

管理文件系统及存储



前言

- 本章主要讲述文件系统、磁盘存储、逻辑卷存储的基础概念以及如何管理、使用文件系统和存储，以及对一些常用操作命令的介绍。

目标

- 学完本课程后，您将能够：
 - 了解文件系统及存储的基本概念
 - 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

1. 文件系统基础概念

- 文件系统
 - openEuler文件系统
 - 系统交换分区

2. 磁盘存储挂载与使用

3. 逻辑卷管理

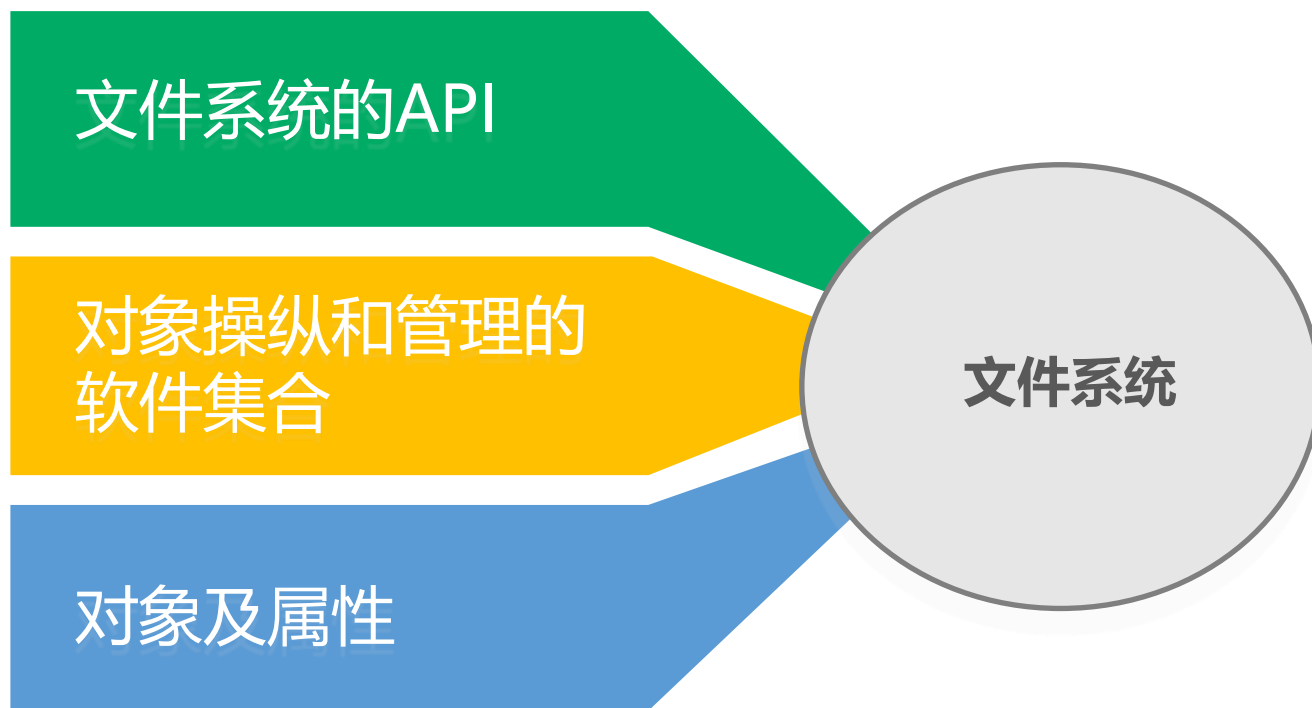
文件系统概述

- 文件系统是操作系统用于明确存储设备或分区上的文件的方法和数据结构；即在存储设备上组织文件的方法。操作系统中负责管理和存储文件信息的软件机构称为文件管理系统，简称文件系统。

