



物理存储及逻辑卷管理

课程介绍

- 本章主要讲述磁盘存储、逻辑卷存储的基础概念以及如何管理、使用、存储，以及对一些常用操作命令的介绍。
- 学完本课程后，您将能够：掌握磁盘存储的挂载和使用方法；掌握逻辑卷的管理方法

目录

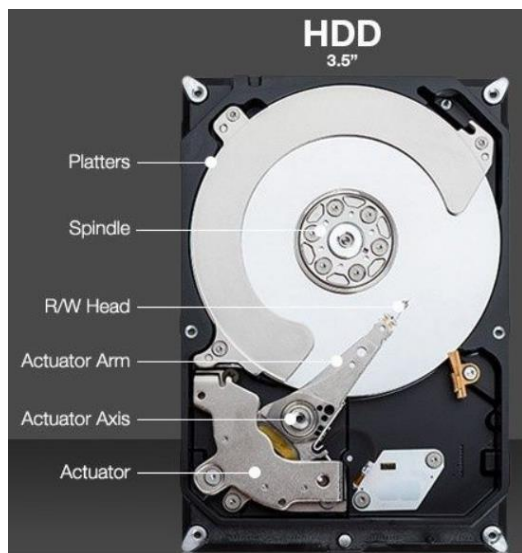
CONTENT

01 磁盘存储挂载与使用

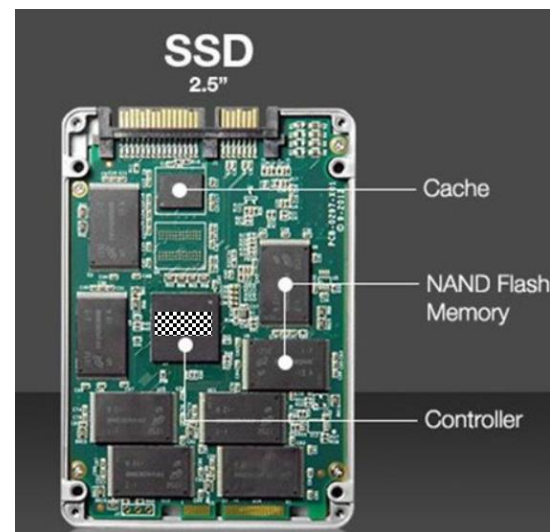
02 逻辑卷管理

磁盘大类

HDD, (Hard Disk Drive的缩写) :
由一个或者多个铝制或者玻璃制成的磁性碟片, 磁头, 转轴, 控制电机, 磁头控制器, 数据转换器, 接口和缓存等几个部分组成。工作时, 磁头悬浮在高速旋转的碟片上进行读写数据。机械硬盘是集精密机械、微电子电路、电磁转换为一体的电脑存储设备。



SSD,固态硬盘 (Solid State Drive) :
固态硬盘 (SSD) 是由多个闪存芯片加主控以及缓存组成的阵列式存储, 属于以固态电子存储芯片阵列制成的硬盘。相对机械硬盘, 读取速度更快, 寻道时间更小, 可加快操作系统启动速度和软件启动速度。



磁盘接口说明

磁盘接口类型	说明
IDE (Integrated Device Electronics:电子集成驱动器)	最初硬盘的通用标准，任何电子集成驱动器都属于IDE，甚至包括SCSI；
SATA (Serial ATA)	SATA的出现将ATA和IDE区分开来，而IDE则属于Parallel-ATA(并行ATA)。
SCSI (Small Computer System Interface: 小型计算机	SCSI硬盘就是采用这种接口的硬盘。SAS(Serial Attached SCSI)就是串口的SCSI接口。一般服务器硬盘采用这两类接口，其性能比上述两种硬盘要高，稳定性更强，支持热插拔，但是价格高，容量小，噪音大
FC (FibreChannel)	使光纤通道能够直接作为硬盘连接接口，为高吞吐吞吐量性能密集型系统的设计者开辟了一条提高I/O性能水平的途径。

Linux如何查看磁盘信息 (1)

`fdisk -l` 用于查看系统所有磁盘的信息，包括已挂载和未挂载磁盘

```
[root@openEuler ~]# fdisk -l
Disk /dev/vda: 40 GiB, 42949672960bytes, 83886080sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: AA321D82-C833-4D3E-885C-52FC0ADF3860

Device      Start      End Sectors  Size Type
/dev/vda1    2048    411647  409600  200M EFI System
/dev/vda2    411648  2508799  2097152   1G Linux filesystem
/dev/vda3    2508800 8388403 5889600  8.8G Linux LVM

Disk /dev/mapper/openEuler-root: 34.82 GiB, 37367054336bytes, 72982528sectors
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 /dev/mapper/openEuler-swap: 4 GiB, 4294967296bytes, 8388608sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Linux如何查看磁盘信息 (2)

df -h 命令查看系统挂载、磁盘空间大小和利用率

```
[root@openEuler ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs			1.2G	0	1.2G 0% /dev
tmpfs	1.5G	0	1.5G	0%	/dev/shm
tmpfs	1.5G	18M	1.5G	2%	/run
tmpfs	1.5G	0	1.5G	0%	/sys/fs/cgroup
/dev/mapper/openEuler-root	35G	4.4G	28G	14%	/
tmpfs	1.5G	64K	1.5G	1%	/tmp
/dev/vda2	976M	125M	785M	14%	/boot
/dev/vda1			200M	5.8M	195M 3% /boot/efi
tmpfs	298M	0	298M	0%	/run/user/0
tmpfs	298M	0	298M	0%	/run/user/993

