

“十三五”普通高等教育规划教材

Linux 基础及应用教程 (基于 CentOS 7)

第②版

梁如军 王宇昕 车亚军 等编著



提供电子教案

<http://www.cmpedu.com>



机械工业出版社
CHINA MACHINE PRESS

第4章 本地存储管理

主讲人：梁如军

2015-05-05

本章内容要点

- 磁盘相关概念
- 磁盘分区工具
- 创建和维护LVM 系统
- Linux 文件系统的挂装和卸装
- ext2/ext3/ext4 、 xfs文件系统管理
- 磁盘限额

本章学习目标

- 熟悉硬盘的技术指标以及如何挑选服务器硬盘
- 学会使用磁盘分区工具fdisk/parted
- 理解LVM的相关概念
- 理解静态分区的缺点以及LVM的优势
- 学会扩展和缩减逻辑卷
- 理解Linux 文件系统的概念
- 学会挂装和卸装文件系统
- 学会使用各种移动存储介质
- 掌握ext3/ext4 文件系统的创建、管理、维护

硬盘及其相关概念

- **主轴转速**：指硬盘盘片在一分钟内所能完成的最大转数。
- **平均寻道时间**：指磁头从得到指令到寻找到数据所在磁道的时间，它描述硬盘读取数据的能力。
- **数据传输率**：指的是从硬盘缓存向外输出数据的速度，单位为MB/s。
- **高速缓存**：缓存是数据的临时寄存器，主要用来缓解速度差和实现数据预存取等。
- **单碟容量**：指每张碟片的最大容量。这是反映硬盘综合性能指标的一个重要的因素。

硬盘接口方式

- **FC-AL**接口主要应用于任务级的关键数据的大容量实时存储。可以满足高性能、高可靠和高扩展性的存储需要。
- **SCSI**接口主要应用于商业级的关键数据的大容量存储。
- **SAS**接口是个全才，可以支持**SAS**和**SATA**磁盘，很方便地满足不同性价比的存储需求，是具有高性能、高可靠和高扩展性的解决方案，因而被业界公认为取代并行**SCSI**的不二之选。
- **SATA**接口主要应用于非关键数据的大容量存储，近线存储和非关键性应用（如替代以前使用磁带的备份）。
- **PATA（俗称IDE）**接口已基本淘汰。

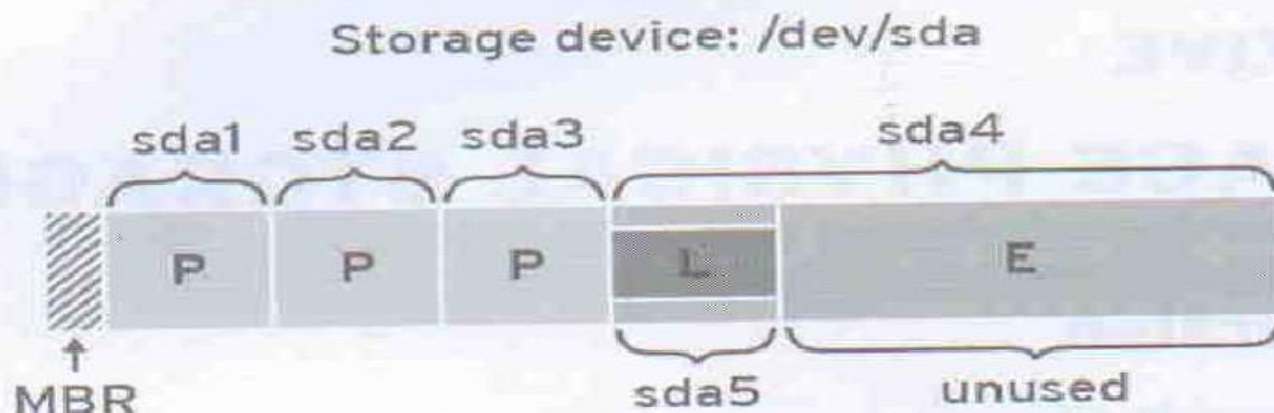
（ Main Boot Record, MBR ）

- **MBR位于硬盘的0磁道0柱面1扇区【512字节】**
 - 装载操作系统的硬盘引导程序【446字节】
 - 硬盘分区表 (Disk Partition Table, DPT)【64字节】
 - 分区ID或者类型
 - 分区起始磁道
 - 分区磁道数
 - 最后两个字节“55, AA”是分区的结束标志
- **MBR是由分区程序（如fdisk）所产生的**
 - 不依赖任何操作系统
 - 硬盘引导程序是可以改变的，从而实现多系统共存。

- 指向附加分区描述符的扩展分区
- 内核最多支持分区数：
 - IDE驱动器为63
 - SCSI驱动器为15
- 为什么是分区驱动？
 - 容量、性能、配额和修复

分区类型

Describe MBR, Primary, Extended, and Logical Partitions



P = Primary partition
E = Extended partition
L = Logical partition

两种硬盘存储方式

■ 基本硬盘存储

- 在基本磁盘上存储数据需要在磁盘上创建主分区、扩展分区和逻辑分区，然后对这些分区进行管理。

■ 动态硬盘存储

- 在动态磁盘上存储数据需要在磁盘上创建动态卷，然后对这些卷进行管理。

磁盘分区工具

- 常用的分区工具
 - fdisk
 - sfdisk
 - GNU parted—高级分区操作（创建、复制、调整大小等等）
- partprobe—重新设置内存中的内核分区表版本

磁盘分区工具——fdisk

- 进入fdisk的交互操作方式
fdisk <硬盘设备名>
- 在命令行方式下显示指定硬盘的分区表信息
fdisk -l <硬盘设备名>

子命令	说明	子命令	说明
a	调整硬盘的启动分区	p	列出硬盘分区表
d	删除一个硬盘分区	q	退出fdisk，不保存更改
l	列出所有支持的分区类型	t	更改分区类型
m	列出所有命令	u	切换所显示的分区大小的单位
n	创建一个新的分区	w	把设置写入硬盘分区表之后退出

磁盘分区工具——parted

■ 交互模式

parted [选项] <硬盘设备名>

■ 命令行模式

parted [选项] <硬盘设备名> <子命令> [<子命令参数>]

□ 子命令

- 打印帮助信息: help [COMMAND]
- 显示分区表: print [free|NUMBER|all]
- 创建新分区: mkpart PART-TYPE [FS-TYPE] START END
- 删除指定分区: rm NUMBER
- 设置分区标记: set NUMBER FLAG STATE

静态分区的缺点

- 在安装 Linux 的过程中如何正确地评估各分区大小是一个难题，因为系统管理员不但要考虑到当前某个分区需要的容量，还要预见该分区以后可能需要的容量的最大值。
- 某个分区空间耗尽时，通常的解决方法是：
 - 使用符号链接 —— 破坏了 Linux 文件系统的标准结构
 - 使用调整分区大小的工具 (如: **Patition Magic** 等) —— 必须停机一段时间进行调整
 - 备份整个系统、清除硬盘、重新对硬盘分区，然后恢复数据到新分区 —— 必须停机一段时间进行恢复操作

