21.磁盘和文件系统管理

一、磁盘介绍与管理

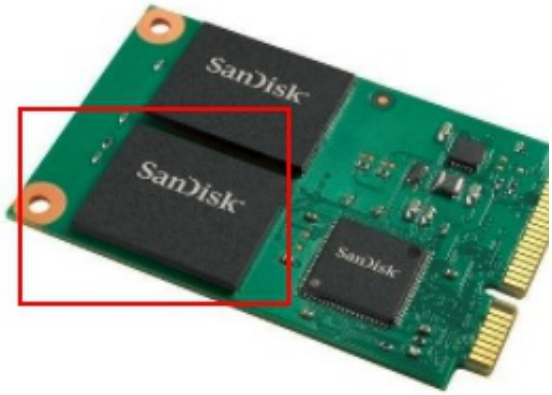
当下流行的磁盘种类

常见硬盘品牌：希捷　西数　日立　HP　DELL　EMC　IBM

硬盘分几种？

1）从工作原理来说：

固态：价格相对贵，寿命长，读取速度快



机械：怕摔、怕磁，（单位换下来的坏盘会做消磁处理），读取速度===磁道寻址时间，潜伏时间

2）从硬盘的接口来说

SATA：用在低端服务器多，理论传输速度达到 600MB/s

SAS3：用在中高服务器，，理论传输速度达到1200MB/s

PCIE 版本5： 理论传输速度达到32 or 25GT/s，mac上用，当然也最贵

SCSI 和 IDE

SAS硬盘：

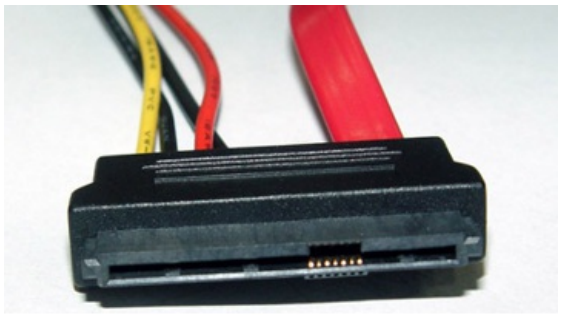
SAS（Serial Attached SCSI），串行连接SCSI接口，串行连接小型计算机系统接口。