磁盘分区

磁盘分区可以将硬盘驱动器划分为多个逻辑存储单元，这些单元称为分区。通过将磁盘划分为多个分区，系统管理员可以使用不同的分区执行不同功能。

磁盘分区的好处：

限制应用或用户的可用空间。

允许从同一磁盘进行不同操作系统的多重启动。

将操作系统和程序文件与用户文件分隔。

创建用于操作系统虚拟内存交换的单独区域。

限制磁盘空间使用情况,以提高诊断工具和备份映像的性能。

磁盘分区类型

通常所说的“硬盘分区”就是指修改磁盘分区表，注意以下情况：

考虑到磁盘的连续性，一般建议将扩展分区放在最后面的柱面内。

一个硬盘只有一个扩展分区，除去主分区，其它空间都分配给扩展分区。

硬盘容量=主分区+扩展分区；扩展分区容量=各个逻辑分区容量之和

磁盘

主分区

扩展分区

逻辑分区1

逻辑分区n

主分区：也称为主磁盘分区，主分区中不能再划分其他类型的分区，因此每个主分区都相当于一个逻辑磁盘。

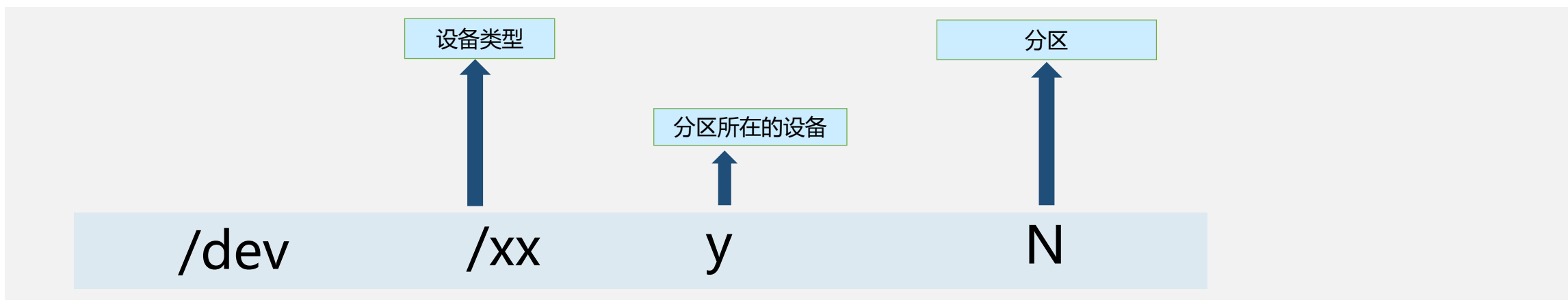
扩展分区：扩展分区与逻辑分区是为了突破分区表中只能保存4个分区的限制而出现的，扩展分区不能直接使用，需要在扩展分区内划分一个或多个逻辑分区后才能使用。

逻辑分区：在扩展分区上面，可以创建多个逻辑分区，硬逻辑分区是盘上一块连续的区域，它是扩展分区的组成部分。

磁盘分区命名规则

在Linux中，没有盘符这个概念，通过设备名来访问设备，设备名存放在/dev目录中。

命名规则如下：



设备类型：通常是hd（IDE磁盘），sd（SCSI磁盘），fd（软驱），vd（virtio 磁盘）

分区所在的设备：例如/dev/hda（第一个IDE磁盘）或/dev/sdb，（第二个SCSI磁盘）

分区：前4个分区（主分区或扩展分区）用数字1到4，逻辑分区从5开始，例如/dev/hda3，是第一个IDE磁盘上第三个主分区或扩展分区；/dev/sdb6是第二个SCSI硬盘上的第二个逻辑分区

注：Linux中，SSD、SAS、SATA类型的硬盘，都用sd来标识，IDE硬盘属于IDE接口类型的硬盘，用hd来标识

磁盘分区方案 - MBR

MBR分区方案

MBR的全称：Master Boot Record（主启动记录）

它位于硬盘开始部分的一个特殊扇区，扇区内部包含已安装系统的启动加载器和硬盘内逻辑分割区的信息。在启动操作系统时，会从扇区内使用一段代码来启动系统。

MBR格式硬盘启动的过程：

使用MBR磁盘分区表格的硬盘在计算机启动时，会先启动主板的BIOS，随后BIOS加载MBR，再从MBR启动操作系统

MBR的局限性主要体现在三个方面：

MBR只适用于最大容量2T的硬盘，如果是更大容量的硬盘使用MBR，那么多出来部分则无法识别；

MBR只支持最大4个主分区，如果要创建更多分区，就必须将其中一个主分区作为“扩展分区”，并再其中创建逻辑分区。

分区和引导数据都存储在同一个地方，如果这个数据被覆盖或损坏，用户将无法启动计算机

fdisk分区工具使用

fdisk是传统的Linux硬盘分区工具，也是Linux系统中最常用的一种硬盘分区工具之一，但不支持大于 2TB 的分区。

命令格式如下：

语法：fdisk(选项)(参数)