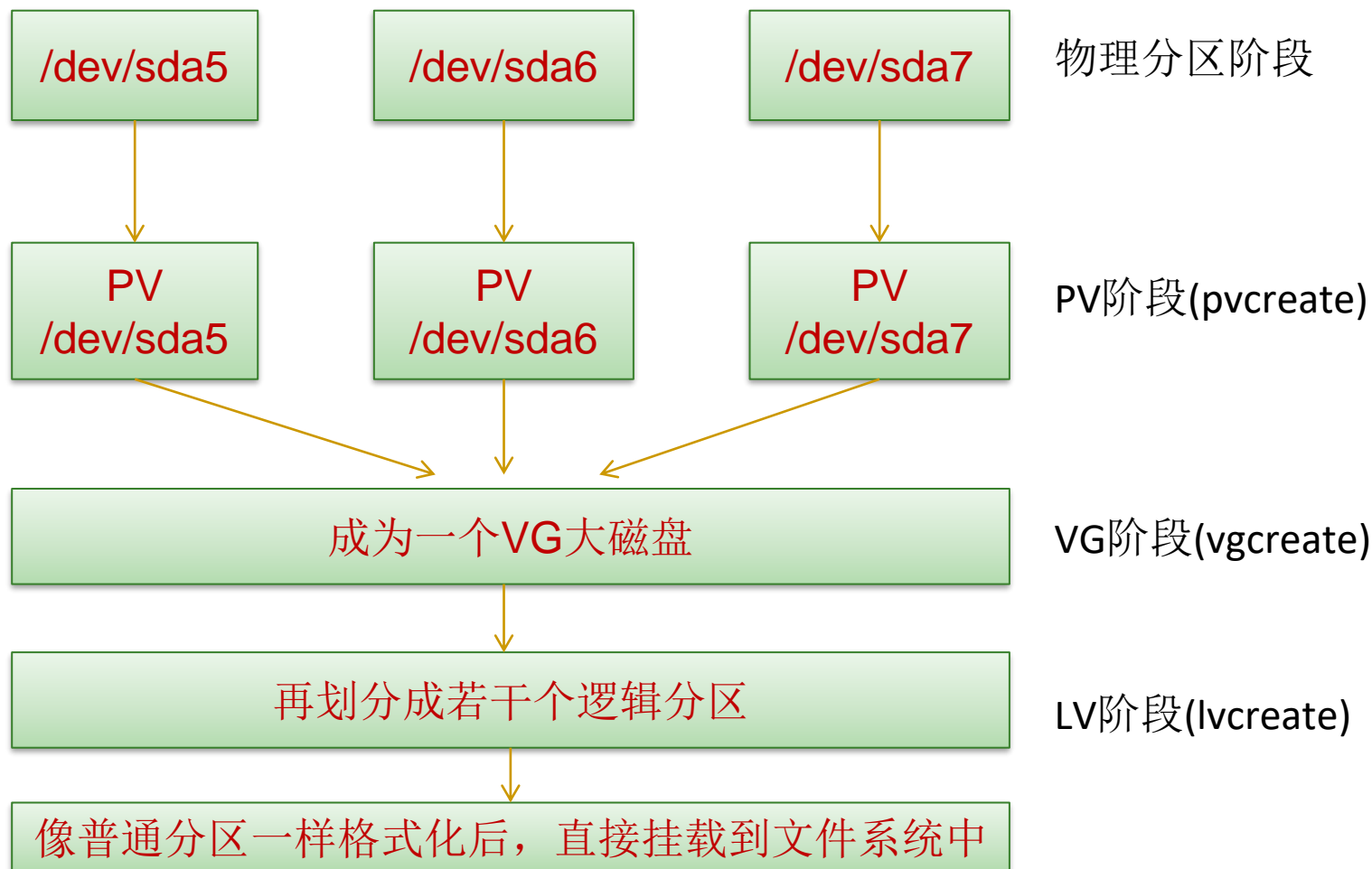
LVM的相关概念

什么是逻辑卷管理器

- LVM 是逻辑盘卷管理（Logical Volume Manager）的简称，它是 Linux 环境下对卷进行方便操作的抽象层。
- LVM 是建立在硬盘和分区之上的一个逻辑层，来为文件系统屏蔽下层磁盘分区布局，从而提高磁盘分区管理的灵活性。
- LVM 允许在多个物理设备间重新组织文件系统，包括重新设定文件系统的大小。

- 通过 LVM 可以轻松管理磁盘分区，如：将若干个磁盘分区连接为一个整块的卷组（**volume group**），形成一个存储池。
- 可以在卷组中随意创建逻辑卷（**logical volumes**），并进一步在逻辑卷上创建文件系统。
- 通过 LVM 可以方便的调整存储卷组的大小，并且可以对磁盘存储按照组的方式进行命名、管理和分配。

LVM 结构图



- 物理卷(physical volume, PV)在 LVM 系统中处于最底层
- 物理卷可以是整个硬盘、硬盘上的分区或从逻辑上与磁盘分区具有同样功能的设备（如：RAID）
- 物理卷是 LVM 的基本存储逻辑块，但和基本的物理存储介质（如分区、磁盘等）比较，却包含有与 LVM 相关的管理参数

- 每一个物理卷被划分为基本单元（称为 **Physical Extent, PE**），具有唯一编号的 **PE** 是可以被 **LVM** 寻址的最小存储单元
- **PE** 的大小可根据实际情况在创建物理卷时指定，默认为 **4MB**
- **PE** 的大小一旦确定将不能改变，同一个卷组中的所有物理卷的 **PE** 的大小需要一致