SAS是新一代的SCSI技术，和现在流行的Serial ATA(SATA)硬盘相同，都是采用串行技术以获得更高的传输速度。

SAS在接口技术上和SATA是向下兼容的。

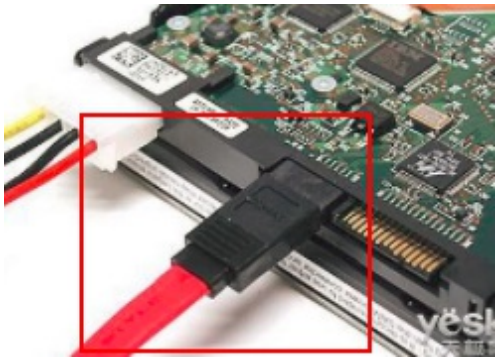
翻个面，也有链接触点：

SAS 磁盘线：



SATA硬盘：

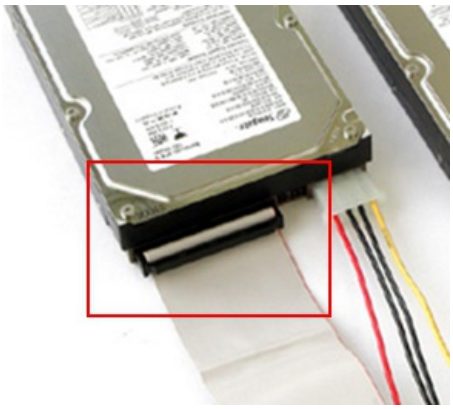
SATA硬盘数据线：



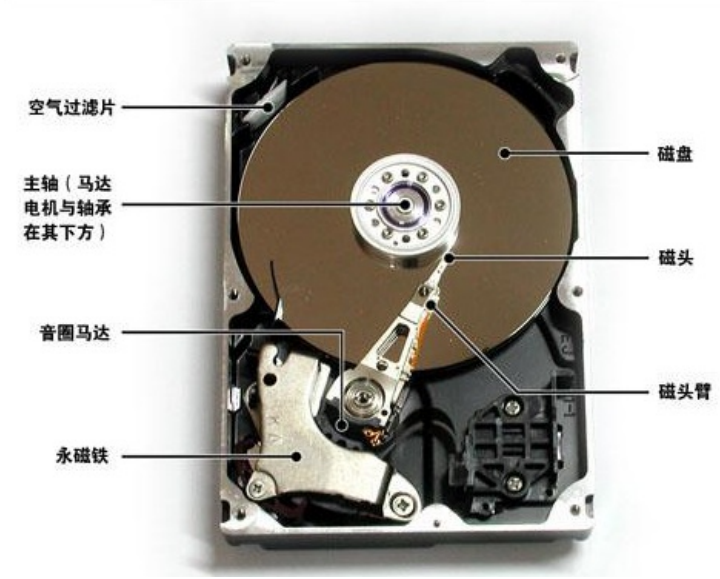
PCIE硬盘

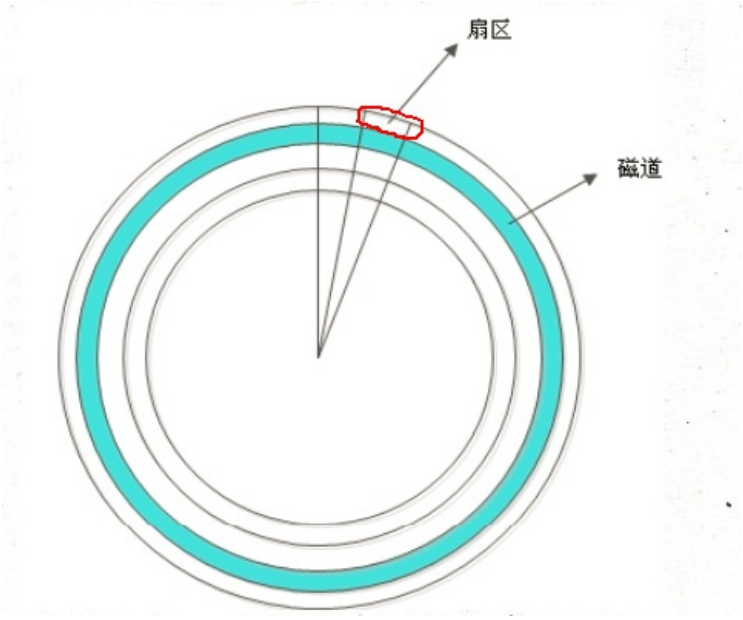
SCSI硬盘，已淘汰：

IDE硬盘，已淘汰：

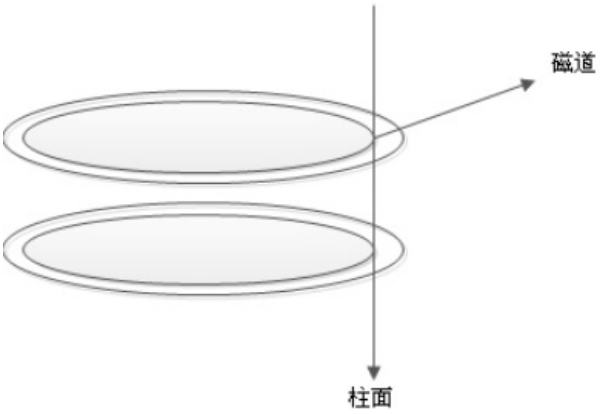


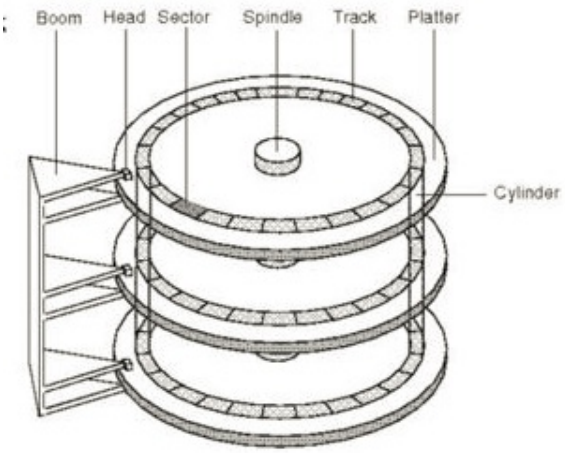
2、硬盘结构





每个扇区的大小事固定的，为 512Byte（相当于0.5KB）。扇区也是磁盘的最小存贮单位。





硬盘的内部是金属盘片，将圆形的盘片划分成若干个扇形区域，这就是扇区。若干个扇区就组成整个盘片。为什么要分扇区？是逻辑化数据的需要，能更好的管理硬盘空间。 以盘片中心为圆心，把盘片分成若干个同心圆，那每一个划分圆的“线条”，就称为磁道。

硬盘内的盘片有两个面，都可以储存数据，而硬盘内的盘片往往不止一张，常见的有两张，那么，两张盘片中相同位置的磁道，就组成一个“柱面”，盘片中有多少个磁道，就有多少个柱面。盘片两面都能存数据，要读取它，必须有磁头，所以，每一个面，都有一个磁头，一张盘片就有两个磁头。

硬盘的存储容量=磁头数×磁道（柱面）数×每磁道扇区数×每道扇区字节数。

磁道从外向内自0开始顺序进行编号，各个磁道上的扇区数是在硬盘格式化时确定的。

windows安装系统的C盘或Linux boot分区一般安装在磁盘最外面还是最里面？

windows ： C盘安装最外，速度也是最快

Linux ： boot分区和 swap分区，装最外面

磁盘写数据时，先从外面往里。引导数据一般保存在0磁道0扇区里！系统启动都需要这个引导区里的引导数据来引导系统启动。

3、磁盘常用管理命令

1、df 命令

```
# 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

```
# 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

3、fdisk -l命令

查看系统中有多少块磁盘，以及没块磁盘的分区情况

二、Linux文件系统结构

1、常见的linux文件系统：

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（逻辑卷管理）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。单个文件系统最大可以支持8EB，单个文件可以支持16TB

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/sys

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

2、Linux文件系统具体由三部分组成 ： 文件名，inode，block

inode的内容

inode包含文件的元信息，相当于windows的文件的属性：

*** 文件的字节数**

* 文件拥有者的User ID
* 文件的Group ID
* 文件的读、写、执行权限
* 文件的时间戳，共有三个：
    ctime指inode上一次变动的时间，
    mtime指文件内容上一次变动的时间，
    atime指文件上一次打开的时间。
* 链接数，即有多少文件名指向这个inode
* 文件数据block的位置

可以用stat命令，查看某个文件的inode信息

inode的大小

inode也会消耗硬盘空间，所以硬盘格式化的时候，操作系统自动将硬盘分成两个区域。一个是数据区，存放文件数据；另一个是inode区（inode table），存放inode所包含的信息。lixun的文件系统中把这部分区域又叫super block（超级块）。

每个inode节点的大小，一般是128字节或256字节。inode节点的总数，在格式化时就给定，假定在一块1GB的硬盘中，每个inode节点的大小为128字节，每1KB就设置一个inode，那么inode table的大小就会达到128MB，占整块硬盘的12.8%。

inode号码

每个inode都有一个号码，操作系统用inode号码来识别不同的文件。

Unix/Linux系统内部不使用文件名，而使用inode号码来识别文件。对于系统来说，文件名只是inode号码便于识别的别称或者绰号。表面上，用户通过文件名，打开文件。实际上，系统内部这个过程分成三步：

首先，系统找到这个文件名对应的inode号码；

其次，通过inode号码，获取inode信息；

最后，根据inode信息，找到文件数据所在的block，读出数据。

使用ls -i命令，可以看到文件名对应的inode号码

使用 df -i命令，查看每个硬盘分区的inode总数和已经使用的数量，可以使用df命令。

```
# df -i
Filesystem            Inodes   IUsed   IFree         IUse% Mounted on
/dev/sda2             640848   151010  489838        24%       /
```

注：由于每个文件都必须有一个inode，因此有可能发生inode已经用光，但是硬盘还未存满的情况，无法在硬盘上创建新文件。

目录文件

Unix/Linux系统中，目录（directory）也是一种文件。打开目录，实际上就是打开目录文件。

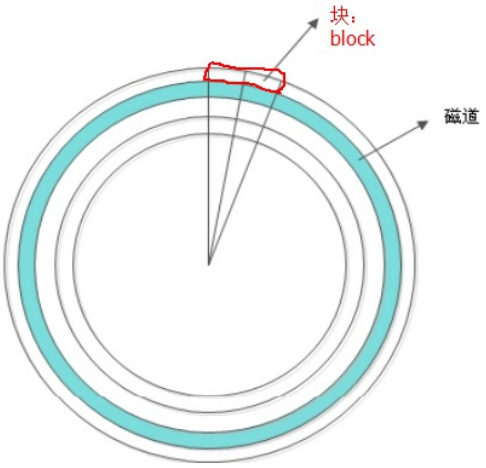
目录文件的结构非常简单，就是一系列目录项的列表。

每个目录项，由两部分组成：所包含文件的文件名，以及该文件名对应的inode号码。

```
# ls -id /etc
8388673 /etc
```

block

操作系统读取硬盘的时候，不会一个个扇区（512字节）地读取，这样效率太低，而是一次性连续读取多个扇区，即一次性读取一个"块"（block）。这种由多个扇区组成的"块"，是文件存取的最小单位。"块"的大小，最常见的是1KB，即连2个扇区组成一个block。或4K，连8个扇区组成一个 block。



block 是真正存储数据的地方。

block是 文件系统 中最小的存储单位

扇区 是 磁盘 中最小的存储单位

block调大：

优点： 速度快，节约寻址时间，缺点：空间浪费

比如： 2T硬盘， 前1.5T，使用4K， 把剩下的500G格式化成64K块。用空间换时间

原因：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，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。

