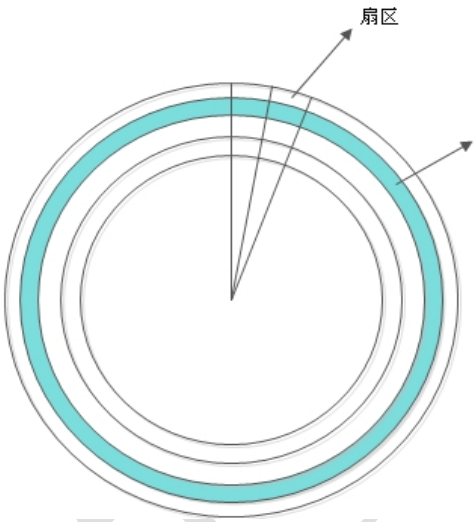
第九章：文件系统管理

尚硅谷云计算 Linux 课程

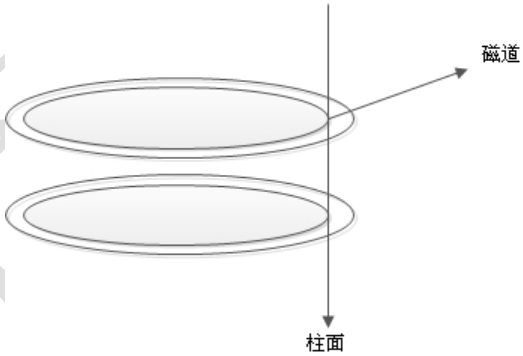
版本：V1.0

一、硬盘结构

1、硬盘的逻辑结构



每个扇区的大小事固定的，为 512Byte。扇区也是磁盘的最小存贮单位。



硬盘的大小是使用“磁头数×柱面数×扇区数×每个扇区的大小”这样的公式来计算的。其中磁头数（Heads）表示硬盘总共有几个磁头，也可以理解成为硬盘有几个盘面，然后乘以二；柱面数（Cylinders）表示硬盘每一面盘片有几条磁道；扇区数（Sectors）表示每条磁道上有几个扇区；每个扇区的大小一般是 512Byte。

2、硬盘接口

- IDE 硬盘接口（Integrated Drive Electronics，并口，即电子集成驱动器）也称作“ATA 硬盘”或“PATA 硬盘”，是早期机械硬盘的主要接口，ATA133 硬盘的理论速度可以达到 133MB/s（此速度为理论平均值），IDE 硬盘接口

- SATA 接口（Serial ATA，串口）是速度更高的硬盘标准，具备了更高的传输速度，并具备了更强的纠错能力。目前已经是 SATA 三代，理论传输速度达到 600MB/s（此速度为理论平均值）
- SCSI 接口（Small Computer System Interface，小型计算机系统接口）广泛应用在服务器上，具有应用范围广、多任务、带宽大、CPU 占用率低及支持热插拔等优点，理论传输速度达到 320MB/s

二、文件系统

1. Linux 文件系统的特性：

✧ super block（超级块）：记录整个文件系统的信息，包括 block 与 inode 的总量，已经使用的 inode 和 block 的数量，未使用的 inode 和 block 的数量，block 与 inode 的大小，文件系统的挂载时间，最近一次的写入时间，最近一次的磁盘检验时间等。

✧ date block（数据块，也称作 block）：用来实际保存数据的（柜子的隔断），block 的大小（1KB、2KB 或 4KB）和数量在格式化后就已经决定，不能改变，除非重新格式化（制作柜子的时候，隔断大小就已经决定，不能更改，除非重新制作柜子）。每个 blcok 只能保存一个文件的数据，要是文件数据小于一个 block 块，那么这个 block 的剩余空间不能被其他文件是要；要是文件数据大于一个 block 块，则占用多个 block 块。Windows 中磁盘碎片整理工具的原理就是把一个文件占用的多个 block 块尽量整理到一起，这样可以加快读写速度。

✧ inode（i 节点，柜子门上的标签）：用来记录文件的权限（r、w、x），文件的所有者和属组，文件的大小，文件的状态改变时间（ctime），文件的最近一次读取时间（atime），文件的最近一次修改时间（mtime），文件的数据真正保存的 block 编号。每个文件需要占用一个 inode。