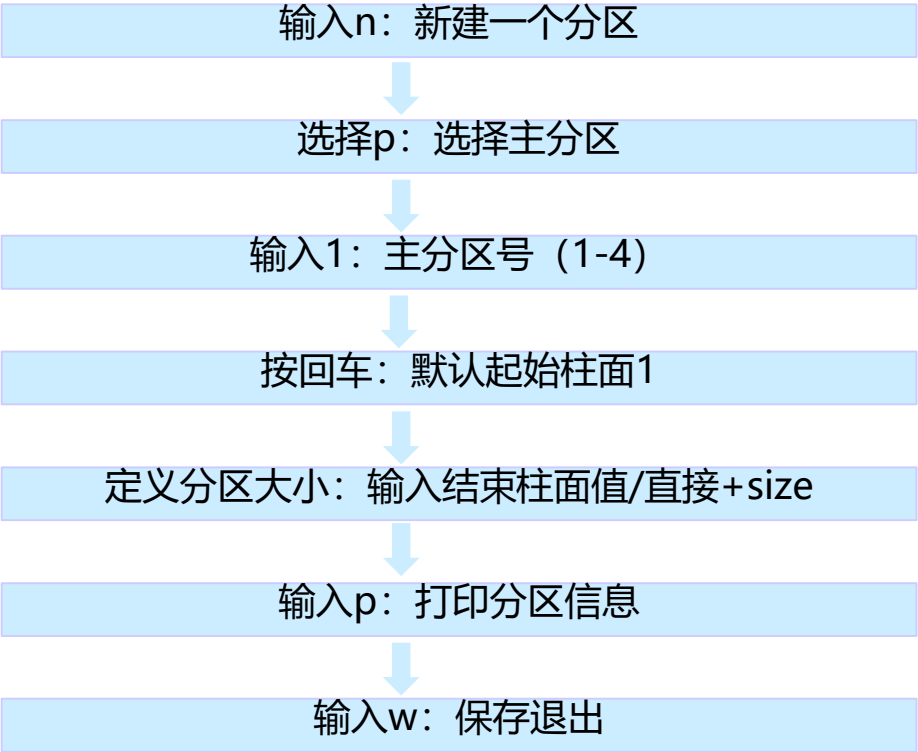
常用选项释义如下：

- b <分区大小>：指定每个分区的大小；
- l：列出指定的外围设备的分区表状况；
- s <分区编号>：将指定的分区大小输出到标准输出上，单位为区块；
- u：搭配"-l"参数列表，会用分区数目取代柱面数目，来表示每个分区的起始地址；
- v：显示版本信息。

fdisk硬盘分区操作

选择好具体某块硬盘后，进入交互模式进行分区操作

如：fdisk /dev/sdb创建主分区流程：



#交互式命令参数如表格所示

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 Linuxswap 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

磁盘分区方案 - GPT

GPT分区方案

随着硬盘驱动器容量的不断增长,老旧的MBR分区方案的2 TB磁盘和分区大小限制已不再是理论上的限制,而是在生产环境中越来越经常遇到的实际问题, 因此,GPT正在取代传统的MBR方案进行磁盘分区。

GPT意为GUID分区表, 驱动器上的每个分区都有一个全局唯一的标识符 (globally unique identifier, GUID) , 对于运行统一可扩展固件接口(UEFI) 固件的系统,GPT是在物理硬盘上布置分区表的标准。

没有主分区和逻辑分区之分, 每个硬盘最多可以有128个分区, GPT 为逻辑块地址分配64位, 因此最大支持18EB的分区大小。

parted分区工具使用

parted是另一款在linux下常用的分区软件，可支持创建2T以上磁盘分区，相对于fdisk，它的使用更加方便，同时提供了动态调整分区大小的功能，命令格式如下：

语法：parted [options] [device [command [options...]]...]