提供用户人性化接口

磁盘管理软件，类似于中间件

底层存储硬件，如磁盘或分区



文件系统类型及使用场景

常见文件类型	使用场景
FAT	windows9X系统使用的文件系统，包括FAT16，FAT32
NTFS	NTFS文件系统是一个基于安全性的文件系统，是Windows NT所采用的独特的文件系统结构，Win 2000采用了更新版本的NTFS文件系统NTFS 5.0
NFS	网络文件系统，用于在UNIX系统间通过网络进行文件共享
RAW	RAW文件系统是一种磁盘未经处理或者未经格式化产生的文件系统
Ext	GNU/Linux 系统中标准的文件系统，其特点为存取文件的性能极好，对于中小型的文件更显示出优势，包括Ext2，Ext3，Ext4
XFS	一种高性能的日志文件系统，最早于1993年，由Silicon Graphics为他们的IRIX操作系统而开发，之后被移植到Linux 内核上，特别擅长处理大文件，同时提供平滑的数据传输。

目录

1. 文件系统基础概念

- 文件系统
 - openEuler文件系统
- 系统交换分区

2. 磁盘存储挂载与使用

3. 逻辑卷管理

openEuler文件系统

- openEuler内核源于Linux，Linux内核支持十多种不同类型的文件系统，比如Btrfs、JFS、ReiserFS、ext、ext2、ext3、ext4、ISO9660、XFS、Minix、MSDOS、UMSDOS、VFAT、NTFS、HPFS、NFS、SMB、SysV、PROC等，表格中列举常用几种并做说明。
- openEuler默认文件系统为ext4。

常用文件系统	说明
Ext	专门为Linux设计的文件系统类型，目前最新版本ext4
XFS	一种高性能的日志文件系统，最早于1993年，由Silicon Graphics为他们的IRIX操作系统而开发，之后被移植到Linux 内核上，特别擅长处理大文件，同时提供平滑的数据传输。
vfat	vfat是Linux对DOS,Windows系统下的FAT(包括fat16和Fat32)文件系统的一个统称
NFS	网络文件系统，用于在UNIX系统间通过网络进行文件共享
ISO 9600	该文件系统中光盘所使用的标准文件系统，Linux对该文件系统也有很好的支持，不仅能读取光盘和光盘ISO映像文件，而且还支持在Linux环境中刻录光盘

目录

1. 文件系统基础概念

- 文件系统
- openEuler文件系统
- 系统交换分区

2. 磁盘存储挂载与使用

3. 逻辑卷管理

系统交换空间

- Linux系统交换空间（swap）就是磁盘上的一块区域，可以是一个分区，也可以是一个文件，简单的说就是当物理内存资源紧张时，将内存中不常访问的资源保存到预先设定的硬盘上的交换空间，来释放该资源占用的内存，这样系统就有更多的物理内存为各个进程服务，而当系统需要访问swap上存储的内容时，再将swap上的数据加载到内存中。
- 物理内存和交换空间的和就是系统可提供的虚拟内存的总量。
- 为什么需要swap分区？
 - 增加系统可用内存空间—当物理内存不够用时，增加swap分区比增加物理内存更经济；
 - 提高系统整体性能—将不常用数据移动到swap后，系统会有更多内存用于缓存，加快系统I/O；
 - 许多linux发行版（如ubuntu）的休眠功能依赖与swap分区—当系统休眠时，会将内存数据保存到swap分区，下次启动时再加载到内存。

swap分区配置

- Linux有两种形式的交换空间：交换分区和交换文件。交换分区就是一个独立的硬盘，没有文件或内容，即swap分区；交换文件是文件系统中的特殊文件，独立于系统 and 数据文件之外。
 - 创建swap分区：fdisk创建分区，mkswap创建swap分区，swapon启用swap分区；
 - 创建swap文件：创建文件，mkswap格式化文件，swapon启用swap文件。
- 推荐的swap大小配置