2. Linux 常见文件系统

文件系统	描　　述
ext	Linux 中最早的文件系统，由于在性能和兼容性上具有很多缺陷，现在已经很少使用
ext2	是 ext 文件系统的升级版本，Red Hat Linux 7.2 版本以前的系统默认都是 ext2 文件系统。于 1993 年发布，支持最大 16TB 的分区和最大 2TB 的文件（1TB=1024GB=1024×1024KB）
ext3	是 ext2 文件系统的升级版本，最大的区别就是带日志功能，以便在系统突然停止时提高文件系统的可靠性。支持最大 16TB 的分区和最大 2TB 的文件
ext4	是 ext3 文件系统的升级版。ext4 在性能、伸缩性和可靠性方面进行了大量改进。ext4 的变化可以说是翻天覆地的，比如向下兼容 ext3、最大 1EB 文件系统和 16TB 文件、无限数量子目录、Extents 连续数据块概念、多块分配、延迟分配、持久预分配、快速 FSCK、日志校验、无日志模式、在线碎片整理、inode 增强、默认启用 barrier 等。它是 CentOS 6.x 的默认文件系统
xfs	XFS 最早针对 IRIX 操作系统开发，是一个高性能的日志型文件系统，能够在断电以及操作系统崩溃的情况下保证文件系统数据的一致性。它是一个 64 位的文件系统，后来进行开源并且移植到了 Linux 操作系统中，目前 CentOS 7.x 将 XFS+LVM 作为默认的文件系统。据官方所称，XFS 对于大文件的读写性能较好。

swap	swap 是 Linux 中用于交换分区的文件系统（类似于 Windows 中的虚拟内存），当内存不够用时，使用交换分区暂时替代内存。一般大小为内存的 2 倍，但是不要超过 2GB。它是 Linux 的必需分区
NFS	NFS 是网络文件系统（Network File System）的缩写，是用来实现不同主机之间文件共享的一种网络服务，本地主机可以通过挂载的方式使用远程共享的资源
iso9660	光盘的标准文件系统。Linux 要想使用光盘，必须支持 iso9660 文件系统
fat	就是 Windows 下的 fat16 文件系统，在 Linux 中识别为 fat
vfat	就是 Windows 下的 fat32 文件系统，在 Linux 中识别为 vfat。支持最大 32GB 的分区和最大 4GB 的文件
NTFS	就是 Windows 下的 NTFS 文件系统，不过 Linux 默认是不能识别 NTFS 文件系统的，如果需要识别，则需要重新编译内核才能支持。它比 fat32 文件系统更加安全，速度更快，支持最大 2TB 的分区和最大 64GB 的文件
ufs	Sun 公司的操作系统 Solaris 和 SunOS 所采用的文件系统
proc	Linux 中基于内存的虚拟文件系统，用来管理内存存储目录/proc
sysfs	和 proc 一样，也是基于内存的虚拟文件系统，用来管理内存存储目录/sysfs
tmpfs	也是一种基于内存的虚拟文件系统，不过也可以使用 swap 交换分区

三、常用的硬盘管理命令

1、df 命令

```
[root@localhost ~]# df  -ahT
#-a  显示特殊文件系统，这些文件系统几乎都是保存在内存中的。如/proc，因为是挂载在内存中，所以占用量都是 0
#-h 单位不再只用 KB，而是换算成习惯单位
#-T  多出了文件系统类型一列
```

2、du 命令

```
[root@localhost ~]# du [选项] [目录或文件名]
选项：
    -a  显示每个子文件的磁盘占用量。默认只统计子目录的磁盘占用量
    -h  使用习惯单位显示磁盘占用量，如 KB，MB 或 GB 等
    -s  统计总占用量，而不列出子目录和子文件的占用量
```