其中的命令选项说明如下：

- h : 显示帮助信息。
- i : 交互模式。
- s : 脚本模式。
- v : 显示parted的版本信息。

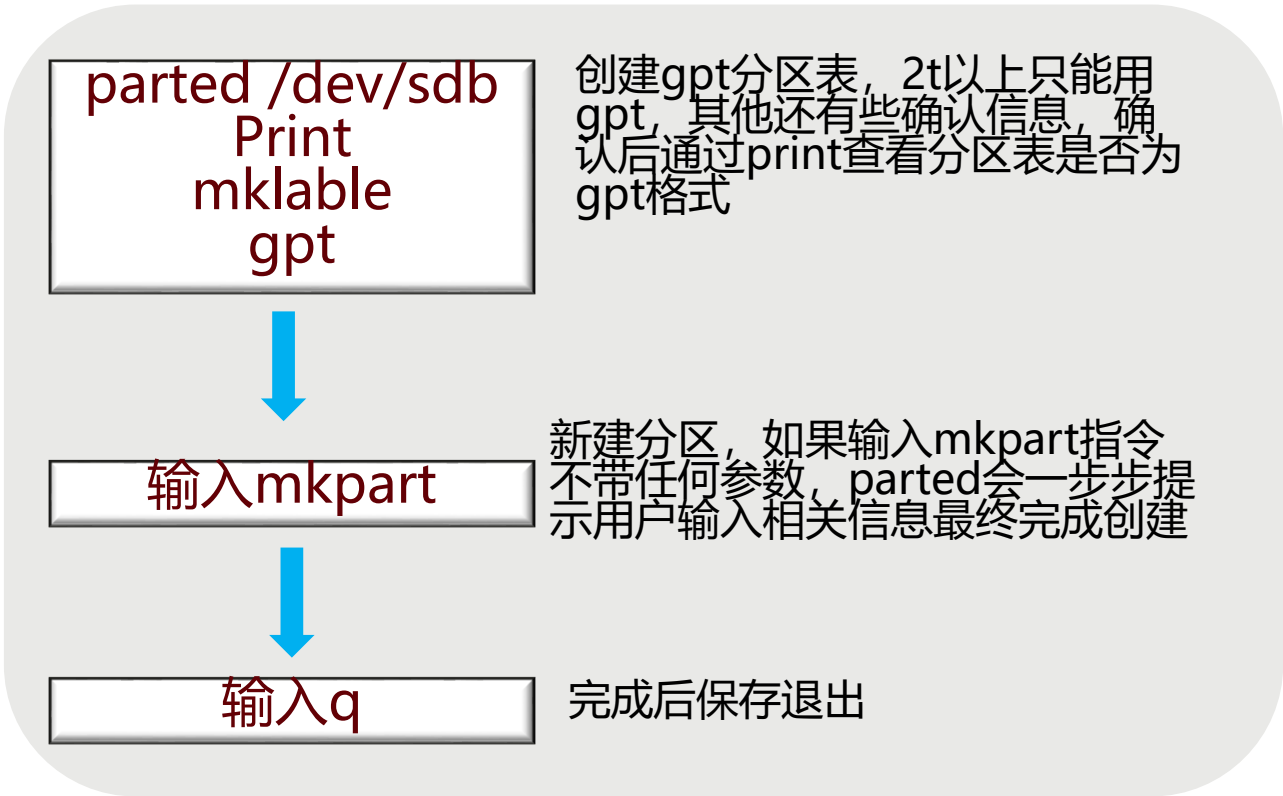
device : 磁盘设备名称，如/dev/sda。

command : parted指令，如果没有设置指令，则parted将会进入交互模式。

parted硬盘分区操作 - 交互式

选择好具体某块硬盘后，进入交互模式进行分区操作

如：parted /dev/sdb #交互式命令参数如表格所示。



命令	说明
align-check	检查分区N的类型(min opt)是否对
mklabel	创建新的磁盘标签 (分区表)
name	给指定的分区命名
print	打印分区表，或者分区
rescue	修复丢失的分区
resizepart	调整分区大小
rm	删除分区
select	选择要编辑的设备，默认只对指定的设备操作，这里可以改变指定的设备
disk_set	更改选定设备上的标志
disk_toggle	切换选定设备上的标志状态
quit	退出
set	更改分区的标记
toggle	设置或取消分区的标记
unit	设置默认的单位
version	显示版本信息

parted硬盘分区操作 - 非交互式

选择某块硬盘，采用非交互式即命令行式配置（命令中带参数）

如：parted /dev/sdb

磁盘格式变成gpt的格式



分区



格式化

- parted /dev/sdb mklabel gpt

- parted /dev/sdb mkpart
primary 0 1000

- mkfs -t ext4 /dev/sdb1

磁盘格式化

格式化是指对磁盘或磁盘中的分区进行初始化的一种操作，将分区格式化成不同的文件系统，供上层OS调用，这种操作通常会导致现有的磁盘或分区中所有的文件被清除。

操作前需要确认所有文件已经备份。

查看支持的文件类型

在操作系统下使用mkfs命令，tab补齐的方式查看当前操作系统支持的文件类型



磁盘格式化

在命令行下输入
`mkfs.ext4 /dev/sdb2`
意思就是格式化根下的sdb2分区，
格式化类型为.ext4



信息确认

格式化完成后，通过命令：
`ll`来查看磁盘信息，如 `ll /dev/sdb2`

fdisk硬盘分区操作

mkfs命令是make filesystem的缩写，用来在特定的分区建立Linux文件系统，命令格式如下：

语法：mkfs [-V] [-t fstype] [fs-options] filesys [blocks]

其中的命令选项说明如下：

device : 预备检查的硬盘分区，例如：/dev/sda1

-V : 详细显示模式

-t : 给定档案系统的型式，Linux 的预设值为 ext2

-c : 在制做档案系统前，检查该partition 是否有坏轨

-l bad_blocks_file : 将有坏轨的block资料加到
bad_blocks_file 里面

block : 给定 block 的大小

磁盘挂载

格式化完成以后，我们还不能使用磁盘，必须挂载以后才能用，原因如下：

Linux的宗旨是一切皆文件，要想使用磁盘，必须先建立一个联系，这个联系就是一个目录，建立联系的过程我们叫做挂载；

当我们访问sdb2底下的这个目录的时候，实际上我们访问的才是sdb2这个设备文件。这个目录相当于一个访问sdb2的入口，可以理解为一个接口，有了这个接口才可以访问这个磁盘。

挂载点目录

根目录下已有media 和mnt 这两个目录被叫做挂载点目录。
除此之外，我们也可以自己创建一个目录作为一个挂载点目录

临时挂载

使用命令：
`mount /dev/sda5 /test`
将/dev/sda5挂载test目录中，重启后失效

永久挂载

使用永久挂载，就意味着它开机自动挂载，使用vim编辑 /etc/fstab

fstab介绍

/etc/fstab文件的作用

该文件是用来存放文件系统的静态信息的文件，系统重新启动的时候，会自动地从这个文件读取信息，并且会自动将此文件中指定的文件系统挂载到指定的目录，这样我们只需要将磁盘的挂载信息写入这个文件中就不需要每次开机启动之后手动进行挂载了。

fstab文件格式

<file system>	<dir>	<type>	<options>	<dump>	<pass>
tmpfs	/tmp	tmpfs	nodev,nosuid	0	0
/dev/sda1	/	ext4	defaults,noatime	0	1
/dev/sda2	none	swap	defaults	0	0

fstab重要参数说明

字段	参数	功能
options	auto	在启动时或键入了 mount -a 命令时自动挂载
	ro	以只读模式挂载文件系统
	rw	以读写模式挂载文件系统
	user	允许任意用户挂载此文件系统
	nouser	只能被 root 挂载
	dev/nodev	解析/不解析文件系统上的块特殊设备
	noatime/nodir atime	不更新文件系统/目录上inode 访问记录，可以提升性能
	defaults	使用文件系统的默认挂载参数
	sync/async	I/O 同步/异步进行
	suid/nosuid	允许 /允许suid 操作和设定 sgid 位，这一参数通常用于一些特殊任务，使一般用户运行程序时临时提升权限
dump	0 /1	0 表示忽略， 1 则进行备份。大部分的用户是没有安装dump 的，对他们而言<dump> 应设为 0
pass	0, 1, 2	根目录应当获得最高的优先权1, 其它所有需要被检查的设备设置为2. 0 表示设备不会被 fsck 所检查

随堂测

- 1.磁盘分区有以下哪几种类型？（多选题）
 - A. 主分区
 - B. 扩展分区
 - C. 逻辑分区
 - D. 物理分区
- 2. `mount /dev/sda5 /test` 将/dev/sda5挂载test目录中，重启后不失效。（判断题）

目录

CONTENT

01 磁盘存储挂载与使用

02 逻辑卷管理

逻辑卷概念

LVM-逻辑卷管理是Linux环境中对磁盘分区进行管理的一种机制，是建立在硬盘和分区之上、文件系统之下的一个逻辑层，可提高磁盘分区管理的灵活性。

物理单元 (PE) : Physical Extents--具有唯一编号的PE是能被LVM寻址的最小单元，PE的大小可以指定，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的物理卷的PE的大小是一致的；