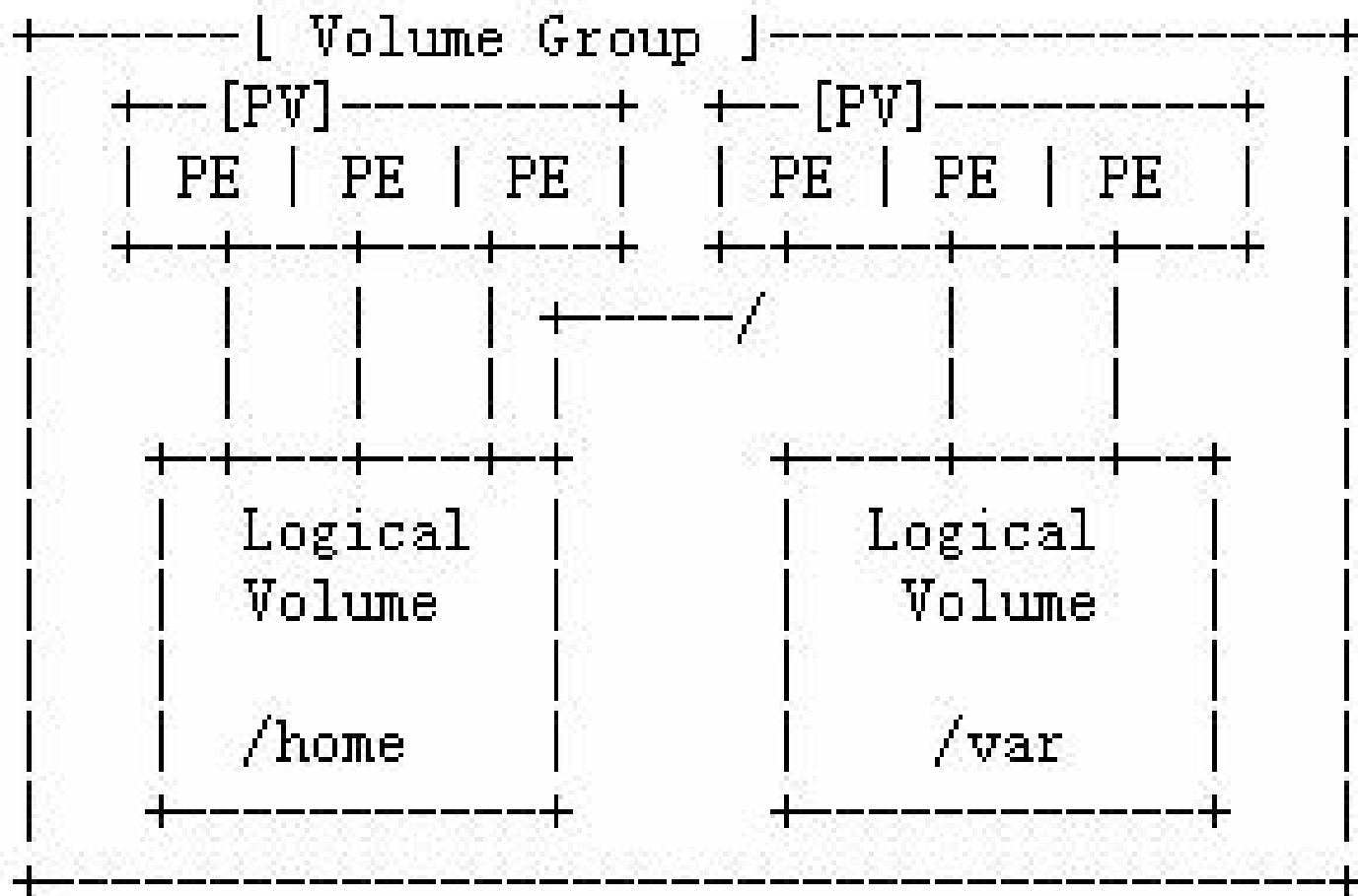
- 卷组(Volume Group, VG)建立在物理卷之上，它由一个或多个物理卷组成
- 卷组创建之后，可以动态添加物理卷到卷组中，在卷组上可以创建一个或多个“LVM 分区”（逻辑卷）
- 一个 LVM 系统中可以只有一个卷组，也可以包含多个卷组
- LVM 的卷组类似于非LVM系统中的物理硬盘

- 逻辑卷(Logical Volume, LV)建立在卷组之上，它是从卷组中“切出”的一块空间
- 逻辑卷创建之后，其大小可以伸缩
- LVM 的逻辑卷类似于非 LVM 系统中的硬盘分区，在逻辑卷之上可以建立文件系统（比如 /home 或者 /usr 等）

- 逻辑卷也被划分为可被寻址的基本单位（称为 Logical Extent, LE）
- 在同一个卷组中，LE 的大小和 PE 是相同的，并且一一对应

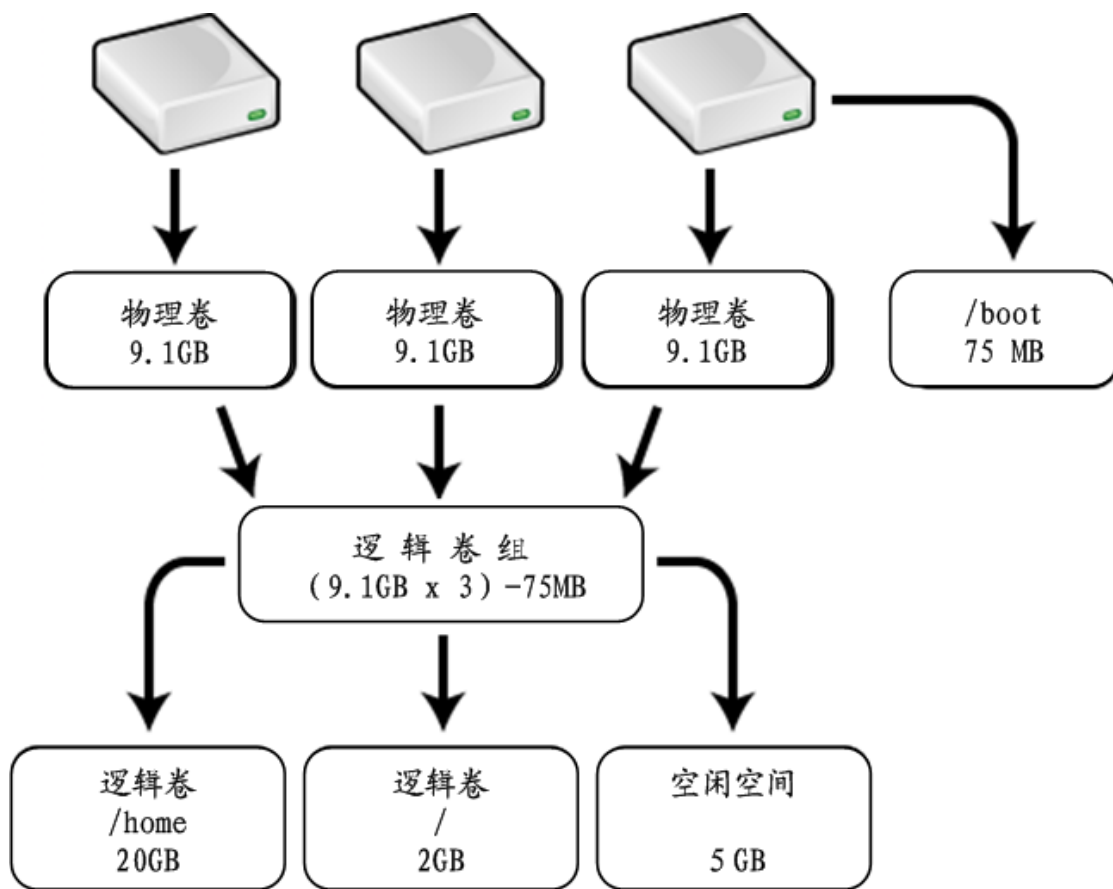
- 和非 LVM 系统将包含分区信息的元数据保存在位于分区的起始位置的分区表中一样，逻辑卷以及卷组相关的元数据也是保存在位于物理卷起始处的卷组描述符区域 (Volume Group Descriptor Area, VGDA) 中。
- VGDA 包括以下内容： PV描述符、VG描述符、LV描述符、和一些PE描述符。

PV-VG-LV-PE 关系图



LVM 与文件系统之间的关系

- `/boot` 分区不能位于卷组中，因为引导装载程序无法从逻辑卷中读取。
- 如果你想把 `/` 分区放在逻辑卷上，必须创建一个与卷组分离的 `/boot` 分区。



PV-VG-LV 的设备名

	含义	设备名
PV	物理卷：磁盘或分区	/dev/sda?
VG	卷组：一组磁盘 和/或 分区	/dev/<VG name>/ (目录)
LV	逻辑卷：LVM 分区	/dev/<VG name>/<LV name>

使用 LVM 系统的步骤

- 创建 LVM 类型的分区
- 在新硬盘上创建物理卷
- 将新创建的物理卷添加到卷组
- 在卷组中创建逻辑卷
- 在逻辑卷中创建文件系统
- 挂装创建的文件系统

LVM管理工具的使用

创建 LVM 的相关命令

■ 创建物理卷

pvcreate <磁盘或分区设备名>

■ 创建卷组

vgcreate <卷组名> <物理卷设备名> [...]