du 与 df 的区别：du 是用于统计文件大小的，统计的文件大小是准确的；df 是用于统计空间大小的，统计的剩余空是准确的

lsof | grep deleted”查看被删除的文件，然后一个进程一个进程的手工 kill 也是可以的

3、fsck 文件系统修复命令

```
[root@localhost ~]# fsck  -y /dev/sdb1
#自动修复
```

4、显示磁盘状态 dumpe2fs

```
[root@localhost ~]# dumpe2fs /dev/sda3
dumpe2fs 1.41.12 (17-May-2010)
Filesystem volume name:   <none>            ←卷标名
Last mounted on:          /                 ←挂载点
Filesystem UUID:          c2ca6f57-b15c-43ea-bca0-f239083d8bd2  ←UUID
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem  features:           has_journal  ext_attr  resize_inode  dir_index  filetype
needs_recovery extent flex_bg spars
e_super large_file huge_file uninit_bg dir_nlink extra_isize
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl    ←挂载参数
Filesystem state:         clean             ←文件系统状态，正常
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              1826816           ←inode 总数
Block count:              7300864           ←块总素
Reserved block count:     365043
Free blocks:              6634637
Free inodes:              1753533
First block:              0
Block size:               4096              ←块大小
Fragment size:            4096
Reserved GDT blocks:      1022
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         8192
Inode blocks per group:   512
Flex block group size:    16
Filesystem created:       Mon Nov 12 22:30:41 2012
Last mount time:          Tue Apr  9 23:53:29 2013
Last write time:          Mon Nov 12 22:45:55 2012
Mount count:              3
Maximum mount count:      -1
Last checked:             Mon Nov 12 22:30:41 2012
Check interval:           0 (<none>)
Lifetime writes:          3199 MB
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               256               ←inode 的大小
```

```
…省略部分输出…

Group 0: (Blocks 0-32767) [ITABLE_ZEROED]    ←第一个数据组的内容
  校验和 0xcb85,8179 个未使用的 inode
  主 superblock at 0, Group descriptors at 1-2
  保留的 GDT 块位于 3-1024
  Block bitmap at 1025 (+1025), Inode bitmap at 1041 (+1041)
  Inode 表位于 1057-1568 (+1057)
  23513 free blocks, 8179 free inodes, 2 directories, 8179 个未使用的 inodes
  可用块数: 9255-32767
  可用 inode 数: 14-8192
…省略部分输出…
```

5、查看文件的详细时间

stat 文件名 查看文件的详细时间

```
例如
[root@localhost ~]# stat test.sh
  File: `test.sh'
    #文件名
  Size: 427             Blocks: 8          IO Block: 4096   regular file
    #文件大小        占用块          系统分区块大小
Device: fd00h/64768d    Inode: 23724038    Links: 1
    #存放文件的设备 inode 号     硬链接数
Access: (0755/-rwxr-xr-x)  Uid: (    0/    root)   Gid: (  0/  root)
    权限            属主                属组
Access: 2011-10-29 22:27:45.000000000 +0800
Modify: 2011-09-14 18:02:00.000000000 +0800
Change: 2011-10-25 22:21:44.000000000 +0800
            access      访问时间
            modify      数据修改时间
            change      状态修改时间
```

6、判断文件类型

file 文件名 判断文件类型

type 命令名 判断命令类型

四、fdisk 命令手工分区

1 fdisk -l

查看系统所有硬盘及分区

2 fdisk /dev/sdb 进行磁盘分区（分区还没有分区号）

fdisk 交互指令说明

命令	说明
a	设置可引导标记
b	编辑 bsd 磁盘标签
c	设置 DOS 操作系统兼容标记
d	删除一个分区
l	显示已知的文件系统类型。82 为 Linux swap 分区，83 为 Linux 分区
m	显示帮助菜单
n	新建分区
o	建立空白 DOS 分区表
p	显示分区列表
q	不保存退出
s	新建空白 SUN 磁盘标签
t	改变一个分区的系统 ID
u	改变显示记录单位
v	验证分区表
w	保存退出
x	附加功能（仅专家）

n---p 主----1 分区号----1 起始柱面-----分区大小+100M-----w
n---e 扩展---2 分区号---124 起始柱面----1024 柱面（所有剩余空间都分配给扩展分区）
n---l 逻辑---不用指定分区号---124 起始柱面----+100M(指定大小)-----w

有时因为系统的分区表**正忙**，则需要**重新启动系统**之后才能使新的分区表生效。

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

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16:


Device or resource busy.
The kernel still uses the old table.
The new table will be used at the next reboot.   ←要求重启动，才能格式化
Syncing disks.
```

3 partprobe 强制重读所有分区文件，重新挂载分区文件内所有分区。这不是分区必须命令，如果没有提示重启，可以不执行，也可以重启系统

（Warning: Unable to open /dev/hdc read-write (Read-only file system). /dev/hdc has been opened read-only.

光盘只读挂载，不是错误，不用紧张）

如果这个命令不存在请安装 parted-2.1-18.el6.i686 这个软件包

4　格式化　　建立文件系统　　ext3 是 linux 默认文件系统