显示磁盘状态信息：CentOS6用： dumpe2fs 分区名 ，CentOS7用xfs_info 分区名

```
meta-data=/dev/sda1              isize=512    agcount=4, agsize=12800 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0 spinodes=0
data     =                       bsize=4096   blocks=51200, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal               bsize=4096   blocks=855, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

2、磁盘分区具和挂载

（1）磁盘常识：

工厂生产的硬盘必须经过低级格式化、分区和高级格式化（文中均简称为格式化）三个处理步骤后，电脑才能利用它们存储数据。其中磁盘的低级格式化通常由生产厂家完成，目的是划定磁盘可供使用的扇区和磁道并标记有问题的扇区；而用户则需要使用操作系统所提供的磁盘工具如“fdisk、gdisk”等程序进行硬盘“分区”和“格式化”。

（2）磁盘两种分区表：MBR和GPT

MBR

将分区信息保存到磁盘的第一个扇区(MBR扇区)的64个字节中，每个分区占用16个字节，这16个字节中存有活动状态标志、文件系统标识、起止柱面号、磁头号、扇区号、隐含扇区数目(4个字节)、分区总扇区数目(4个字节)等内容。

特点

MBR分区主分区数目不能超过4个，很多时候，4个主分区并不能满足需要。

MBR分区方案无法支持**超过2TB容量的磁盘**。因为这一方案用4个字节存储分区的总扇区数，最大能表示2的32次方的扇区个数，按每扇区512字节计算，每个分区最大不能超过2TB。磁盘容量超过2TB以后，分区的起止位置也就无法表示了，BIOS将无法识别分区。**（这种分区表逐渐将被淘汰，当然现在很多系统还在用）**

GPT

GUID磁盘分割表（GUID Partition Table）的缩写，含义“全局唯一标识磁盘分区表”，是一个实体硬盘的分区表的结构布局的标准。

GPT的分区方案之所以比MBR更先进，是因为在GPT分区表头中可自定义分区数量的最大值，也就是说GPT分区表的大小不是固定的。在Windows中，微软设定GPT磁盘最大分区数量为128个。

特点

支持2TB以上的大硬盘。

每个磁盘的分区个数几乎没有限制。操作系统存在允许的最多分区数的限制，比如win限制128个

（3）使用fdisk管理分区（MBR分区表）

1 fdisk -l

查看系统所有硬盘及分区

2 fdisk /dev/sdb 进行磁盘分区

fdisk的内部命令：

a toggle a bootable flag

b edit bsd disklabel

c toggle the dos compatibility flag

d delete a partition 删除分区

g create a new empty GPT partition table

G create an IRIX (SGI) partition table

l list known partition types 显示分区类型（linux系统内核，很多个分区的类型，id）

m print this menu 打印帮助菜单

n add a new partition 添加新的分区

o create a new empty DOS partition table

p print the partition table 显示分区表

q quit without saving changes 不保存，退出

s create a new empty Sun disklabel

t change a partition's system id 改变分区类型

u change display/entry units

v verify the partition table

w write table to disk and exit 写分区表信息到硬盘，保存操作并退出

x extra functionality (experts only)

3 分区过程，前面的xfs文件系统磁盘备份与恢复中，已经讲过了，在做一遍，对里边的细节在解释一下

4 格式化 建立文件系统

 # mkfs.xfs /dev/sdb1 #格式化分区mkfs.xfs格式化等xfs文件系统，如果格式化成ext4，那就mkfs.ext4

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：

 # mke2fs [选项] 分区设备文件名 （不建议用）

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

如：# mke2fs -t ext4 -b 2048 /dev/sdb1

默认就很好，底层的东西，能不改就不改！

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

9 /etc/fstab/文件修复：

报错的情况下继续登录，修改/etc/fstab 报错，可能改不了，是挂载的问题，# mount -o remount,rw /，然后修改，最后重启！

（4）使用gdisk管理分区（GPT分区表）

```
# gdisk /dev/sdb
Command (? for help): ?     #  查看帮助
b       back up GPT data to a file
c        change a partition's name
d       delete a partition     #删除分区
i        show detailed information on a partition
l        list known partition types
n       add a new partition          # 添加一个分区
o       create a new empty GUID partition table (GPT)
p       print the partition table          # 打印分区表
q       quit without saving changes      # 退出不保存
r        recovery and transformation options (experts only)
s        sort partitions
```

```
t       change a partition's type code  #修改分区系统id
v       verify disk
w       write table to disk and exit     # 写入分区表并退出
x       extra functionality (experts only)
?       print this menu
```

演示一下：分区过程：n----p-----w-----y

（5）扩展swap分区，查看swap分区的大小可以用free -h命令，专门查看内存的命令

Swap分区在系统的物理内存不够用的时候，把硬盘空间中的一部分空间释放出来，以供当前运行的程序使用。

a、分一个新区，准备给swap

b、mkswap /dev/sdb1（格式化成swap格式） mkfs -t xfs

c、swapon /dev/sdb1（激活/swap，加入到swap分区中，这个是临时生效，要永久生效还是要执行d步骤）

d、vim /etc/fstab（开机自启动新添加的swap分区），在最后追加：

/dev/sdb1 swap swap defaults 0 0

补充：/dev/sdb1的分区系统id从8300改8200：

