分区名为primary 分区,起始磁盘1000GB,结束2000GB。

(parted) mkpart

Partition name? []? primary #自定义名称

File system type? [ext2]? ext4 #指定文件系统格式为ext4

Start? 1000GB #由于上块磁盘分区的结束位置是1000GB,所以我们从最1000GB开始分配即可

End? 2000GB #定义结束位置为2000GB

### 继续新增另一个分区名为primary分区，起始磁盘2000GB，结束100%大小。

(parted) mkpart primary 2000GB 100%

(parted) print #查看分区列表

Model: VMware, VMware Virtual S (scsi)

Disk /dev/sdc: 3221GB

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

Partition Table: gpt

Number Start End Size File system Name Flags

1 17.4kB 1000GB 1000GB primary

2 1000GB 2000GB 1000GB primary

3 2000GB 3221GB 1221GB primary

### 退出parted

(parted) quit

Information: You may need to update /etc/fstab.

### fdisk查看磁盘状态。能够发现新增磁盘分区状态。

[root@localhost ~]# fdisk -l

Disk /dev/sda: 53.7 GB, 53687091200 bytes

255 heads, 63 sectors/track, 6527 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: 0x000a5374

Device Boot Start End Blocks Id System

/dev/sda1 \* 1 26 204800 83 Linux

Partition 1 does not end on cylinder boundary.

/dev/sda2 26 217 1536000 82 Linux swap / Solaris

Partition 2 does not end on cylinder boundary.

/dev/sda3 217 6528 50686976 83 Linux

Disk /dev/sdb: 107.4 GB, 107374182400 bytes

255 heads, 63 sectors/track, 13054 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

WARNING: GPT (GUID Partition Table) detected on '/dev/sdc'! The util fdisk doesn't support GPT. Use GNU Parted.

Disk /dev/sdc: 3221.2 GB, 3221225472000 bytes

255 heads, 63 sectors/track, 391625 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

Device Boot Start End Blocks Id System

/dev/sdc1 1 267350 2147483647+ ee GPT

[root@test100 /]# parted -l

Model: VMware, VMware Virtual S (scsi)

Disk /dev/sda: 53.7GB

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

Partition Table: msdos

Number Start End Size Type File system Flags

1 1049kB 211MB 210MB primary ext4 boot

2 211MB 1784MB 1573MB primary linux-swap(v1)

3 1784MB 53.7GB 51.9GB primary ext4

Model: VMware, VMware Virtual S (scsi)

Disk /dev/sdb: 107GB

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

Partition Table: loop

Number Start End Size File system Flags

1 0.00B 107GB 107GB ext4

Model: VMware, VMware Virtual S (scsi)

Disk /dev/sdc: 3221GB

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

Partition Table: gpt

Number Start End Size File system Name Flags

1 17.4kB 1000GB 1000GB primary

2 1000GB 2000GB 1000GB primary

3 2000GB 3221GB 1221GB primary

### 新增完成后，格式化分区。

[root@test100 /]# mkfs.ext4 /dev/sdc1

[root@test100 /]# mkfs.ext4 /dev/sdc2

[root@test100 /]# mkfs.ext4 /dev/sdc3

### 修改/etc/fstab实现自动挂载

建议使用UUID进行挂载 (blkid命令,查看所有磁盘UUID)

[root@test100 /]# blkid /dev/sdc1

/dev/sdc1: UUID="a40cc3d2-ec33-4ef6-a419-f32070bfa55c" TYPE="ext4"

[root@test100 /]# vim /etc/fstab

a40cc3d2-ec33-4ef6-a419-f32070bfa55c /opt ext4 defaults 0 0