■ 创建逻辑卷

lvcreate <-L 逻辑卷大小> <-n 逻辑卷名> <卷组名>

lvcreate <-l PE值> <-n 逻辑卷名> <卷组名>

PE值可以通过使用命令 `vgdisplay|grep "Free PE"` 获得。

创建 LVM 的命令举例

- 创建两个物理卷

```
# pvcreate /dev/sdb{1,5}
```

- 用已创建的2个物理卷创建名为wwwVG的卷组

```
# vgcreate wwwVG /dev/sdb1 /dev/sdb5
```

- 在wwwVG卷组中创建大小为2GB，名字为www的逻辑卷

```
# lvcreate -L 2G -n www wwwVG
```


查看卷信息

- 查看物理卷

- # pvdiskdisplay [<物理卷设备名>]

- 查看卷组

- # vgdiskdisplay [<卷组名>]

- 查看逻辑卷

- # lvdiskdisplay [<逻辑卷设备名>]

- 若卷组中无剩余空间，首先扩展卷组
 - 添加硬盘，在磁盘上创建 **8e** 类型的分区
 - 在分区上创建物理卷
 - 将物理卷添加到卷组中

```
# vgextend <卷组名> <物理卷设备名> [...]
```
- 若卷组中有剩余空间，扩展卷组中的逻辑卷

```
# lvextend <-L +逻辑卷增量> <逻辑卷设备名称>
```

```
# lvextend <-l +PE值> <逻辑卷设备名称>
```
- 对已扩展的逻辑卷中的文件系统进行容量扩展

```
# resize2fs <分区或逻辑卷设备名>
```

扩展逻辑卷举例

- 将两个物理卷扩展到已存在的VolGroup00卷组中

```
# vgextend VolGroup00 /dev/sdb1 /dev/sdb2
```
- 在VolGroup00卷组中扩展LogVolHome逻辑卷，扩展大小为6GB

```
# lvextend -L +6G /dev/VolGroup00/LogVolHome
```
- 为home文件系统（ext3）扩充容量

```
# resize2fs -f /dev/VolGroup00/LogVolHome
```

- 使用umount命令卸载文件系统
- 使用e2fsck命令检查文件系统
- 使用resize2fs命令缩减文件系统容量
- 缩减逻辑卷
 - # lvreduce <-L -逻辑卷增量> <逻辑卷设备名称>
 - # lvreduce <-l -PE值> <逻辑卷设备名称>

LVM 常用命令集

任务	PV	VG	LV
创建	<code>pvccreate</code>	<code>vgcreate</code>	<code>lvcreate</code>
删除	<code>pvremove</code>	<code>vgremove</code>	<code>lvremove</code>
显示信息	<code>pvs</code>	<code>vgs</code>	<code>lvs</code>
扫描列表	<code>pvscan</code>	<code>vgscan</code>	<code>lvscan</code>
显示属性	<code>pvdisplay</code>	<code>vgdisplay</code>	<code>lvdisplay</code>
更改属性	<code>pvchange</code>	<code>vgchange</code>	<code>lvchange</code>
扩展		<code>vgextend</code>	<code>lvextend</code>
缩减		<code>vgreduce</code>	<code>lvreduce</code>

- 可以使用 `lvm help` 命令显示上述命令的功能。
- 可以使用命令参数 `-h` 查看每个命令的使用方法，如：
`# lvreduce -h`

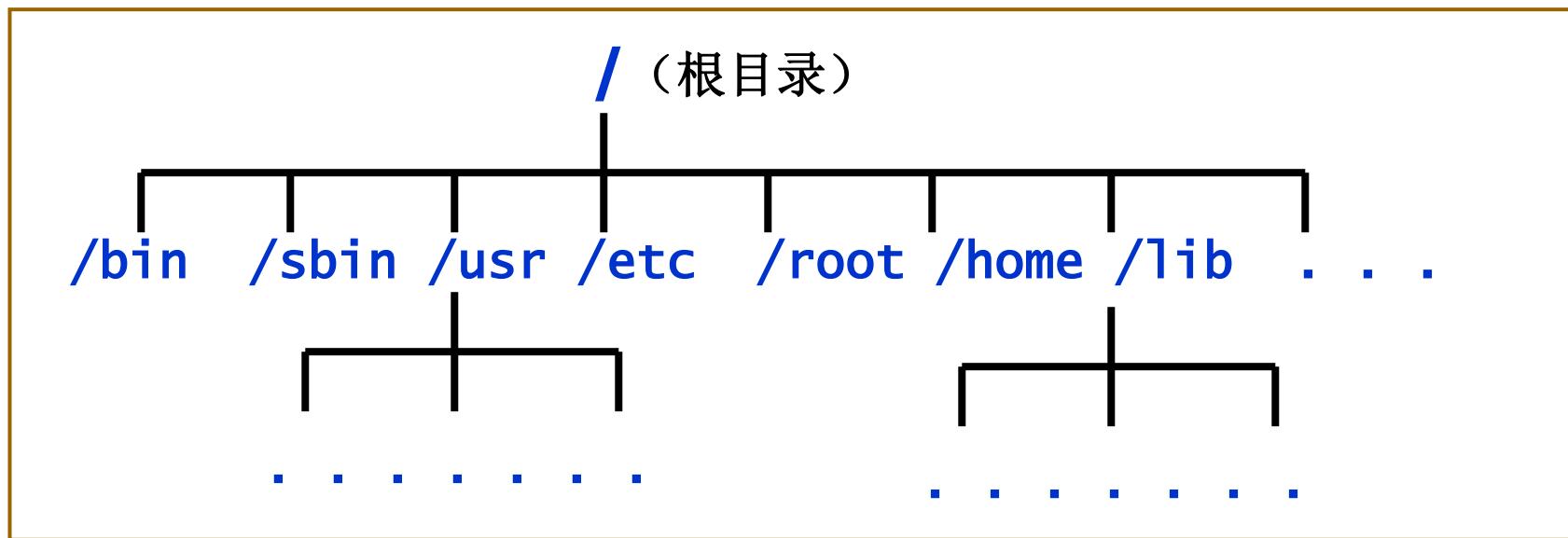
文件系统的概念

文件系统（File System） 的各种定义

- 文件系统是包括在一个磁盘（硬盘、光盘及其它存储设备）上的目录结构；一个磁盘设备可以包含一个或多个文件系统。
- 文件系统是在一个磁盘（硬盘、光盘及其它存储设备）上组织文件的方法。
- 文件系统是文件的数据结构或组织方法。
- 文件系统是基于被划分的存储设备上的一种文件的命名、存储、组织及读取的方法。
- 一个文件系统是有组织存储文件或数据的方法，目的是易于查询和存取。文件系统是基于一个存储设备，比如硬盘或光盘，并且包含文件物理位置的维护。