```
Command (?  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): 8200      改为 swap 的 ID
```

三、磁盘配额、LVM管理和ssm存储管理器使用

1、用户的磁盘配额

（1）概念

Linux系统是多用户任务操作系统，在使用系统时，会出现多用户共同使用一个磁盘的情况，如果其中少数几个用户占用了大量的磁盘空间，势必压缩其他用户的磁盘的空间和使用权限。因此，系统管理员应该适当的开放磁盘的权限给用户，以妥善分配系统资源。

磁盘配额是一种磁盘空间的管理机制，使用磁盘配额可限制用户或组在某个特定文件系统中所能使用的最大空间。

具体的功能：

1）给用户配额和给用户组配额，给组配额基本没什么用

2）能磁盘容量限制和文件个数限制

3）软限制和硬限制

4）宽限时间

如果用户的空间占用数处于软限制和硬限制之间，系统会在用户登陆时警告用户磁盘将满，这个

时间就是宽限时间，默认是 7 天。如果达到了宽限时间，用户的磁盘占用量还超过软限制，那么软限

制就会升级为硬限制。

（2）磁盘配额条件

a、内核必须支持磁盘配额，centos6和7默认就是支持的，不用管他，也可以看一下/boot/config-3.10.0-957.el7.x86_64

b、系统中必须安装了 quota 工具，没找到就yum安装一下

（3）磁盘配额步骤（讲之前随便说一下yum install -y bash-completion加强tab键代码自动补全）

1）分 5GB 的/dev/sdb1 分区，格式化xfs，并将它挂载到/disk 目录当中

2）建立需要做限制的用户

3）在分区上开启磁盘配额功能

ext4系统分区# mount -o remount,usrquota,grpquota /disk 临时生效

xfs系统分区# mount -o uquota,gquota /dev/sdb1 /disk 发现从新挂载不起作用，要挂载的时候-o uquota

我们要想永久生效，则需要修改/etc/fstab 文件，改成：

ext4系统分区 /dev/sdb1 /disk xfs

defaults,usrquota,grpquota 0 0

xfs系统分区 /dev/sdb1 /disk xfs

defaults,uquota,gquota 0 0

注意：永久生效，是要重启才能起作用的

#mount 看一下：

虽然我们写的参数是uquota，但是mount里面显示的却是和ext文件系统相同的参数usrquota

4）建立磁盘配额的配置文件，XFS文件系统，不需要配置文件生成了，不用做这步骤

quotacheck [选项] [分区名]

选项：

-a：扫描/etc/mtab 文件中所有启用磁盘配额功能的分区。如果加入此参数，命令后面 就不需要加入分区名了

-c：不管原有的配置文件，重新扫描并建立新的配置文件

-u：建立用户配额的配置文件，也就是生成 aquota.user 文件

-g：建立组配额的配置文件，会生成 aquota.group 文件

-v：显示扫描过程

-m：强制以读写的方式扫描文件系统，和-M 类似。一般扫描根分区时使用。

-f：强制扫描文件系统，并写入新的配置文件。一般扫描新添加的硬盘分区时使用

quotacheck -auvg，　如果需要给根分区开启配额功能，需要加m：# quotacheck -avum

注意：需要关闭 SELinux，否则会报错

5）设置用户的配额限制

edquota [选项] [用户名或组名]

选项：

-u 用户名：　设定用户配额

-g 组名：　设定组配额

-t：　设定宽限时间 ，一般默认不用改

-p：　复制配额限制。配完一个用户后，我们可以来试试 ： # edquota -p user2 源用户 -u user3目标用户

如果已经设定好某个用户的配额限制，其他用户的配额限制如果和这个用户相同，那么可以直接复制配额限制

```
# edquota -u user1
Filesystem          blocks       soft       hard     inodes     soft
hard
  /dev/sdb1              0      40000      50000          0        8
10
```

注意：xfs文件系统还提供了一个命令来配置限额

xfs_quota -x -c 'limit bsoft=30m bhard=50m isoft=3 ihard=5 user1' /disk

#-x为专家模式，-c为以交换式或参数的形式设置要执行的命令，后面单引号中的命令为，设置user1用户的磁盘使用软限制为30M，硬限制为50M，创建文件数量的软限制为3个，硬限制为5个，对/disk这个目录有效

6）启动和关闭配额（开启后，第一个用户的配置限制设定就OK了），XFS文件系统，配置后自动生效，不用做这步骤

quotaon [选项] [分区名]

选项：

-a：依据/etc/mtab 文件启动所有的配额分区。如果不加-a，后面就一定要指定分区名

-u：启动用户配额

-g：启动组配额

-v：显示启动过程的信息

quotaon -auvg

quotaoff [选项] [分区名]

选项

-a：依据/etc/mtab 文件关闭所有的配额分区。如果不加-a，后面就一定要指定分区名

-u：关闭用户配额

-g：关闭组配额

quotaoff -a

#依据/etc/mtab 文件关闭配额分区

7）磁盘配额查询， **xfs文件系统也通用**

quota [选项] [用户名或组名]

选项：

-u 用户名： 查询用户配额

-g 组名： 查询组配额

-v: 显示详细信息

-s: 以习惯单位显示容量大小，如 M，G

quota -uvs user1

8）repquota 查询文件系统配额， **xfs文件系统也通用,xfs文件系统查询方法：# xfs_quota -x -c report /disk**

repquota [选项] [分区名]

选项：

-a: 依据/etc/mtab 文件查询配额。如果不加-a 选项，就一定要加分区名

-u: 查询用户配额

-g: 查询组配额

-v: 显示详细信息

-s: 以习惯单位显示容量大小

repquota -auvs

（4）测试一下：#先将sdb2目录权限设置成777，因为这是root创建的目录，不这样设置其他用户无法写入文件

$ dd if=/dev/zero of=/disk/testfile bs=1M count=60 #建立 testfile 文件，指定大小 60MB

（5）删除磁盘配额

a、ext4配额删除

1、quotaoff -a 关闭配额功能

2、删除/disk下的配额配置文件

3、umount，卸载磁盘

4、fstab文件删除对应的自动挂载

b、xfs删除配额，不需要quotaoff