RAM 大小	推荐的交换空间
≤ 2GB	2X RAM
2GB – 8GB	= RAM
>8GB	8GB

目录

1. 文件系统基础概念

2. 磁盘存储挂载与使用

- 磁盘基础

- 磁盘分区

- 格式化与挂载

3. 逻辑卷管理

磁盘类型

硬盘
材质

HDD:
机械硬盘

SSD:
固态硬盘

接口
类型

- IDE
- SCSI
- SATA
- SAS
- FC

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: 串行ATA)	SATA的出现将ATA和IDE区分开来，而IDE则属于Parallel-ATA(并行ATA)。所以，一般来说，IDE称为并口，SATA称为串口。
SCSI (Small Computer System Interface: 小型计算机系统专用接口)	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, 42949672960 bytes, 83886080 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
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	83884031	81375232	38.8G	Linux LVM

```
Disk /dev/mapper/openEuler-root: 34.82 GiB, 37367054336 bytes, 72982528 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 /dev/mapper/openEuler-swap: 4 GiB, 4294967296 bytes, 8388608 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
```

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
```

目录

1. 文件系统基础概念

2. 磁盘存储挂载与使用

- 磁盘基础

- 磁盘分区

- 格式化与挂载

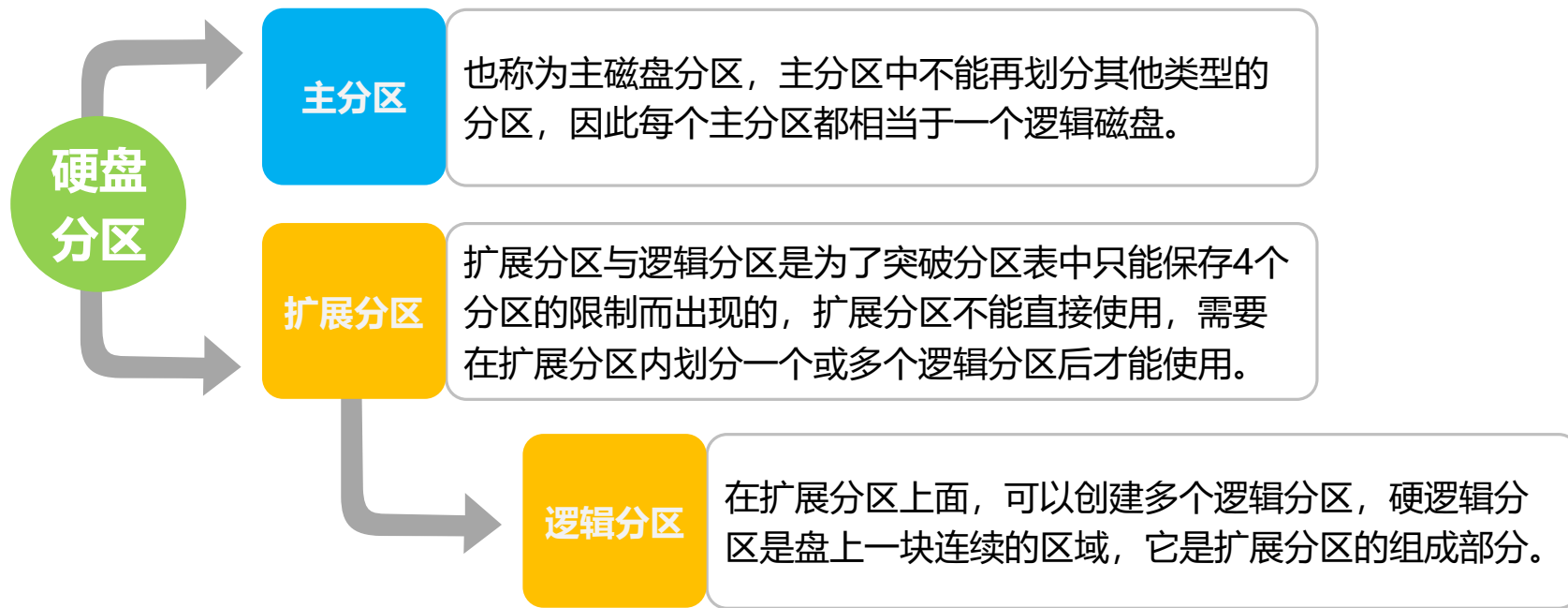
3. 逻辑卷管理

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。

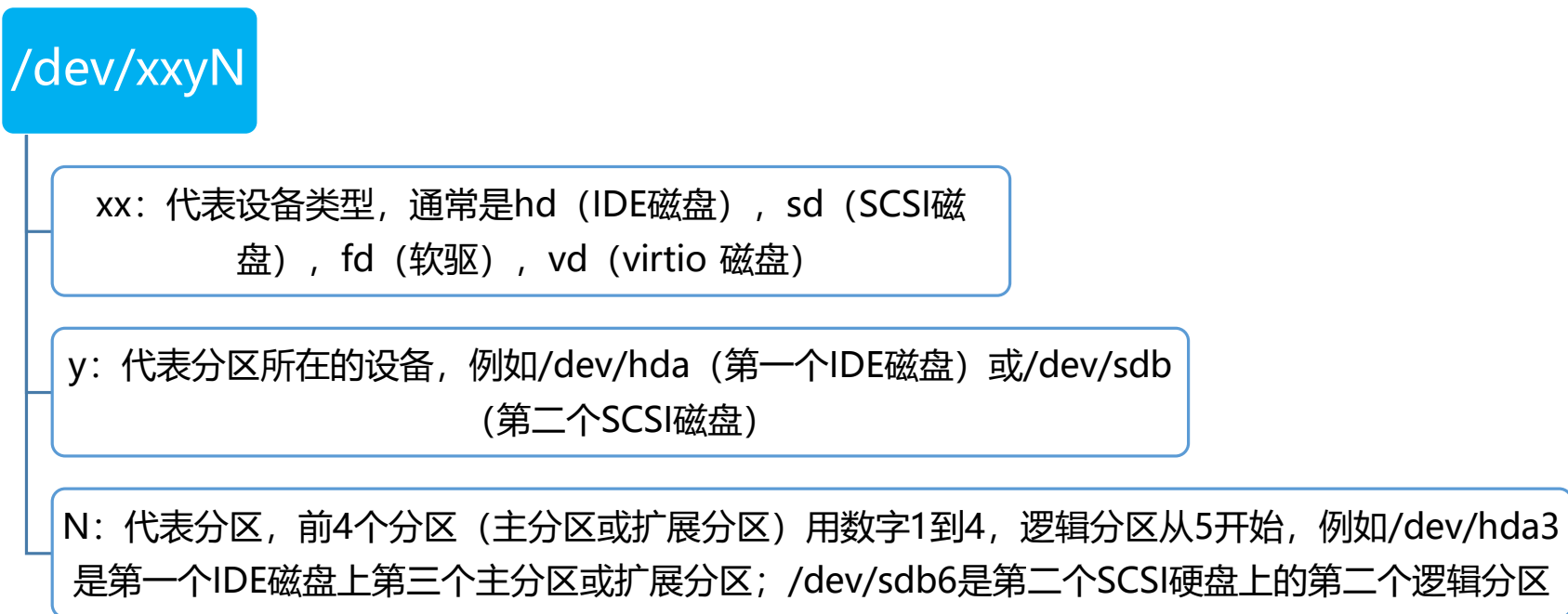