逻辑单元 (LE) : Logical Extents--是逻辑卷中可用于分配的最小存储单元，逻辑区域的大小取决于逻辑卷所在卷组中的物理区域的大小，在同一个卷组中，LE的大小和PE是相同的，通常一一对应；

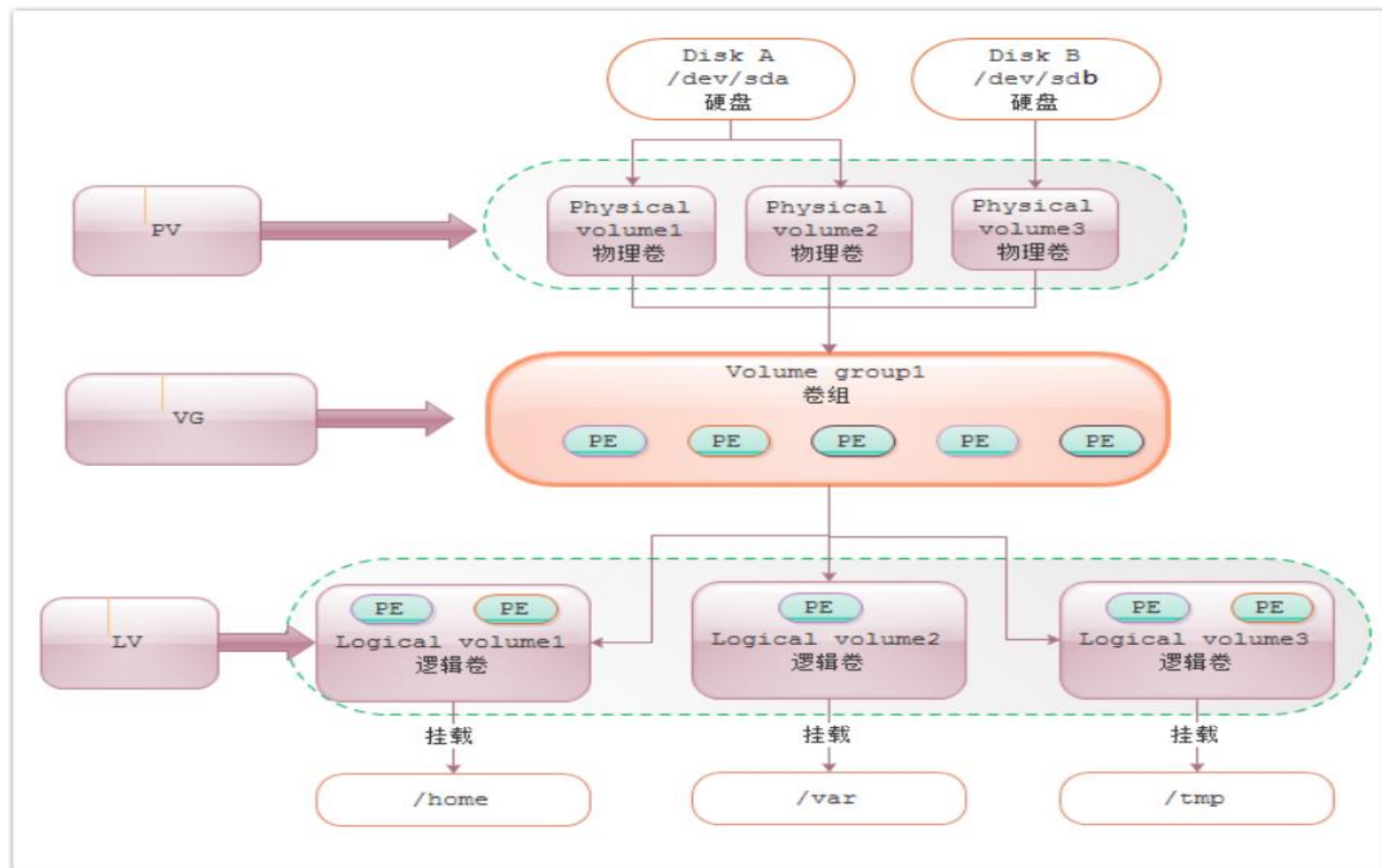
物理卷 (PV) : Physical Volume--底层真正提供容量，存放数据的设备，它可以是整个硬盘、硬盘上的分区等；

卷组 (VG) : Volume Group--建立在物理卷之上，它由一个或多个物理卷组成，即把物理卷整合起来提供容量分配，一个LVM系统中可以只有一个卷组，也可以包含多个卷组；

逻辑卷 (LV) : Logical Volume--逻辑卷建立在卷组之上，它是从卷组中“切出”的一块空间，它是最终用户使用的逻辑设备，逻辑卷创建之后，其大小可以伸缩。