```
#方法一：使用xfs_quota命令将对应用户的软硬限制全部设置成0
# xfs_quota -x -c "limit bsoft=0 bhard=0 isoft=0 ihard=0 user1" /disk
#方法二：编辑对应用户的quota配置，将软硬限制全部设置成0
# edquota -u user1
#然后编辑
```

（6）非交互设定用户磁盘配额，主要用于批量用户设置配额的时候，用在shell脚本中：

 # setquota -u 用户名 容量软限制 容量硬限制 个数软限制 个数硬限制 分区名

2、LVM逻辑盘卷管理

（1）简介

LVM是逻辑盘卷管理（Logical Volume Manager）的简称，它是Linux环境下对磁盘分区进行管理的一种机制，LVM是建立在硬盘和分区之上的一个逻辑层，来提高磁盘分区管理的灵活性。

LVM最大的特点就是可以对磁盘进行动态管理。使用了LVM管理分区，动态的调整分区的大小，标准分区是做不到的！

（2）LVM包含的组件

PV（Physical Volume）- 物理卷

物理卷在逻辑卷管理中处于最底层，它可以是实际物理硬盘上的分区，也可以是整个物理硬盘，也可以是raid设备。

VG（Volume Group）- 卷组

卷组建立在物理卷之上，一个卷组中至少要包括一个物理卷，在卷组建立之后可动态添加物理卷到卷组中。

一个逻辑卷管理系统工程中可以只有一个卷组，也可以拥有多个卷组。

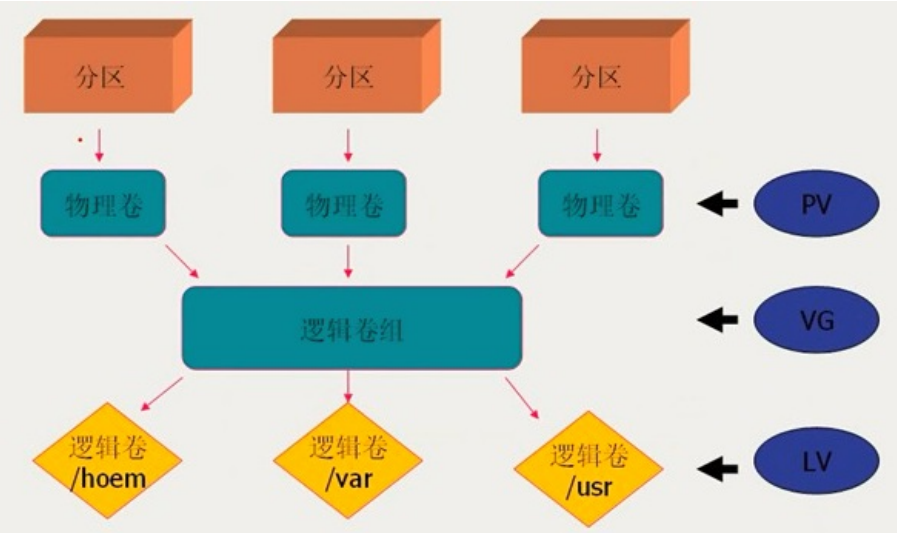
LV（Logical Volume）- 逻辑卷

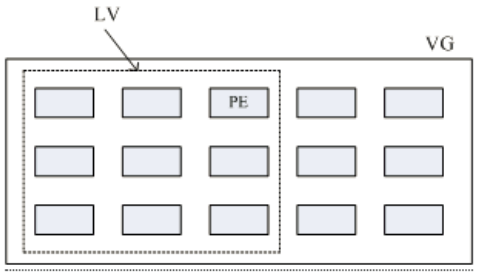
逻辑卷建立在卷组之上，卷组中的未分配空间可以用于建立新的逻辑卷，逻辑卷建立后可以动态地扩展和缩小空间。

系统中的多个逻辑卷可以属于同一个卷组，也可以属于不同的多个卷组。

PE（Physical Extent）- 物理块

PE是整个LVM 最小的储存区块，默认每个PE区块是4MB大小，也就是说，其实我们的数据都是由写入PE 来处理的。简单的说，这个PE 就有点像文件系统里面的block 大小。





（3）建立 LVM 的步骤：

a、首先要有新的物理盘或者分区。

b、 然后把物理分区/或整个磁盘建立成为物理卷（PV）

c、 接下来把物理卷整合成为卷组（VG）。卷组就已经可以动态的调整大小了，可以把物理卷PV加入卷组，实现扩容

d、 最后就是把卷组再划分成为逻辑卷（LV），当然逻辑卷也是可以直接调整大小的。

对于上层应用或者用户来说逻辑卷就是分区，所以也需要格式化和挂载。

（4）使用LVM的模式进行磁盘的分区，以及LVM的管理

a、安装的时候，图形化LVM分区演示

b、命令的方式演示LVM的扩容

磁盘准备

这里我们演示分区和整块硬盘来做位PV，要三个硬盘，一个硬盘分2个区，先有一个硬盘和一个分区组成一个PV，2PV组成VG，剩下的扩容

安装LVM管理工具

检查系统中是否安装了LVM管理工具

```
# rpm -qa | grep lvm
```

如果未安装，则使用yum 方式安装

```
# yum install lvm*
```

新建分区

gdisk分区，前面讲过了

创建PV

pvcreate [设备文件名]

建立物理卷时，我们说即可以把整块硬盘都建立成物理卷，也可以把某个分区建立成物理卷。

查看PV

pvdisplay

还可以使用命令pvscan 查看简略信息。

pvscan

创建VG

vgcreate [选项] 卷组名 物理卷名　　#可以将多个PV组成一个VG，物理卷名之间用空格分隔

选项：

-s PE 大小：指定 PE 的大小，单位可以是 MB,GB,TB 等。如果不写默认 PE 大小事4MB，一般不改

查看VG

vgdisplay

说明：

VG Name　　VG的名称

VG Size　　VG的总大小

PE Size　　PE的大小，默认为4MB

Total PE　　PE的总数量，5114 x 4MB = 19.98GB

Free PE / Size　　剩余空间大小

同样可以使用命令vgs 查看简要信息。

vgs

创建LV

lvcreate [选项] [-n 逻辑卷名] 卷组名

选项：

-L 容量：指定逻辑卷大小，单位 MB，GB，TB 等

-l 个数：按照 PE 个数指定逻辑卷大小，这个参数需要换算容量，太麻烦，一般不用

-n 逻辑卷名：指定逻辑卷名

lvcreate -L 5G -n lv1 vg1

查看LV的信息

lvdisplay

说明：

LV Path　　LV的路径，全名

LV Name　　LV的名字

VG Name　　所属的VG

LV Size　　LV的大小

再来看LV 的简要信息

#lvs

格式化LV

mkfs.xfs /dev/vg0/lv1

挂载使用

mkdir /mnt/lv1

mount /dev/vg0/lv1 /mnt/lv1/

df -Th

将挂载信息写入/etc/fstab

添加测试数据

对LVM进行扩容操作，所以向/mnt/lv1 中写入测试数据以验证LVM 的磁盘动态管理。扩容不会影响原理的数据！

touch /mnt/lv1/test_lvm_dynamic.disk

touch /mnt/lv1/test_lvm_dynamic.disk2

touch /mnt/lv1/test_lvm_dynamic.disk3

LVM的扩容操作

LVM最大的好处就是可以对磁盘进行动态管理，而且不会丢失现有的数据。

假如有一天，lv1的使用量达到了80%，需要扩容，那我们该怎么做呢？

VG的先扩容

增加PV，然后讲新的PV添加到VG中

vgextend vg1 /dev/sdb7

vg1变大后就有了很多剩余空间，所以我们可以从vg0中再分配点空间给lv1。

LV的扩容

查看vg1 的剩余容量，决定好扩容多少

对lv1进行扩容。

lvextend -L +1G /dev/vg1/lv1　　有+：原有的基础质上增加xxG

lvextend -L 30G /dev/vg1/lv1　　没有+：容量改变到xxxG

说明：在lv1原有的基础上增加了1G.

使用df -Th 命令查看实际的磁盘容量。

发现实际容量并没有变化，因为我们的系统还不认识刚刚添加进来的磁盘的文件系统，所以还需要对文件系统进行扩容。

 # resize2fs /dev/vg1/lv1 注意：resize2fs是属于ext4文件系统，xfs文件系统相同功能的命令改成了：**xfs_growfs**

查看测试数据

数据正常，对lv1的在线动态扩容完成。

注意：我们一般情况下，都是对磁盘进行扩容哈，缩减没的什么实际意义，这里就不演示一下！

另外：xfs文件系统只支持增大分区空间的情况，不支持减小的情况（切记！！！！！）

ext4文件系统LVM的缩减操作

LV的缩减

A. umount 文件系统

B. 缩减文件系统