磁盘分区类型

- 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：



- 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

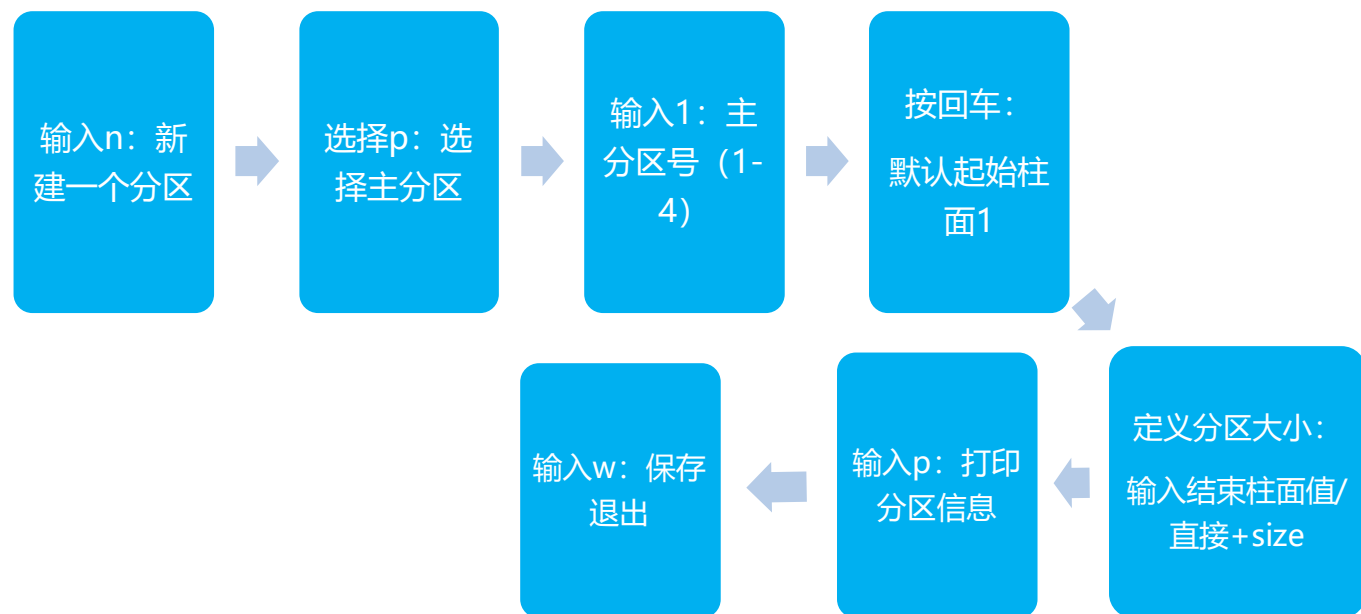
- 主启动记录(Master Boot Record) 分区方案指定了在运行BIOS固件的系统上应如何对磁盘进行分区，它是存在于驱动器开始部分的一个特殊的启动扇区。
- 如果是SCSI接口硬盘则最多只能有15（其中扩展分区不能直接使用所以不计算）个分区，其中主分区最多4个，逻辑分区最多12个。IDE接口硬盘最多只能有63（其中扩展分区不能直接使用所以不计算）个分区，其中主分区最多4个，逻辑分区最多60个。
- 由于分区大小数据以32位值存储,使用MBR方案分区时,最大磁盘和分区大小限制为2TB。

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

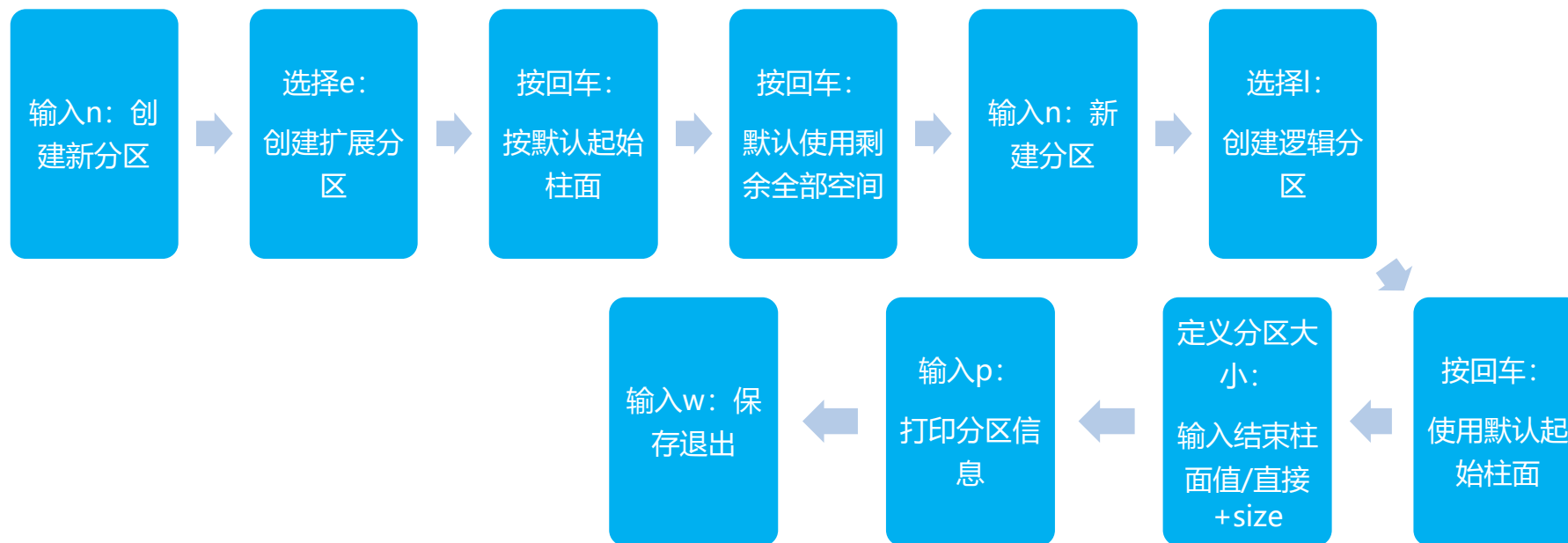
- 选择好具体某块硬盘后，进入交互模式进行分区操作