Linux的文件系统结构

- Linux下的所有文件和目录以一个树状的结构组织构成了 Linux 中的文件系统。
 - Linux文件系统标准（Linux File System Standard, FSSTND）
 - 文件系统层次结构标准（File System Hierarchy Standard, FHS）



- Linux的内核采用了称之为**虚拟文件系统**（**Virtual File System, VFS**）的技术，因此Linux 可以支持多种不同的文件系统类型。
- Linux可支持的文件系统
 - Linux目前几乎支持所有的UNIX类的文件系统，如HFS、XFS、JFS、Minix FS 及 UFS 等
 - Linux 支持 NFS 文件系统
 - Linux 也支持 NTFS 和 vfat（FAT32）

Linux支持的日志文件系统



- Linux 支持
 - ext3/ext4
 - JFS (IBM)
 - XFS (SGI)
 - Reiserfs
- 日志文件系统的优点
 - 提高了文件的存储安全性
 - 降低了文件被破坏的机率
 - 缩短了对磁盘的扫描时间
 - 减少了磁盘整体扫描次数

Linux下常见的文件系统

- **ext2/ext3/ext4**
 - Linux使用的**标准文件系统**
- **swap**
 - 交换文件系统
- **FAT32/vfat**
 - Windows文件系统
- **NFS**
 - 网络文件系统
- **iso9660**
 - 标准光盘文件系统

使用Linux文件系统的一般方法

- 在硬盘上创建分区或逻辑卷
 - 可以使用**fdisk**命令创建分区。
 - 可以使用**LVM**的相关命令创建逻辑卷
- 在分区/**LV**上建立文件系统
 - 类似于在**Windows**下进行格式化操作。
- 挂装文件系统到系统中
 - 手工挂装：使用**mount**命令
 - 启动时自动挂装：编辑“**/etc/fstab**”添加相应的配置行。
- 卸装文件系统
 - 对于可移动介质上的文件系统，当使用完毕可以使用**umount**命令实施卸装操作。

挂装和卸载文件系统

挂装文件系统

——mount命令

- 功能：挂装文件系统

- 格式

`mount [选项] [<分区设备名>] [<挂装点>]`

- 常用选项

- `-t <文件系统类型>`：指定文件系统类型

- `-r`：使用只读方式来挂载

- `-a`：挂装/etc/fstab文件中记录的设备

- `-o iocharset=cp936`：使挂装的设备可以显示中文文件名

- `-o loop`：使用回送设备挂装ISO文件和映像文件

mount命令举例

```
$ mount
```

```
# mount -t ext3 /dev/sdb1 /opt
```

```
# mount -t ext4 /dev/sdb2 /data
```

```
# mount -t vfat /dev/hda6 /mnt/win
```

```
# mount -t vfat -o iocharset=cp936 /dev/hda8 /mnt/win2
```

```
# mount -o loop CentOS-6.5-x86_64-bin-DVD1.iso /mnt/iso
```

```
# mount -o remount,ro /data
```

卸载文件系统

- **umount命令的格式**
 - # umount < 分区设备名或挂装点>
- **举例**
 - # umount /dev/hda6
 - # umount /dev/sdb1
 - # umount /opt

挂装/卸装 文件系统的注意事项

- 挂装点目录必须存在
- 应该在挂装目录的上级目录下进行挂装操作
- 不该在同一个挂装点目录下挂装两个文件系统
- 当文件系统处于“**busy**”状态时不能进行卸装
- 文件系统何时处于“**busy**”状态
 - 文件系统上面有打开的文件
 - 某个进程的工作目录在此文件系统上
 - 文件系统上面的缓存文件正在被使用

- **fuser**命令可以根据文件（目录、设备）查找使用它的进程，同时也提供了杀死这些进程的方法。

- 使用举例

- 查看挂接点有哪些进程需要杀掉

- ```
fuser -cu /mount_point
```

- 杀死这些进程（向其发送[SIGKILL, 9]信号）

- ```
# fuser -ck /mount_point
```

- 查看是否还有进程在访问挂接点

- ```
fuser -c /mount_point
```

- 卸载挂接点上的设备

- ```
# umount /mount_point
```

可移动介质

- 挂载（**Mounting**）意味着使外来的文件系统看起来如同是主目录树的一部分。
- 访问前、介质必须被挂载
- 摘除时，介质必须被卸载
- 按照默认设置，非根用户只能挂载某些设备（光盘、**DVD**、软盘、**USB**等等）
- 挂载点通常在 **/media** 下

- 在Gnome/KDE中自动挂载
- 使用命令行手工挂载
 - CD/DVD（只读）
`mount /media/cdrom`
 - CD/DVD（刻录）
`mount /media/cdrecorder`
- 卸载
 - `umount /media/cdrom`
 - `umount /media/cdrecorder`
 - `eject` —— 卸载并弹出光盘

- 被内核探测为SCSI设备
 - /dev/sdaX、/dev/sdbX、或类似的设备文件
- 在Gnome/KDE中自动挂载
 - 在[计算机]窗口中创建图标
 - 挂载在/media/<设备ID>下
 - <设备ID>被厂商内建在设备中
- 命令行下手动挂载/卸载
 - `mount -t vfat /dev/sda1 /mnt/usb1`
 - `umount /mnt/usb1`

- 必须被手工挂载和卸载
 - `mount /media/floppy`
 - `umount /media/floppy`
- DOS软盘可以使用mtools工具
 - 透明地挂载和卸载设备
 - 使用DOS命名规则
 - `mmdir a:`
 - `mcopy /home/file.txt a:`