```
mkfs  -t  ext4  /dev/sdb1
mkfs  -t  ext4  /dev/sdb5
```

mkfs 命令非常简单易用，不过是不能调整分区的默认参数的（比如块大小是 4096），这些默认参数除非特殊情况，否则不需要调整，如果想要调整就需要使用 mke2fs 命令进行重新格式化，命令格式如下：

```
[root@localhost ~]# mke2fs [选项] 分区设备文件名
选项：
    -t 文件系统：    指定格式化成哪个文件系统，如 ext2，ext3，ext4
    -b 字节：        指定 block 块的大小
    -i 字节：        指定“字节/inode”的比例，也就是多少个字节分配一个 inode
    -j：             建立带有 ext3 日志功能的文件系统
    -L 卷标名：      给文件系统设置卷标名，就不使用 e2label 命令设定了

举个例子：
[root@localhost ~]# mke2fs -t ext4  -b 2048  /dev/sdb6
#格式化分区，并指定 block 的大小为 2048
```

5　建立挂载点

```
mkdir  /disk1----------------/dev/sdb1     把 sdb1 打算挂载到/disk1 目录中
mkdir  /disk5----------------/dev/sdb5
```

6　挂载

```
mount  /dev/sdb1  /disk1
mount  /dev/sdb5  /disk5
```

7　查看

```
mount      查看所有已经挂载的分区和光盘
fdisk  -l  查看系统分区
df         查看分区占用百分比
```

8　自动挂载

修改分区自动挂载文件

```
vi /etc/fstab                注意：此文件直接参与系统启动，如果修改错误，系统启动报错

/dev/sdb1              /disk1                  ext3    defaults        1 2
```

第一列：　设备文件名
第二列　　挂载点
第三列　　文件系统
第四列　　挂载选项

第五列	1	是否可以被备份	0 不备份	1 每天备份 2 不定期备份
第六列	2	是否检测磁盘 fsck	0 不检测	1 启动时检测 2 启动后检测

也可以使用 UUID 进行挂载，UUID（硬盘通用唯一识别码，可以理解为硬盘的 ID）

- 这个字段在 CentOS 5.5 的系统当中是写入分区的卷标名或分区设备文件名的，现在变成了硬盘的 UUID。这样做的好处是当硬盘增加了新的分区，或者分区的顺序改变，再或者内核升级后，任然能够保证分区能够正确的加载，而不至于造成启动障碍
- 那么每个分区的 UUID 到底是什么呢？我们讲过的 dumpe2fs 命令是可以查看到的，命令如下：