```
# resize2fs /dev/vg1/lv1 4G
```

提示需要先运行磁盘检查。

C. 检查磁盘

```
# e2fsck -f /dev/vg1/lv1
```

D. 再次执行缩减操作

缩减文件系统成功，下面缩减LV的大小。

```
# resize2fs /dev/vg1/lv1 4G
```

E. 缩减LV

```
# lvreduce /dev/vg1/lv1 -L 4G
```

说明：Step E 和Step D 缩减的大小必须保持一致，这里的4G是缩减到的大小；如果使用的是"-4G"，则表示容量减少多少的意思。

F. 挂载查看

LV 缩减成功。

VG的缩减

A. umount 文件系统

B. 查看当前的PV详情

```
# pvdisplay
```

C. 将/dev/sdg 从vg1 中移除

```
# vgreduce vg1 /dev/sdg
```

D. 再次查看PV情况

/dev/sdg 已经不属于vg1了。

E. 查看vg1 的情况

vg1 的大小减少了5GB.

删除LVM，ext和xfs文件系统通用

如果要彻底的来移除LVM的话，需要把创建的步骤反过来操作。

umount 文件系统

移除LV

lvremove /dev/vg1/lv1

移除VG

vgremove vg1

移除PV

pvremove /dev/md5 /dev/sdf1 /dev/sdg /dev/sdh

LVM 移除成功。

3、system-storage-manager检查SSM

安装SSM

yum -y install system-storage-manager

查看磁盘信息

ssm list dev

实验背景：

公司要搭建一台邮件服务器，需要你创建一个名为mail 的LVM存储池，并在其上创建一个名为mail-lv，

初始大小为1G的lvm卷，格式化为xfs文件系统，并将其挂载/mail-lv目录下。

创建目录

mkdir /mail-lv

用的一条命令，可以把我们前面学的LVM管理，自动完成：

ssm create -s lv大小 -n lv名称 --fstype lv文件系统类型 -p 卷组名 设备 挂载点

自动把设备变成pv，创建vg ， lv ，格式化文件系统， 自动挂载

ssm create -s 1G -n mail-lv --fstype xfs -p mail /dev/sdb[3-4] /mail-lv

四、恢复ext4文件系统下误删除的文件，注意这里讲的工具只能在CentOS6.x上使用，在CentOS7.x上无效

工具extundelete介绍：（http://extundelete.sourceforge.net/）

1.extundelete的文件恢复工具，该工具最给力的一点就是支持ext3/ext4双格式分区恢复。

2. **在实际线上恢复过程中，切勿将extundelete安装到你误删的文件所在硬盘，这样会有一定几率将需要恢复的数据彻底覆盖。**

3. extundelete还是有很大的不完整性,基于整个磁盘的恢复功能较为强大,基于目录和文件的恢复还不够强大。

4. extundelete执行完毕后在当前目录生产一个RECOVERED_FILES目录，里面即是恢复出来的文件，还包括文件夹。

5.任何的文件恢复工具，在使用前，均要将要恢复的分区卸载或挂载为只读，防止数据被覆盖使用。

```
umount /dev/partition
mount -o remount,ro /dev/partition
```

6.保持良好的习惯，绝对比恢复数据要更简单。

extundelete安装

0.yum install -y e2fsprogs* e2fslibs*

1.上传extundelete工具源码包

2.tar -axf extundelete-0.2.4.tar.bz2 -C /usr/local/src

3.cd /usr/local/src/extundelete-0.2.4

4../configure --prefix=/usr/local/extundelete

5.make

6.make install

7.ln -s /usr/local/extundelete/bin/* /usr/local/bin/

extundelete恢复文件的步骤：

1. umount或者read only 分区

```
umount /dev/partition
mount -o remount,ro /dev/partition
```

卸载的原因：文件删除后，仅仅是将文件的inode节点中的扇区指针清零，实际文件还储存在磁盘上，如果磁盘继续以读写模式挂载，这些已删除的文件的数据块就可能被操作系统重新分配出去，在这些数据库被新的数据覆盖后，这些数据就真的丢失了，恢复工具也无力回天。

2.切换到存储恢复文件的目录

```
cd $dir
```

3.命令用法

语法：extundelete [--options] device-file

选项：

a、# extundelete /dev/sda1 --restore-all #恢复某分区里所有被删除的数据，文件名还是原来的

b、 #查看已经删除的文件的inode号

```
# extundelete --inode 2 /dev/sda1
```

然后恢复指定的inode的文件，保存在当前目录下的RECOVERED_FILES里，文件名为【file.$inode】

```
#extundelete /dev/sda1 --restore-inode 13,14
```

c、--restore-file 'filename' #恢复指定的文件（被删除的），文件位于当前目录下的RECOVERED_FILES/$filename

```
# extundelete /dev/sda1 --restore-file cc.txt
```

注意：如果在根下删除文件了，想恢复，怎么办？

把extundelete安装到另一台系统版本一样的服务器上，安装号的文件目录拷贝到U盘，

把U盘插入需要恢复的服务器，恢复时，恢复的文件要保存到U盘中

五、ext4文件系统的备份与还原，前第11章节，已经讲了xfs文件系统的备份和还原

ext4文件系统常用的备份与恢复的命令dump和restore命令

1、安装和基本了解dump

```
# yum -y install dump
```

在正式介绍 dump 命令之前，我们需要知道 dump 命令可以支持 0～9 共 10 个备份级别。其中，0 级别指的就是完全备份，1～9 级别都是增量备份级别。

也就是说，当我们备份一份数据时，第一次备份应该使用 0 级别，会把所有数据完全备份一次；第二次备份就可以使用 1 级别了，它会和 0 级别进行比较，把 0 级别备份之后变化的数据进行备份；第三次备份使用 2 级别，2 级别会和 1 级别进行比较，把 1 级别备份之后变化的数据进行备份；以此类推。

需要注意的是，只有在备份整个分区或整块硬盘时，才能支持 1～9 的增量备份级别；如果只是备份某个文件或不是分区的目录，则只能使用 0 级别进行完全备份。

dump 命令格式如下：