直接挂装使用映像文件

- 用mount命令加-o loop选项挂装光盘镜像文件

```
# mount -o loop <ISO文件名> <挂装点>
```

- 例如:

```
# mount -o loop CentOS-6.5-x86_64-bin-DVD1.iso  
/mnt/iso
```

- 卸装ISO文件

```
# umount /mnt/iso
```

类似地，可以用同样的方法挂装使用 **IMG** 映像文件。

系统启动挂装表

系统启动时 自动挂装文件系统

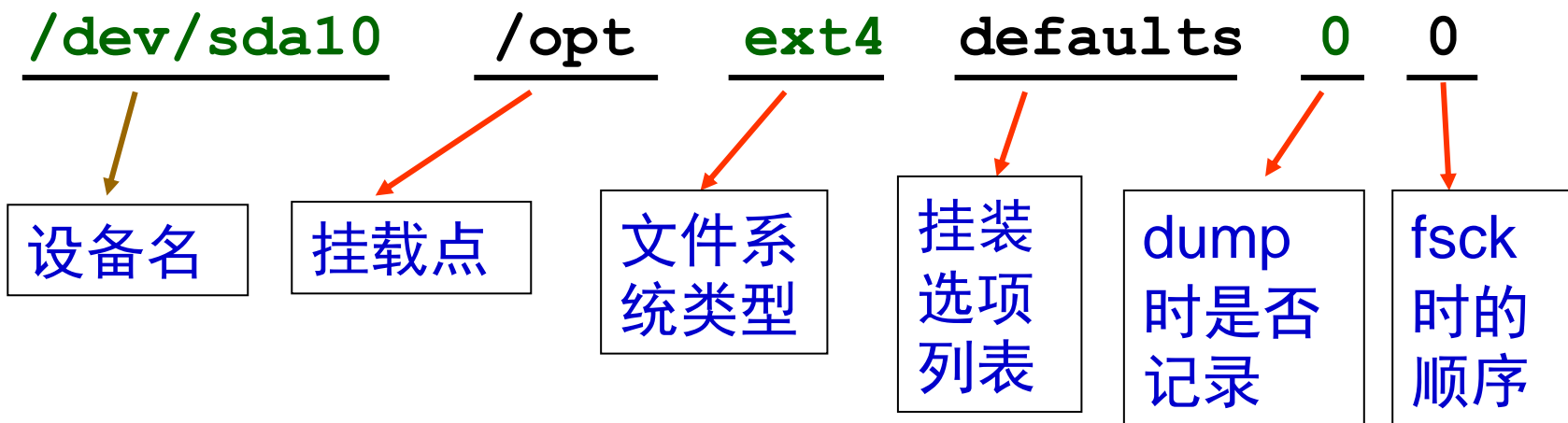
■ /etc/fstab

- ❑ **fstab** (file system table) 是一个纯文本文件，开机后，系统会自动搜索该文件中的内容，对列于该文件中的文件系统进行自动挂载。
- ❑ 系统重启时保留文件系统体系结构
- ❑ 配置文件系统体系结构
- ❑ 被 mount、fsck 和其它程序使用
- ❑ 使用 `mount -a` 命令挂载 `/etc/fstab` 中的所有文件系统
- ❑ 可以在设备栏使用文件系统卷标

/etc/fstab文件的格式

■ /etc/fstab包含的信息

- 每一行说明一个文件系统的挂载信息
- 每一行由 6 列信息组成，列与列之间用 TAB 键隔开，一般格式如下：



/etc/fstab文件的列信息

分区或LV	挂装点	文件系统类型	挂装选项	备份频率	检查顺序
fs_spec	fs_file	fs_type	fs_options	fs_dump	fs_pass

- **fs_spec**: 设备或远程文件系统
- **fs_file**: 挂装点目录
- **fs_type**: 文件系统类型
- **fs_options**: 文件系统挂载选项
- **fs_dump**: 被” dump”命令用来检查一个文件系统应该以多快频率进行转储，若不需要转储则该字段为“0”
- **fs_pass**: 被” fsck”命令用来决定在启动时需要被扫描的文件系统的顺序，若无需在启动时扫描则该字段为“0”

文件/etc/fstab实例

#	<file system>	<mount point>	<type>	<options>	<dump>	<pass>
	LABEL=/	/	ext4	defaults	1	1
	none	/dev/pts	devpts	gid=5,mode=620	0	0
	LABEL=/home	/home	ext4	defaults	1	2
	none	/proc	proc	defaults	0	0
	none	/dev/shm	tmpfs	defaults	0	0
	LABEL=/usr	/usr	ext4	defaults	1	2
	/dev/sda5	swap	swap	defaults	0	0
	/dev/cdrom	/mnt/cdrom	udf,iso9660	noauto,owner,kudzu,ro	0	0
	/dev/fd0	/mnt/floppy	auto	noauto,owner,kudzu	0	0
	/dev/hda1	/mnt/win_c	vfat			
			defaults,pagecode=936,iocharset=cp936,umask=000		0	0

挂装选项

选项	说明
defaults	使用 rw, suid, dev, exec, auto, nouser 和 async 挂装设备
acl/noacl	支持/不支持 POSIX Access Control Lists (ACL)
async	以非同步方式（延迟写）执行文件系统的输入输出操作
atime/noatime	每次访问文件时都 更新/不更新 文件的访问时间， atime 为默认值， noatime 会提高文件系统的访问速度
auto/noauto	使用 mount -a 或开机时 会/不会自动挂装
dev/nodev	可以/不可 解读文件系统上的字符或区块设备

挂装选项（续）

选项	说明
<code>exec/noexec</code>	可以/不可 执行文件系统上的二进制文件
<code>suid/nosuid</code>	开启/禁用 SUID 和 SGID 设置位
<code>user/nouser</code>	允许普通用户/仅超级用户 挂装这个文件系统
<code>users</code>	使一般用户可以挂装/卸载,用于桌面环境,包含 <code>noexec</code> 、 <code>nosuid</code> 、 <code>nodev</code> 选项
<code>rw/ro</code>	以 读写/只读 方式挂装文件系统。
<code>remount</code>	重新挂装已挂装的文件系统（通常用于 <code>mount</code> 命令行）

交换空间

交换分区和交换文件

- Linux系统使用交换分区/文件实现虚拟内存技术，它是系统RAM的补充。
- 基本设置包括：
 - 创建交换分区或者文件
 - 使用 `mkswap` 创建交换文件系统
 - 在 `/etc/fstab` 文件中添加适当的条目
 - 使用 `swapon -a` 激活交换空间
- 生成交换文件
 - `dd if=/dev/zero of=/swapfile bs=1024 count=X`

- 功能：从标准输入、文件或设备读取数据，依照指定的格式来转换数据，再输出至文件、设备或标准输出。
- 格式：**dd** <选项=值>
- 常用选项
 - **if=file**: 输入文件名或设备名，确省标准输入
 - **of=file**: 输出文件名或设备名，确省标准输出
 - **bs=n**: 同时设置输入输出块的大小为n
 - **count=n**: 仅仅读取指定的n块数据

dd命令举例

- 将软盘的内容制作成映像文件备份到硬盘
`# dd if=/dev/fd0 of=fdtemp.img`
- 将硬盘上的软盘映像文件恢复至软盘
`# dd if=fdtemp.img of=/dev/fd0`
- 将硬盘首扇备份到文件bootsector.bin中
`# dd if=/dev/sda of=bootsector.bin bs=512 count=1`
- 创建一个800MB的交换文件
`# dd if=/dev/zero of=/swapfile bs=4096 count=200000`