逻辑卷原理及优点

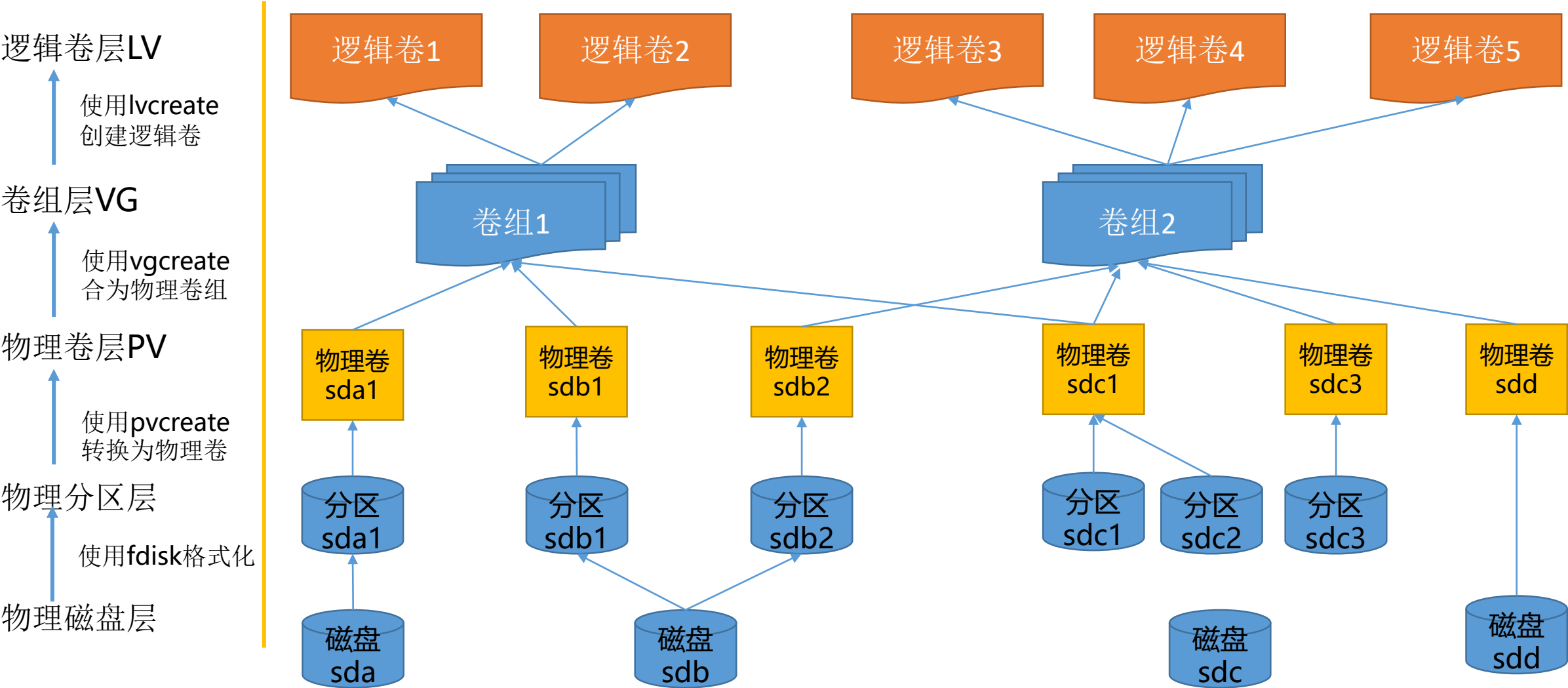
逻辑卷 (lv) 是将几个磁盘分区或者块设备(pv, pv可以位于不同的磁盘分区里, pv大小可以不一)组织起来形成一个大的扩展分区 (vg卷组, 一个vg至少要包含一个pv), 该扩展分区不能直接用, 需要将其划分成逻辑卷 (lv) 才能使用, lv可以格式化成不同的文件系统, 挂载后直接使用。



逻辑卷优点

- 灵活的容量
- 可伸缩的存储池
- 在线的数据再分配
- 方便的设备命名
- 磁盘条带化
- 卷镜像和卷快照

逻辑卷创建流程



创建物理卷PV

pvcree命令用于创建物理卷，可以使用物理磁盘或者磁盘分区创建。

命令： `pvdisplay [option] devname`

option：命令参数选项。常用的参数选项有：

-f：强制创建物理卷，不需要用户确认。

-u：指定设备的UUID。

-y：所有的问题都回答 “yes” 。

devname：指定要创建的物理卷对应的设备名称，如果需要批量创建，可以填写多个设备名称，中间以空格间隔。

示例1：将/dev/sdb、/dev/sdc创建为物理卷。

```
# pvcreate /dev/sdb /dev/sdc
```


查看物理卷PV

可在root权限通过pvdisplay命令查看物理卷的信息，包括：物理卷名称、所属的卷组、物理卷大小、PE大小、总PE数、可用PE数、已分配的PE数和UUID。

命令： `pvdisplay [option] devname`

其中：

option：命令参数选项。常用的参数选项有：

-s：以短格式输出。

-m：显示PE到LE的映射。

devname：指定要查看的物理卷对应的设备名称。如果不指定物理卷名称，则显示所有物理卷的信息。

示例：显示物理卷/dev/sdb的基本信息。

```
# pvdisplay /dev/sdb
```

也可以使用pvscan、pvs查看物理卷概要信息

修改物理卷属性

可在root权限下通过pvchange命令修改物理卷的属性。

命令： `pvchange [option] pvname ...`

其中：

option：命令参数选项。常用的参数选项有：

-u：生成新的UUID。

-x：是否允许分配PE”。

pvname：指定要修改属性的物理卷对应的设备名称，如果需要批量修改，可以填写多个设备名称，中间以空格间隔。

示例：禁止分配/dev/sdb物理卷上的PE

```
# pvchange -x n /dev/sdb
```

删除物理卷

可在root权限下通过pvremove命令删除物理卷。

命令： `pvremove [option] pvname ...`

其中：

option：命令参数选项。常用的参数选项有：

-f：强制删除物理卷，不需要用户确认。

-y：所有的问题都回答“yes”。

pvname：指定要删除的物理卷对应的设备名称，如果需要批量删除，可以填写多个设备名称，中间以空格间隔。

示例：删除物理卷/dev/sdb

```
# pvremove /dev/sdb
```

管理卷组-创建卷组

vgcreate命令用于创建LVM卷组。卷组将多个物理卷组织成一个整体，屏蔽了底层物理卷细节。在卷组上创建逻辑卷时不用考虑具体的物理卷信息。

命令： `vgcreate [option] vgname pvname ...`

其中的option选项说明如下：

- l：卷组上允许创建的最大逻辑卷数。
- p：卷组中允许添加的最大物理卷数。
- s：卷组上的物理卷的PE大小。

vgname：要创建的卷组名称。

pvname：要加入到卷组中的物理卷名称

示例：创建卷组 vg1，并且将物理卷/dev/sdb和/dev/sdc添加到卷组中。

```
# vgcreate vg1 /dev/sdb /dev/sdc
```