```
# dump [选项]  备份之后的文件名  原文件或目录
```

选项：

- 0～9：就是我们说的 0～9 共 10 个备份级别；

-f 文件名：指定备份之后的文件名；

-u：备份成功之后，把备份时间记录在 /etc/dumpdates 文件中；
-v：显示备份过程中更多的输出信息；
-j：调用 bzlib 库压缩备份文件，其实就是把备份文件压缩为 .bz2 格式，默认压缩等级是 2；
-W：查询/etc/dumpdates 文件的记录信息；

dump 命令是一个较为复杂的命令，如果我们只是想要实现数据的备份与恢复，那么掌握以上几个选项就足够了。

2、备份分区

我们先来看看如何使用 0 级别备份分区。为了安全，我们新建一个分区来做这个实验。

```
# dump -0uj -f /root/disk1.bak.bz2  /disk1    #备份命令。先执行一次完全备份,并压缩和更新备份时间
```

如果 /disk1分区的内容发生了变化，则可以使用 1 级别进行增量备份。当然，如果数据会继续发生变化，则可以继续使用 2～9 级别增量备份。命令如下：

```
# touch aaa.txt  #新建一些文件
# dump -1uj -f /root/disk1.bak1.bz2 /disk1/    #增量备份/boot分区，并压缩
```

如果备份的是整个分区，那么是可以使用“dump -W”命令来查询分区的备份时间及备份级别的。不过要注意，如果备份时没有使用“-u”选项，那么“dump -W”命令是不会记录备份的时间和级别的。命令如下：

```
# dump -W
```

3、备份文件或目录

dump 命令也可以文件或目录，不过，只要不是备份分区，就只能使用 0 级别进行完全备份，而不再支持增量备份。同时，不能使用“-u”选项更新分区的备份时间，当然也不能使用“dump -W”命令查询到文件或目录的备份。

我们说 /etc/ 目录是重要的配置文件目录，那么我们就备份这个目录来看看吧。命令如下：

```
# dump -0j -f /root/etc.dump.bz2 /etc/     #完全备份/etc/目录
```

4、restore命令格式

restore 模式选项 选项

模式选项：restore命令常用的模式有以下四种，这四种模式不能混用

-C ：比较备份数据和实际数据的变化

-i ：进入交互模式，手工选择需要恢复的文件。

-t ：查看模式，用于查看备份文件中拥有哪些数据。

-r ：还原模式，用于还原数据。

选项：

-f ： 指定备份文件的文件名

比较备份数据和实际数据的变化示例

```
rf -rf /disk1/aaa.txt     #把/disk1目录中的一个文件删除，造成丢失的假象
restore -C -f /boot/disk1.bak.bz2    #发现文件丢失
```

restore 还原模式示例

```
#还原disk1.bak.bz2分区备份
#先还原完全备份的数据
mkdir disk1.test
cd disk1.test
restore -r -f /root/disk1.bak.bz2     #解压缩
restore -r -f /root/disk1.bak1.bz2  #恢复增量备份的数据
```

六、RAID磁盘阵列的原理与搭建

1、RAID概念

RAID （ Redundant Array of Independent Disks ）即独立磁盘冗余阵列，通常简称为磁盘阵列。简单地说， RAID 是由多个独立的磁盘驱动器组成的磁盘子系统，从而提供比单个磁盘更高的存储性能和数据冗余的技术。在服务器整个系统中， RAID 被看作是由两个或更多磁盘组成的存储空间，通过并发地在多个磁盘上读写数据来提高存储系统的 I/O 性能。

2、常见的RAID类型

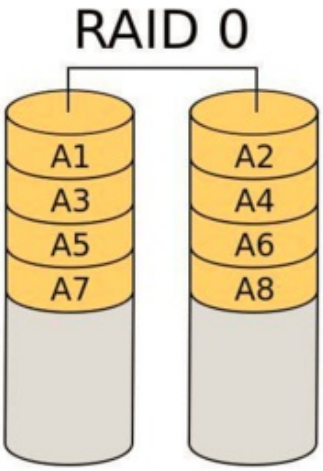
1）RAID 0

0 :(stripe)条带模式，至少需要两块硬盘，每一份数据平均分成多份存储在多个磁盘中，且都处于一个水平条带上

优点:读写速度提高，用了多少块盘就是多少倍

缺点:无冗余能力(也称容错能力)

空间利用率:多个磁盘的总和，100%

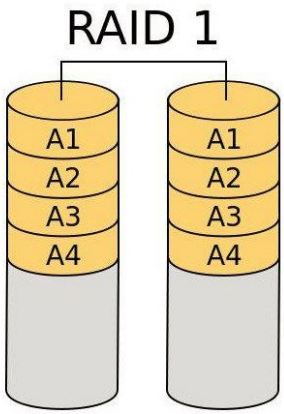


2）RAID 1

1:(mirror)镜像模式，磁盘数量需要2的倍数，两个磁盘中存储的数据完全一致，当一个盘损坏时，数据依然可以进行读写

优点:有冗余能力

缺点:磁盘利用率 只有50%，写的速度下降

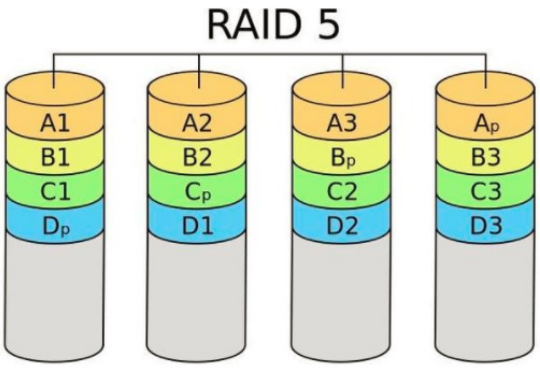


3）RAID 5

5:check code校验码模式，至少需要三块硬盘，每一次存储数据时，采用条带模式存储到n-1数量的磁盘中，另外一个磁盘存放的是其他几个磁盘中的数据以某种加密方式之后得出的加密数据，且每一次存储，存储加密数据的磁盘都是不断变化的，当其中任何一个盘的数据损坏时，都可以通过加密方式和其余两块磁盘的数据来得知另外一个盘的数据，具有较高的冗余能力

优点:读写速度快，有冗余能力

缺点:磁盘利用率为n-1/n，两个磁盘损坏时，数据将丢失

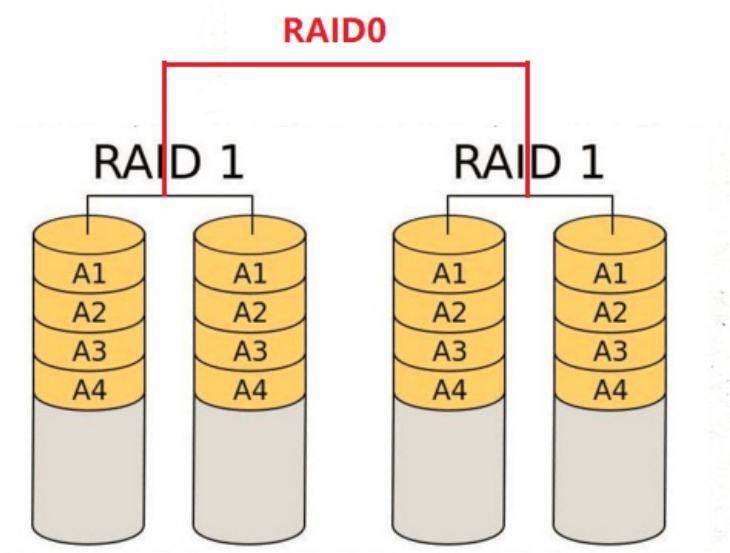


4） 组合起来用RAID10

10:mirror+stripe模式，至少需要4快硬盘，先将两块硬盘组成Raid1，然后将两组Raid1组合成Raid0，存储一分数据时，每一Raid1的组合存储的数据都是均分的，然后Raid1组合再用mirror模式存储

优点:读写速度快，冗余能力强

缺点:磁盘利用率为50%



3、RAID-0-1-5-10搭建及使用

1）RAID的实现方式

硬RAID：需要RAID卡，我们的磁盘是接在RAID卡的，由它统一管理和控制。

数据也由它来进行分配和维护；它有自己的cpu，处理速度快，实际工作中就是用这个的。

而且不同厂商的RAID卡的配置是不一样的，要看说明书来配置，插入硬盘raid卡，配置raid卡，

然后在raid磁盘阵列上安装Linux操作系统，课上没法讲！　RAID和LVM整合

2）软RAID：通过操作系统中的软件来实现，工作场景中基本不用，可以给大家演示，理解RAID原理还是可以的！

使用的命令mdadm：

常用的 [options]:

-A,--assemble：激活一个以前定义的阵列

-C,--create：创建一个新的阵列

-Q,--query：查看一个device，判断它为一个 mddevice 或是 一个 md 阵列的一部分

-D,--detail：打印一个或多个 md device 的详细信息

-E,--examine：打印 device 上的 mdsuperblock 的内容

-G,--grow：改变在用阵列的大小，增加或减少组成阵列的磁盘个数

-h, --help：帮助信息，用在以上选项后，则显示该选项信息

-v,--verbose：显示细节

-r：移除设备

-f： 将设备状态定为故障

-c：设定阵列的块chunk块大小 ，单位为KB

-s,--scan：扫描配置文件或 /proc/mdstat以搜寻丢失的信息。配置文件/etc/mdadm.conf

-l 或--level ：设定磁盘阵列的级别，0 1 5 10 50

-n或--raid-devices ：指定阵列成员（分区/磁盘）的数量

-x或--spare-devicds：指定阵列中备用盘的数量

实例演示一下软RAID的配置：

1、准备硬盘，在虚拟机上添加14块硬盘

2、创建raid0，用第2和3两块硬盘，sdb，sdc：

创建RAID0的具体步骤：

创建：