- 如：fdisk /dev/sdb #交互式命令参数如表格所示
- 创建主分区流程：



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

fdisk硬盘分区操作

- 创建扩展分区流程：



- 注：扩展分区创建完成后不能直接使用，必须要创建逻辑分区

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

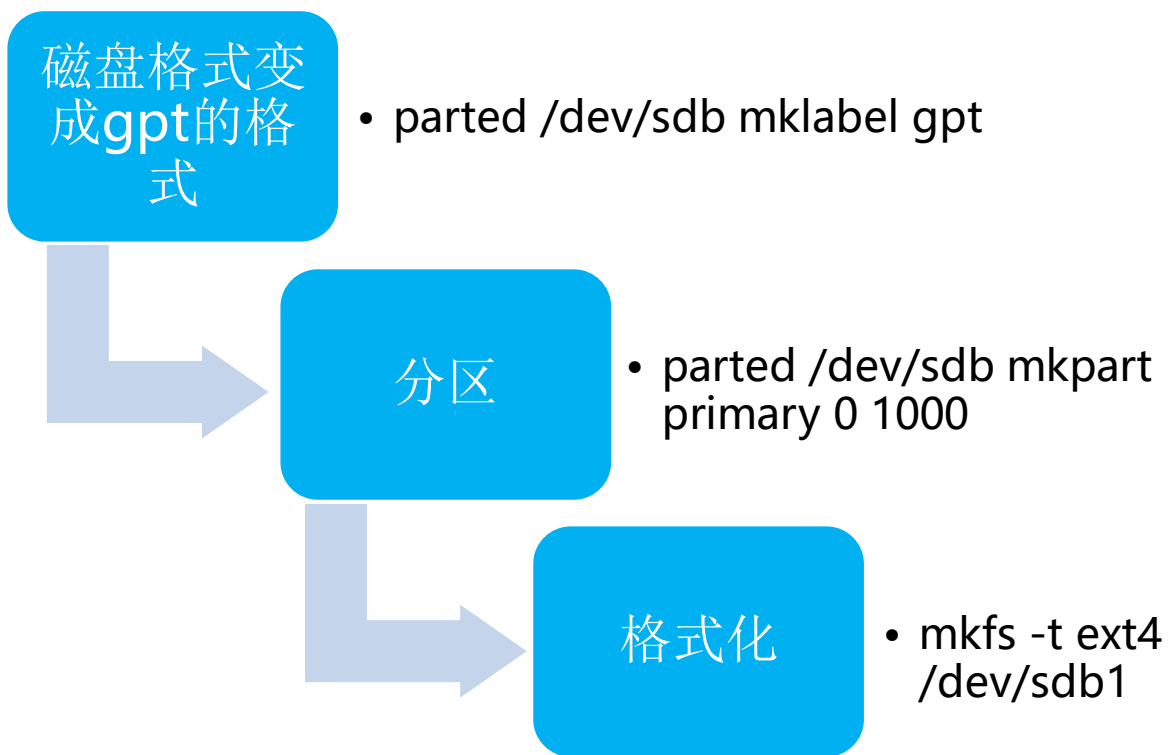
- 选择好具体某块硬盘后，进入交互模式进行分区操作。
- 如：parted /dev/sdb #交互式命令参数如表格所示



命令	说明
align-check	检查分区N的类型(min opt)是否对齐
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
- 如：parted /dev/sdb



目录

1. 文件系统基础概念

2. 磁盘存储挂载与使用

- 磁盘基础

- 磁盘分区

- 格式化与挂载

3. 逻辑卷管理