管理卷组-查看卷组

可在root权限下通过vgdisplay命令查看卷组的信息。

命令： `vgdisplay [option] [vgname]`

其中的option选项说明如下：

-s：以短格式输出。

-A：仅显示活动卷组的属性。

vgname：指定要查看的卷组名称。如果不指定卷组名称，则显示所有卷组的信息。

示例：显示卷组vg1的基本信息

```
# vgdisplay vg1
```

管理卷组-修改卷组属性

可在root权限下通过vgchange命令修改卷组的属性。

命令： `vgchange [option] vgname`

其中的option选项说明如下：

a: 设置卷组的活动状态。

vgname: 指定要修改属性的卷组名称。

示例：将卷组vg1状态修改为活动。如果修改为非活动，则是-a n

```
# vgchange -a y vg1
```

管理卷组-扩展卷组

可在root权限下通过vgextend命令动态扩展卷组。它通过向卷组中添加物理卷来增加卷组的容量。

命令： `vgextend [option] vgname pvname ...`

其中的option选项说明如下：

-d：调试模式。

-t：仅测试。

vgname：要扩展容量的卷组名称。

pvname：要加入到卷组中的物理卷名称。

示例：向卷组vg1中添加物理卷/dev/sdb。

```
# vgextend vg1 /dev/sdb
```

管理卷组-收缩卷组

可在root权限下通过vgreduce命令删除卷组中的物理卷来减少卷组容量。不能删除卷组中剩余的最后一个物理卷。

命令： `vgreduce [option] vgname pvname ...`

其中的option选项说明如下：

- a：如果命令行中没有指定要删除的物理卷，则删除所有的空物理卷。
- removemissing：删除卷组中丢失的物理卷，使卷组恢复正常状态。

vgname：要收缩容量的卷组名称。
pvname：要从卷组中删除的物理卷名称。

示例：从卷组vg1中移除物理卷/dev/sdb2。

```
# vgreduce vg1 /dev/sdb2
```


管理卷组-删除卷组

可在root权限下通过vgremove命令删除卷组。

命令： `vgremove [option] vgname`

其中的option选项说明如下：

-f：强制删除卷组，不需要用户确认。

vgname：指定要删除的卷组名称。

示例：删除卷组vg1

```
# vgrremove vg1
```

管理逻辑卷-创建逻辑卷

可在root权限下通过lvcreate命令创建逻辑卷。

命令： `lvcreate [option] vgname`

其中的option选项说明如下：

- L：指定逻辑卷的大小，单位为“kKmMgGtT”字节。
- l：指定逻辑卷的大小（LE数）。
- n：指定要创建的逻辑卷名称。
- s：创建快照。

vgname：要创建逻辑卷的卷组名称。

示例1：在卷组vg1中创建10G大小的逻辑卷。

```
# lvcreate -L 10G vg1
```

示例2：在卷组vg1中创建200M的逻辑卷，并命名为lv1。

```
# lvcreate -L 200M -n lv1 vg1
```

管理逻辑卷-查看逻辑卷

可在root权限下通过lvdisplay命令查看逻辑卷的信息，包括逻辑卷空间大小、读写状态和快照信息等属性。

命令： `lvdisplay [option] [lvname]`

其中的option选项说明如下：

-v：显示LE到PE的映射

lvname：指定要显示属性的逻辑卷对应的设备文件。如果省略，则显示所有的逻辑卷属性。

说明：逻辑卷对应的设备文件保存在卷组目录下，例如：在卷组vg1上创建一个逻辑卷lv1，则此逻辑卷对应的设备文件为/dev/vg1/lv1。

示例：显示逻辑卷lv1的基本信息。

```
# lvdisplay /dev/vg1/lv1
```

管理逻辑卷-调整逻辑卷大小

可在root权限下通过lvresize命令调整LVM逻辑卷的空间大小，可以增大空间和缩小空间。使用lvresize命令调整逻辑卷空间大小和缩小空间时需要谨慎，因为有可能导致数据丢失。

命令： `lvresize [option] vgname`

其中的option选项说明如下：

- L: 指定逻辑卷的大小，单位为“kKmMgGtT” 字节。
- l: 指定逻辑卷的大小（LE数）。
- f: 强制调整逻辑卷大小，不需要用户确认

lvname: 指定要调整的逻辑卷名称。

示例1：为逻辑卷/dev/vg1/lv1增加200M空间

```
# lvresize -L +200 /dev/vg1/lv1
```

示例2：为逻辑卷/dev/vg1/lv1减少200M空间。

```
# lvresize -L -200 /dev/vg1/lv1
```

管理逻辑卷-扩展逻辑卷

可在root权限下通过lvextend命令动态在线扩展逻辑卷的空间大小，而不中断应用程序对逻辑卷的访问。

命令： `lvextend [option] lvname`

其中的option选项说明如下：

- L：指定逻辑卷的大小，单位为“kKmMgGtT”字节。
- l：指定逻辑卷的大小（LE数）。
- f：强制调整逻辑卷大小，不需要用户确认。

lvname：指定要扩展空间的逻辑卷的设备文件。

示例：为逻辑卷/dev/vg1/lv1增加100M空间。

```
# lvextend -L +100M /dev/vg1/lv1
```

逻辑卷扩容

无需卸载
在线扩容

01

扩容前先查看确认是否有足够vg空间: `vgs`

02

使用命令扩容: `lvextend -L +SIZE lv_device`

03

调整文件系统的大小: `resize2fs device lv_device`

管理逻辑卷-收缩逻辑卷

可在root权限下通过lvreduce命令减少逻辑卷占用的空间大小。使用lvreduce命令收缩逻辑卷的空间大小有可能会删除逻辑卷上已有的数据，所以在操作前必须进行确认。