```
[root@localhost ~]# dumpe2fs /dev/sdb5
```

或

```
[root@localhost ~]# ls -l /dev/disk/by-uuid/
```

9 重启测试

或 mount -a 重新挂载所有内容，用它进行测试

五、/etc/fstab/文件修复

我们重新启动系统吧。苍天啊，大地啊，哪位神仙姐姐显灵了啊，真的报错了，如图 9-16 所示：

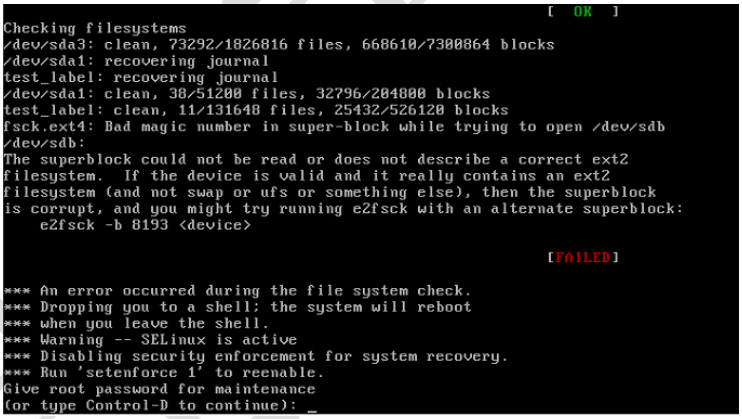


图 9-16 系统启动报错

先别哭天抹泪了，仔细看看，他提示你输入 root 密码啊，好像还有戏啊，我们输入密码试试，如图 9-17 所示：

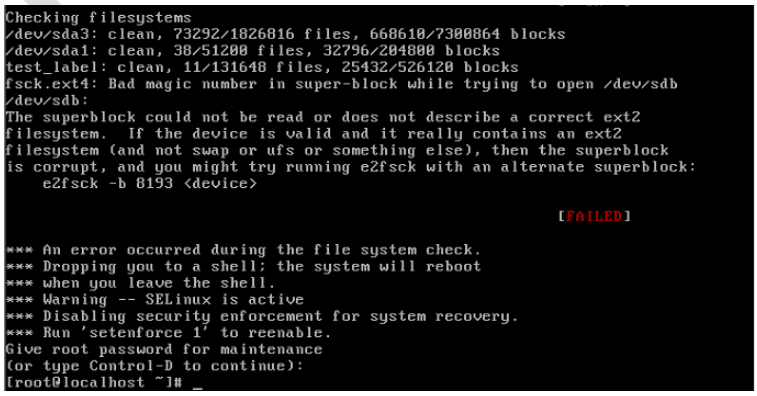
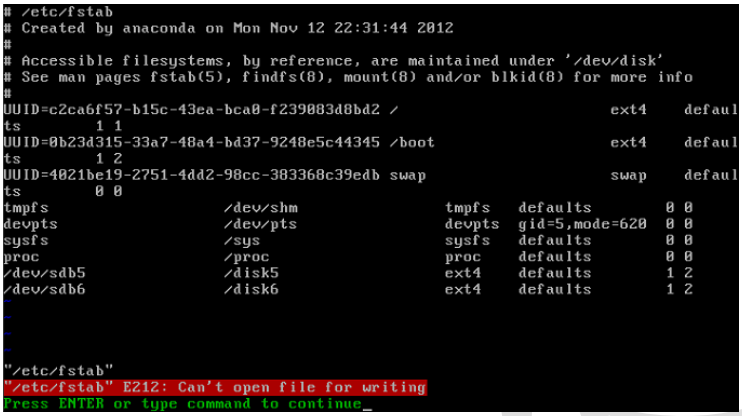


图 9-17 root 登陆

啊！我们又看到了系统提示符，赶快把/etc/fstab 文件修改回来吧。晕，报错了，如图 9-18 所示：



9-18 修改/etc/fstab 报错

别慌，分析下原因提示是没有写权限，那么只要把/分区重新挂载下，挂载为读写权限不就可以修改了吗？命令如下

```
[root@localhost ~]# mount -o remount,rw /
```

再去修改/etc/fstab 文件，把它改回来就可以正常启动了啊。

六、parted 命令分区

我们 Linux 系统中有两种常见的分区表 MBR 分区表（主引导记录分区表）和 GPT 分区表（GUID 分区表），其中：

- ✧ MBR 分区表：支持的最大分区是 2TB（1TB=1024GB）；最多支持 4 个主分区，或 3 个主分区 1 个扩展分区
- ✧ GPT 分区表：支持最大 18EB 的分区（1EB=1024PB=1024*1024TB）；最多支持 128 个分区，其中 1 个系统保留分区，127 个用户自定义分区

不过 parted 命令也有点小问题，就是命令自身分区的时候只能格式化成 ext2 文件系统，不支持 ext3 文件系统，那就更不用说 ext4 文件系统了（截止到 CentOS 6.8 还是这样，这里只是指不能用 parted 命令把分区格式化成 ext4 文件系统，但是 parted 命令还是可以识别 ext4 文件系统的）。不过这没有太多的影响，因为我们可以先分区再用 mkfs 进行格式化嘛！

一）parted 交互模式

```
[root@localhost ~]# parted /dev/sdb
#打算继续划分/dev/sdb 硬盘
GNU Parted 2.1
使用 /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)                                  ←parted 的等待输入交互命令的位置
```

parted 交互命令	说　　明
check NUMBER	做一次简单的文件系统检测
cp [FROM-DEVICE] FROM-NUMBER TO-NUMBER	复制文件系统到另外一个分区
help [COMMAND]	显示所有的命令帮助
mklabel,mktable LABEL-TYPE	创建新的磁盘卷标（分区表）
mkfs NUMBER FS-TYPE	在分区上建立文件系统

mkpart PART-TYPE [FS-TYPE] START END	创建一个分区
mkpartfs PART-TYPE FS-TYPE START END	创建分区，并建立文件系统
move NUMBER START END	移动分区
name NUMBER NAME	给分区命名
print [devices|free|list,all|NUMBER]	显示分区表，活动设备，空闲空间，所有分区
quit	退出
rescue START END	修复丢失的分区
resize NUMBER START END	修改分区大小
rm NUMBER	删除分区
select DEVICE	选择需要编辑的设备
set NUMBER FLAG STATE	改变分区标记
toggle [NUMBER [FLAG]]	切换分区表的状态
unit UNIT	设置默认的单位
Version	显示版本

二）建立分区

1、 查看分区