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，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。



前面已讲过linux支持的文件类型，可通过mkfs. 然后tab键补全，查看可支持的文件类型有哪些，选择需要的类型进行格式化



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



格式化完成后，通过命令：ll来查看磁盘信息，
如 ll /dev/sdb2

磁盘格式化命令 - mkfs

- 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，意思就是说将/dev/sda5挂载到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
 - 用来存放文件系统的静态信息的文件，系统启动的时候，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/nodiratime	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

目录

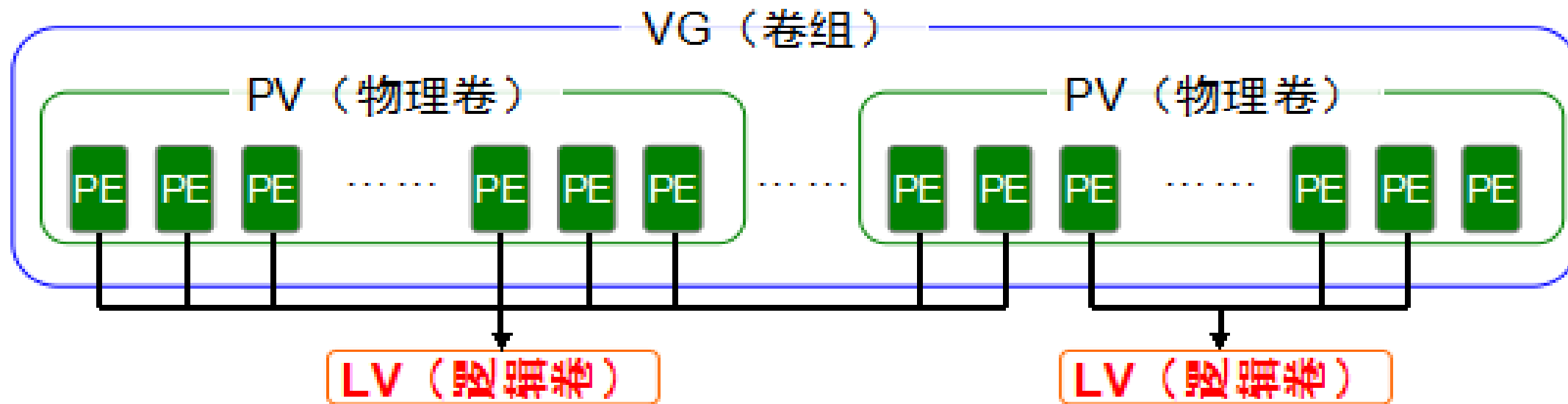