EXT 2/3/4文件系统管理

创建文件系统

■ 前端命令mkfs的格式

mkfs -t <fstype> -c <分区设备名>

- -t fstype: 指定文件系统类型
- -c: 建立文件系统前先检测有无坏块

■ 举例

mkfs -t ext3 -c /dev/hda2

mkfs -t vfat /dev/hdb2

mke2fs -c /dev/hda2

mkfs.ext4 /dev/sda1

- **fsck**是操作系统扫描文件系统内容检查内部一致性的工具。
- 主要功能
 - 检测并修正链接中断的目录
 - 检测并修正错误时间标记
 - 检测并修正指向错误磁盘区域的i-node
- 命令格式
 - **fsck** [选项][-t 文件系统类型] <设备名> [特定文件系统的附加选项]

检查文件系统（续）

■ fsck举例

- 检查文件系统，对所有问题回答“yes”

```
# fsck -t ext4 /dev/sdb2
```

- 自动检查并修复文件系统

```
# e2fsck -p /dev/sda5
```

- 强制进行文件系统检查，标识损坏区块，对所有问题回答“yes”

```
# fsck.ext3 -f -y /dev/hda2
```

检查文件系统注意事项

- 一般情况下，无需用户手动执行**fsck**命令。
- 在系统启动过程中，一旦系统检测到了不一致就会自动运行**fsck**命令。
- 手动执行**fsck**命令，应该在单用户模式且文件系统被卸装的情况下进行。

显示和调整文件系统属性

- 显示文件系统属性参数
 - `tune2fs -l <device>`
 - `dumpe2fs -h <device>`
- 可调整的文件系统属性参数
 - 保留块
 - 默认挂载选项
 - `fsck` 频率

■ 格式

□ `tune2fs [<选项>] <设备名>`

■ 常用选项

- `-c`: 表示文件系统在 `mount` 次数达到设定后, 需要运行 `fsck` 检查文件系统。
- `-i`: 文件系统的检查间隔时间。系统在达到时间间隔时, 自动检查文件系统。
- `-j`: 为 `ext2` 文件系统添加文件系统日志, 转换为 `ext3` 文件系统。
- `-m`: 设置保留的空间百分比, 预设为 `5%`。
- `-o`: 设置默认加载参数。
- `-L`: 为指定设备设置卷标, 不大于 `16` 字符。

tune2fs命令举例

- 设置默认挂载参数

```
# tune2fs -o acl,user_xattr /dev/sda1
```

- 将ext2文件系统转换成ext3文件系统。

```
# tune2fs -j /dev/sdb2
```

- 将每两次完整性检查之间的装载次数调整为10次。

```
# tune2fs -c 10 /dev/sdb5
```

- 将每两次完整性检查的时间间隔调整为2个月。

```
# tune2fs -i 2m /dev/sdb5
```

- 调整指定文件系统的预留块比例为2%。

```
# tune2fs -m 2 /dev/sdb5
```

- 禁用强制文件系统检查

```
# tune2fs -i0 -c0 /dev/sda1
```

文件系统的LABEL和UUID

- 标识块设备的传统方法
 - 设备名 (`/dev/sda1`, `/dev/my_vg/my_lv`)
- 标识块设备的其他方法
 - 文件系统的LABEL
 - `e2label device [fslabel]`
 - 文件系统的UUID
 - 在生成文件系统时自动为设备指定UUID

使用LABEL和UUID

■ 显示Label

`e2label <设备名>`

■ 根据Label查找设备名

`findfs LABEL=<label>`

■ 在 mount 命令中使用 LABEL和UUID

`mount [options] -L <label> <mount_point>`

`mount [options] -U <uuid> <mount_point>`

■ 在 /etc/fstab 中使用 LABEL和UUID

`UUID=uuid /mountpoint ext4 default 1 2`

`LABEL=label /mountpoint ext4 default 1 2`

■ 显示UUID

`blkid <设备名>`

■ 根据UUID查找设备名

`findfs UUID=<uuid>`

磁盘限额

- 磁盘限额是系统管理员用来监控和限制用户或组对磁盘的使用的工具。
- 磁盘限额可以从两方面限制
 - 限制用户或组可以拥有的**inode**数（即文件个数）
 - 限制分配给用户或组的磁盘块的数目
- 磁盘配额是以每一使用者，每一文件系统为基础的。如果使用者可以在超过一个以上的文件系统上建立文件，那么必须在每一文件系统上分别设定。

磁盘限额的限制策略

- 硬限制：超过此设定值后不能继续存储新的文件。
- 软限制：超过此设定值后仍旧可以继续存储新的文件，同时系统发出警告信息，建议用户清理自己的文件，释放出更多的空间。
- 时限：超过软限制多长时间之内（默认为7天）可以继续存储新的文件。

配置磁盘限额的前提

- 查看内核是否支持
 - 默认安装时，是支持quota的。
- 查看系统中是否安装了quota的RPM
 - Red Hat/CentOS 默认已经安装。
- 查看启动脚本是否在系统启动时打开了quota
 - RHEL/CentOS默认已经打开。

磁盘限额配置步骤

(1) 启用文件系统的quota功能

■ 编辑/etc/fstab

- 在中添加文件系统挂载选项

usrquota,grpquota

- 例如

```
LABEL=/home    /home    ext3
                defaults,grpquota,usrquota  1  2
```

■ 重新挂装文件系统

```
# mount -o remount /home
```

磁盘限额配置步骤

(2) 创建quota数据库并开启quota

■ 创建quota数据库

```
# quotacheck -cmvug /home
```

■ 开启quota功能

```
# quotaon -avug
```

磁盘限额配置步骤

(3) 设置用户和组的quota

- 交互式编辑配额
 - edquota
- 命令式设置配额
 - setquota
- 将参考用户/组的配额复制给其他用户/组
 - edquota -p <protoname>
 - setquota -p <protoname>

edquota命令

- 编辑指定用户的配额
 - ❑ `edquota [-u] [-f filesystem] <username>`
- 编辑指定组的配额
 - ❑ `edquota -g [-f filesystem] <groupname>`
- 编辑指定用户的配额时限
 - ❑ `edquota -t [-u] [-f filesystem]`
- 编辑指定组的配额时限
 - ❑ `edquota -t -g [-f filesystem]`

-f filesystem 表示对指定的文件系统设置配额，省略时表示对所有启用了quota的文件系统进行设置。

setquota命令

■ 为指定用户的设置配额

- `setquota [-u] <username> <块软限制 块硬限制 inode 软限制 inode硬限制> <-a|文件系统>`

■ 为指定组的设置配额

- `setquota -g <groupname> <块软限制 块硬限制 inode 软限制 inode硬限制> <-a|文件系统>`

若对所有启用了quota的文件系统设置配额，则使用-a；否则，若对指定的文件系统设置配置，则需要指定文件系统。

- `setquota -t -g <块时限 inode时限> <-a|文件系统>`

使用参考用户或组复制配额

- 将参考用户 **protouname** 的限额设置复制给待设置用户 **username**
 - ❑ `edquota [-u] -p <protouname> <username>`
 - ❑ `setquota [-u] -p <protouname> <username> <-a|filesystem>`
- 将参考组 **protogname** 的限额设置复制给待设置组 **groupname**
 - ❑ `edquota -g -p <protogname> <groupname>`
 - ❑ `setquota -g -p <protogname> <groupname> <-a|filesystem>`