```
(parted) print
#输入 print 指令
Model: VMware, VMware Virtual S (scsi)          ←硬盘参数，是虚拟机啊
Disk /dev/sdb: 21.5GB                           ←硬盘大小
Sector size (logical/physical): 512B/512B       ←扇区大小
Partition Table: msdos                          ←分区表类型，就是 MBR 分区表

Number  Start   End     Size    Type      File system  标志
 1      32.3kB  5379MB  5379MB  primary
 2      5379MB  21.5GB  16.1GB  extended
 5      5379MB  7534MB  2155MB  logical   ext4
 6      7534MB  9689MB  2155MB  logical   ext4
#看到了我们使用 fdisk 分的区，其中 1 分区没有格式化，2 分区是扩展分区不能格式化
```

使用 print 可以查看分区表信息，包括硬盘参数，硬盘大小，扇区大小，分区表类型和分区信息。分区信息总共七列，分别是：

- ✧ Number：分区号
- ✧ Start：分区起始位置，这里不再像 fdisk 用柱面表示，而是使用 Byte 更加直观
- ✧ End：分区结束位置
- ✧ Size：分区大小
- ✧ Type：分区类型
- ✧ File system：文件系统类型。我不是说 parted 不支持 ext4 文件系统吗？注意，我一直都是说 parted 不能直接把分区直接格式化成 ext4 文件系统，但是它是可以识别的。
- ✧ 标志：Flags，就是分区的标记

2、修改成 GPT 分区表

```
(parted) mklabel gpt
#修改分区表命令
警告: 正在使用 /dev/sdb 上的分区。          ←由于/dev/sdb 分区已经挂载，所以有警告
                                            ←注意如果强制修改，原有分区及数据会消失
忽略/Ignore/放弃/Cancel? ignore             ←输入 ignore 忽略报错
警告: The existing disk label on /dev/sdb will be destroyed and all data on this disk will
be lost. Do you want to continue?
是/Yes/否/No? yes                           ←输入 yes
警告: WARNING: the kernel failed to re-read the partition table on /dev/sdb (设备或资源
忙).  As a result, it may not reflect all of
your changes until after reboot.            ←下次重启后，才能生效

(parted) print                              ←查看下分区表吧
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdb: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt                        ←分区表已经变成 GPT

Number  Start  End  Size  File system  Name  标志
                                            ←所有的分区都消失了
```

修改了分区表，如果这块硬盘已经有分区了，那么原有分区和分区中的数据都会消失，而且需要重启系统才会生效。

还有我们转换分区表的目的是为了支持大于 2TB 的分区，如果分区并没有大于 2TB，那么这步是可以不执行的。

注意：一定要把/etc/fstab 文件中和原有分区的内容删除掉，才能重启动。要不系统启动就一定会报错的。

3、 建立分区

因为修改过了分区表，所以/dev/sdb 中的所有数据都消失了，所以我们可以重新对这块硬盘分区了。不过建立分区时，默认文件系统就只能建立成 ext2 了，命令如下：

```
(parted) mkpart
#输入创建分区命令，后面不要参数，全部靠交互指定
分区名称?  []? disk1                        ←分区名称，我起名叫 disk1
文件系统类型?  [ext2]?                      ←文件系统类型，直接回车，使用默认 ext2
起始点? 1MB                                 ←分区从 1MB 开始
结束点? 5GB                                 ←分区到 5GB 结束
#分区完成
(parted) print                              ←查看下吧
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdb: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
```

```
Number  Start   End     Size    File system  Name  标志
1       1049kB  5000MB  4999MB               disk1          ←分区 1 已经出现
```

不知道大家注意到了吗？我们现在的 print 查看的分区，和第一次查看 MBR 分区表的分区时有些不一样了，少了 Type 这个字段，也就是分区类型的字段，多了 Name 分区名字段。分区类型是标识主分区、扩展分区和逻辑分区的，不过这种标识只在 MBR 分区表中使用，我们现在已经变成了 GPT 分区表了，所以就不再有 Type 类型了。也就说折磨我们很久的主分区、扩展分区和逻辑分区的概念不再有用了，阿门！

4、 建立文件系统

分区分完了，我们还需要格式化。不过我们已经知道如果使用 parted 交互命令格式化的话，只能格式化成 ext2 文件系统。我们这里是要演示下 parted 命令的格式化方法，所以就格式化成 ext2 吧，命令如下：