```
# mdadm  -C -v /dev/md0 -l 0 -n 2  /dev/sdb  /dev/sdc
```

查看阵列信息

```
#   mdadm -Ds
# mdadm -D /dev/md0
```

生成配置文件

```
# mdadm -Ds > /etc/mdadm.conf
```

对创建的RAID0进行文件系统创建并挂载

```
# mkfs.xfs /dev/md0
# mkdir /raid0
```

```
# mount /dev/md0 /raid0/
#  df -Th /raid0/
```

开机自动挂载

```
#  blkid /dev/md0    blkid是查询uuid号的命令！
# echo "UUID=5bba0862-c4a2-44ad-a78f-367f387ad001 /raid0 xfs
defaults 0 0" >> /etc/fstab
```

3、创建raid1，用第4和5两块硬盘，用第6块盘热备份盘，raid1中的一块盘坏了，备份盘会自动顶替故障盘

具体步骤：

创建

```
#  mdadm -C -v /dev/md1 -l 1 -n 2 -x 1 /dev/sd[d,e,f]
```

将RADI信息保存到配置文件

```
# mdadm  -Dsv >  /etc/mdadm.conf
```

查看 RAID 阵列信息：

```
# mdadm  -D /dev/md1
```

在RAID设备上创建文件系统

```
# mkfs.xfs /dev/md1
# mkdir /raid1
# mount /dev/md1  /raid1/
```

准备测试文件

```
# cp /etc/passwd /raid1/
```

模拟损坏

下面模拟RAID1中数据盘/dev/sde出现故障，观察/dev/sdf备用盘能否自动顶替故障盘

```
# mdadm  /dev/md1 -f  /dev/sde
```

查看一下阵列状态信息

```
# mdadm  -D /dev/md1
```

查看数据是否丢失

```
# ls /raid1/    #数据正常，没有丢失
```

移除损坏的设备：

```
# mdadm  -r /dev/md1 /dev/sde
```

查看信息：

```
# mdadm  -D /dev/md1
```

更新配置文件

```
# mdadm  -Dsv > /etc/mdadm.conf
```

4、创建RAID5，用第7、8、9三块硬盘做，用第10块硬盘做热备份盘

具体步骤：

创建RAID-5

```
#  mdadm -C -v /dev/md5 -l 5 -n 3 -x 1  /dev/sd{g,h,i,j}
# mdadm -D /dev/md5
```

停止MD5阵列

```
# mdadm -Dsv > /etc/mdadm.conf    #停止前，一定要先保存配置文件
# mdadm  -D /dev/md5  ##停止前，请确认数据已经同步完
# mdadm  -S /dev/md5
```

激活MD5阵列

```
# mdadm  -As
```

扩展RAID5磁盘阵列，将热备盘增加到md5中，使用md5中可以使用的磁盘数量为4块

```
# mdadm -G /dev/md5 -n 4    #-G或--grow    改变阵列大小或形态
# mdadm -Dsv > /etc/mdadm.conf    #保存配置文件
```

备注：阵列只有在正常状态下，才能扩容，降级及重构时不允许扩容。对于raid5来说，只能增加成员盘，不能减少。而对于raid1来说，可以增加成员盘，也可以减少。

```
# mdadm -D /dev/md5  #查看状态
```

5、创建RAID10，用11，12，13，14块盘

具体步骤：

```
# mdadm -C -v /dev/md10 -l 10 -n 4 /dev/sd[k,l,m,n]    #一条命令就可以ok了
```

6、删除RAID所有信息及注意事项

```
# umount /dev/md0 /raid0    #如果你已经挂载raid，就先卸载。
# mdadm  -Ss                  #停止raid设备
# rm -rf /etc/mdadm.conf      #删除raid配置文件
#  mdadm  --zero-superblock /dev/sdb  #清除物理磁盘中的raid标识
#  mdadm  --zero-superblock /dev/sdc  #清除物理磁盘中的raid标识
......
参数：--zero-superblock :    #擦除设备中的MD超级块
```