查看磁盘限额

- 查看指定用户或组的quota设置
 - # quota [-vl] [-u <username>]
 - # quota [-vl] [-g <groupname>]
 - # quota -q
- 显示文件系统的磁盘限额汇总信息
 - 显示指定文件系统的磁盘限额汇总信息
 - # repquota [-ugv] filesystem...
 - 显示所有文件系统的磁盘限额汇总信息
 - # repquota [-augv]

- 简述硬盘的技术指标？如何挑选服务器硬盘？
- **fdisk/parted**命令有哪些常用的子命令？含义是什么？
- 什么是**MBR/GPT**，它存放了什么信息？
- 使用**LVM**比使用固定分区有哪些优点？
- 简述**PV-VG-LV-PE**的逻辑关系。

本章思考题（续1）

- 什么是Linux文件系统？Linux下常用的文件系统有哪些？
- 非日志文件系统和日志文件系统有何区别？
- 简述在Linux环境下使用文件系统的一般方法。
- 如何创建文件系统？创建文件系统的操作类似于Windows下的什么操作？
- 如何设置ext2/3/4文件系统的属性？
- 如何挂装和卸装文件系统？
- 如何使用可移动存储介质（软盘、光盘、USB盘）？
- 如何直接挂装使用ISO文件和IMG文件？
- 如何在系统启动时自动挂装文件系统？简述/etc/fstab文件各个字段的含义。
- 简述添加新硬盘并扩展现有逻辑卷的步骤。

- 学会使用**fdisk/parted**命令进行硬盘分区。
- 熟悉**LVM**的命令工具。
- 学会扩展和缩减逻辑卷的大小。
- 学会创建不同类型的文件系统。
- 学会挂装和卸装文件系统。
- 学会使用可移动存储介质（软盘、光盘、**USB**盘）。
- 学会使用**ext2/ext3/ext4**文件系统的维护命令。
- 学会操作系统挂装表文件**/etc/fstab**。

- 学会设置ext4文件系统的磁盘限额。
- 学习RAID、DRBD的相关概念。
- 学习在Linux下使用mdadm工具配置和管理软RAID。
- 熟悉并对比不同日志文件系统的优缺点，以便在实际工作中选择使用。
- 了解Red Hat集群文件系统GFS和MooseFS分布式文件系统的应用和管理。

RAID

磁盘阵列（RAID）的功能



- 整合现有的磁盘空间
- 提高磁盘读取效率
- 提供容错性

磁盘阵列的种类

- 硬件磁盘阵列
 - 使用磁盘阵列控制卡
 - 使用 **dmraid** 工具管理
- 软件磁盘阵列
 - 由 **Linux** 的内核仿真磁盘阵列的功能
 - 使用 **mdadm** 工具管理

磁盘阵列的级别

	RAID0	RAID1	RAID5
模式	Stripe	Mirror	Stripe set with Parity
磁盘数量	2个或以上	2个	3个或以上
磁盘容量	可不同	必须相同	必须相同
可用空间	多个磁盘容量之和	单个磁盘容量	(磁盘数-1)*单盘容量
优点	读快，写快	容错，读快	读快，容错
缺点	无容错	写速普通，浪费空间	写入慢，需更多的计算能力

- 由 Linux 的内核仿真磁盘阵列的功能
 - 多个磁盘合成一个“阵列”来提供更好的性能、冗余，或者两者兼具
 - 软RAID的组成单元是分区
 - 分区类型为"fd - Linux raid autodetect"。
 - 支持多种“RAID级别”
 - RHEL/CentOS允许使用同一个磁盘的不同分区产生磁盘阵列，也可以由多个磁盘的分区产生磁盘阵列；但前者无法提供容错性，也无法提升读取的效率，仅用于实验环境。
- m

创建软 RAID

- 首先准备若干类型为“fd”的分区
- 创建并启用软RAID设备

```
# mdadm --create|-C <MD_DEVICE> \ # 创建MD设备
-I LEVEL|--level=LEVEL \ # 指定RAID级别，如：0|1|5
-n N|--raid-devices=N \ # 指定组成RAID的设备数量
[-x M|--spare-devices=M] \ # 指定备用的附加设备数量
[-a yes|--auto=yes] \ # 自动重建MD设备文件
[-v|--verbose] \ # 显示详细地创建过程
<DEVICES>... \ # 指定组成RAID的分区设备
```

- 为软RAID设备创建文件系统

创建软 RAID 举例

■ 创建并启用RAID1设备

```
# mdadm --verbose --create /dev/md0 \  
--level raid1 --raid-devices 2 \  
/dev/sda10 /dev/sdb2
```

■ 创建并启用RAID5设备

```
# mdadm -v -C /dev/md1 -a yes -l 5 -n 3 -x 1 \  
/dev/sda6 /dev/sd{b,c,d}1
```

检查 RAID 设备的状况

■ 命令行方式

- 格式: `mdadm --detail -D <MD_DEVICE>`

- 例如:

 - # `mdadm --detail /dev/md0`

 - # `mdadm -D /dev/md1`

■ 服务方式

- RHEL/CentOS 提供了 `mdmonitor` 服务

- 启动 `mdmonitor` 服务会帮你监控每一个 MD 设备

- 一旦发现某个 MD 磁盘故障, 就会给超级用户发送提醒邮件

解决MD设备故障

——具备容错能力的 RAID1,5

- 在线卸载出现故障的磁盘分区
 - 使用 **mdadm MD_DEVICE --remove DEVICE**
- 拔出故障的磁盘驱动器
 - 若计算机不支持热插拔，必须先关机
- 安装正常的磁盘驱动器，创建新的组成单元
 - 使用 **fdisk** 命令创建新的类型为“fd”的分区
 - 分区大小应该与原来的（出现故障的）一致
- 在线加入MD设备的新的组成单元
 - 使用 **mdadm MD_DEVICE --add DEVICE**

- mdadm 工具提供了磁盘故障仿真功能
 - 方便管理员进行故障测试，以免在故障真正出现时手足无措
 - 仅用于具有容错能力的阵列（**RAID1** 和 **RAID5**）
- 命令格式

```
# mdadm <MD_DEVICE> --set-faulty|-f <DEVICE>...
```
- 命令举例
 - 模拟 /dev/md0 设备中的 /dev/sda1 组成单元故障

```
# mdadm /dev/md0 -f /dev/sda1
```

启用与停用MD设备

- 使用 `mdadm --create|-C` 命令创建MD设备时会自动启用
- 开机时会根据 `/etc/mdadm.conf` 文件的设置启动所有的MD设备
- 手工启用与停用MD设备
 - 启用: `mdadm --assemble --scan [MD_DEVICE]`
 - 停用: `mdadm --stop --sacn [MD_DEVICE]`
 - 省略 MD_DEVICE 时将依据 `/etc/mdadm.conf` 中的定义, 启用/停用所有的MD设备