```
(parted) mkfs
#格式化命令（很奇怪也是 mkfs，但是这只是 parted 的交互命令）
WARNING: you are attempting to use parted to operate on (mkfs) a file system.
parted's file system manipulation code is not as robust as what you'll find in
dedicated, file-system-specific packages like e2fsprogs.  We recommend
you use parted only to manipulate partition tables, whenever possible.
Support for performing most operations on most types of file systems
will be removed in an upcoming release.
警告: The existing file system will be destroyed and all data on the partition will be lost.
Do you want to continue?
是/Yes/否/No? yes                          ←警告你格式化数据会丢失，没关系，已经丢失过了
分区编号？ 1
文件系统类型？  [ext2]?                    ←指定文件系统类型，写别的也没有用，直接回车

(parted) print                             ←格式化完成，查看下
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdb: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End     Size    File system  Name   标志
 1      1049kB  5000MB  4999MB  ext2         disk1          ←拥有了文件系统
```

如果要格式化成 ext4 文件系统，请 mkfs 命令帮忙吧（注意不是 parted 交互命令中的 mkfs，而是系统命令 mkfs）！

5、 调整分区大小

parted 命令还有一大优势，就是可以调整分区的大小（windows 中也可以实现，不过要不需要转换成动态磁盘，要不需要依赖第三方工具，如硬盘分区魔术师）。起始 Linux 中 LVM 和 RAID 是可以支持分区调整的，不过这两种方法也可以看成是动态磁盘方法，我们在下一个章节中介绍。使用 parted 命令调整分区要更加简单。

注意：parted 调整已经挂载使用的分区时，是不会影响分区中的数据的，也就是说数据不

会丢失。但是一定要**先卸载分区**，再调整分区大小，否则数据是会出现问题的。还有要调整大小的分区必须已经建立了文件系统（格式化），否则会报错

命令如下：

```
(parted) resize
分区编号？ 1                              ←指定要修改的分区编号
起始点？  [1049kB]? 1MB                   ←分区起始位置
结束点？  [5000MB]? 6GB                   ←分区结束位置
(parted) print                            ←查看下
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdb: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End     Size    File system  Name    标志
 1      1049kB  6000MB  5999MB  ext2         disk1              ←分区大小改变
```

6、 删除分区

```
(parted) rm
#删除分区命令
分区编号？ 1                              ←指定分区号
(parted) print                            ←查看下
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdb: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start  End  Size  File system  Name  标志              ←分区消失
```

还有件事要注意，parted 中所有的操作都是立即生效，没有保存生效的概念。这点和 fdisk 交互命令明显不同，所以所做的所有操作大家要倍加小心。

那么到底使用 fdisk 进行分区？还是 parted 命令呢？这个完全看个人习惯，我们更加习惯 fdisk 命令。

七、分配 swap 分区

1、分区，并修改为 swap 分区 ID

```
[root@localhost ~]# fdisk /dev/sdb
#拿/dev/sdb 分区
Command (m for help): t                   ←修改分区的系统 ID
Selected partition 1                      ←只有一个分区，所以不用选择分区了
Hex code (type L to list codes): 82       ←改为 swap 的 ID
Changed system type of partition 1 to 82 (Linux swap / Solaris)
```

2、格式化

```
[root@localhost ~]# mkswap /dev/sdb1
Setting up swapspace version 1, size = 522076 KiB
no label, UUID=c3351dc3-f403-419a-9666-c24615e170fb
```

9.8.3 使用 swap 分区

在使用 swap 分区之前，我们先来说说 free 命令，命令如下：

```
[root@localhost ~]# free
             total       used       free     shared    buffers     cached
Mem:       1030796     130792     900004          0      15292      55420
-/+ buffers/cache:      60080     970716
Swap:      2047992          0    2047992
```

```
[root@localhost ~]# swapon 分区设备文件名

例：
[root@localhost ~]# swapon /dev/sdb1
```

让 swap 分区开机之后自动挂载

```
/dev/sdb1                swap                    swap    defaults        0 0
#加入新 swap 分区的相关内容，我这里是直接使用的分区的设备文件名，大家当然也可以
#使用 UUID 号了。
```