命令： `lvreduce [option] lvname`

其中的option选项说明如下：

- L：指定逻辑卷的大小，单位为“kKmMgGtT”字节。
- l：指定逻辑卷的大小（LE数）。
- f：强制调整逻辑卷大小，不需要用户确认。

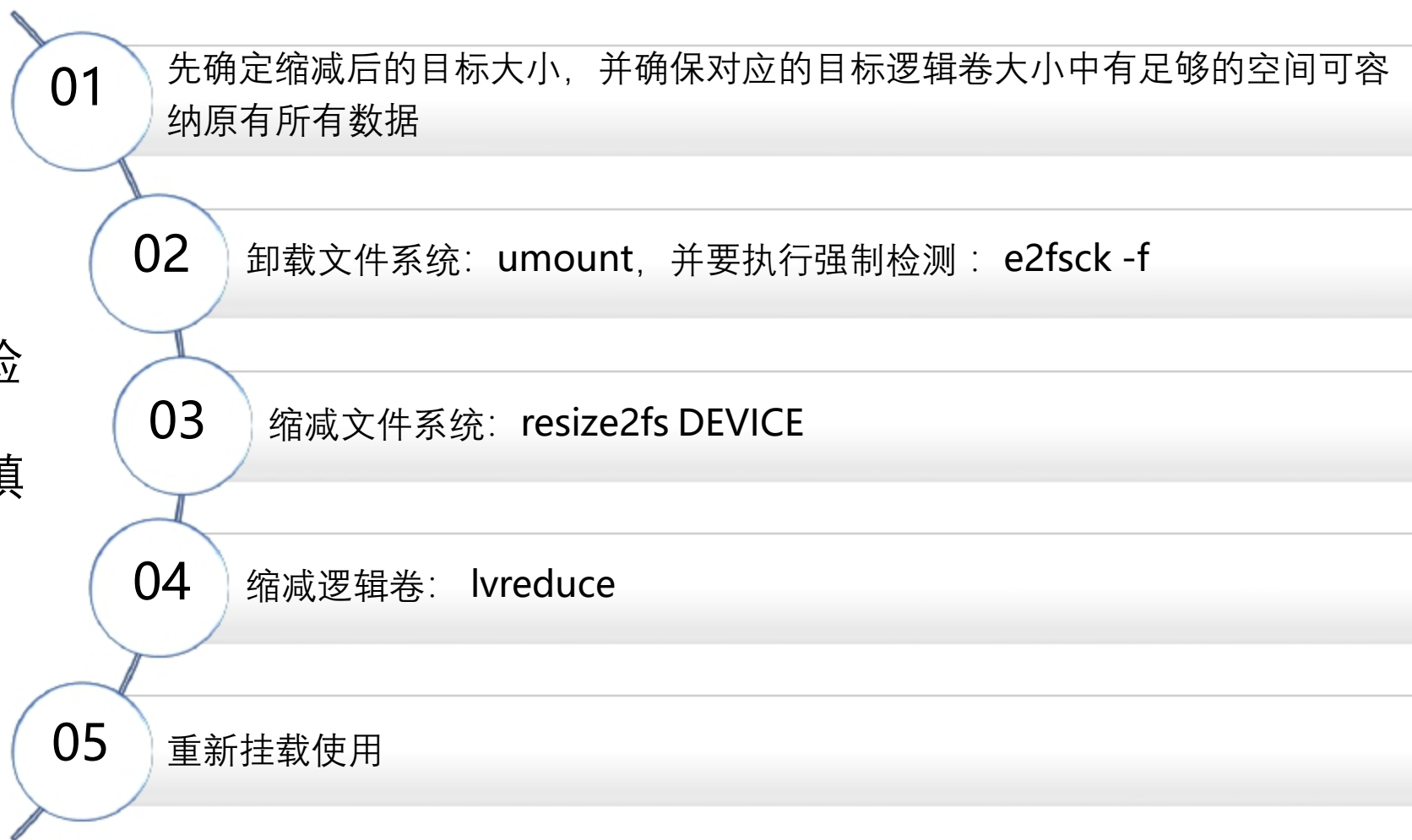
lvname：指定要扩展空间的逻辑卷的设备文件。

示例：将逻辑卷/dev/vg1/lv1的空间减少100M

```
# lvreduce -L -100M /dev/vg1/lv1
```

逻辑卷缩容

缩容有风险
操作需谨慎



管理逻辑卷-删除逻辑卷

可在root权限下通过lvremove命令删除逻辑卷。如果逻辑卷已经使用mount命令加载，则不能使用lvremove命令删除。必须使用umount命令卸载后，逻辑卷方可被删除。

命令： `lvremove [option] vgname`

其中的option选项说明如下：

-f：强制删除逻辑卷，不需要用户确认。

vgname：指定要删除的逻辑卷。

示例：删除逻辑卷/dev/vg1/lv1。

```
# lvremove /dev/vg1/lv1
```

逻辑卷容量变更

lvresize指令用于调整逻辑卷空间大小，可以增大空间和缩小空间，实际上包含了lvextend和lvreduce的功能，因此扩容及缩容步骤相同。

语法：lvresize [选项] 逻辑卷名称

其中的命令选项说明如下：

- L：指定逻辑卷的大小，单位为“kKmMgGtT”字节。
- l：指定逻辑卷的大小（LE数）。

其中的参数说明如下：

逻辑卷：指定要创建的逻辑卷名称。

如使用lvresize指令增加容量

```
[root@openEuler ~]# lvresize -L +200M /dev/vg1000/lvol0    #将逻辑卷空间增加200M
```

随堂测

- 1. 将分区/dev/hdb6格式化的命令是哪个？（单选题）
 - A. `mkfs -t ext4 /dev/hdb6`
 - B. `format -t ext4 /dev/hdb6`
 - C. `format /dev/hdb6`
 - D. `makefile -t ext4 /dev/hdb6`
- 2. 逻辑卷缩减有风险，所以要卸载并强行检测文件系统。（判断题）

Thank you