1. 文件系统基础概念
2. 磁盘存储挂载与使用
- 3. 逻辑卷管理**
 - 逻辑卷基础
 - 管理逻辑卷
 - 动态调整逻辑卷

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

- 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。



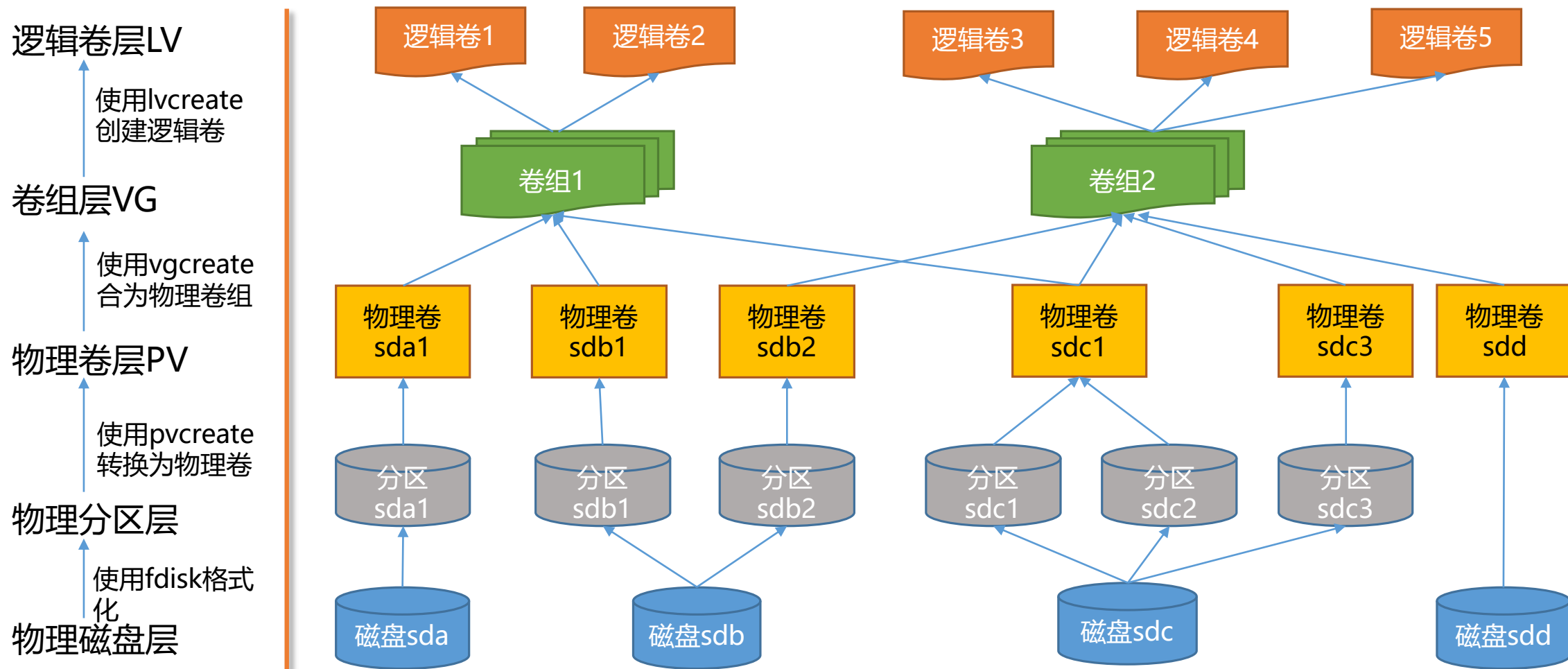
逻辑卷优点



目录

1. 文件系统基础概念
2. 磁盘存储挂载与使用
- 3. 逻辑卷管理**
 - 逻辑卷基础
 - 管理逻辑卷
 - 动态调整逻辑卷

逻辑卷创建流程



逻辑卷管理 - pv的使用

- pvcreate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。
- 语法：pvcreate [选项] 设备文件名
 - 其中的命令选项说明如下：
 - - 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。

▫ 其中的参数说明如下：

指定要创建的物理卷对应的设备文件名

注：使用分区创建时，磁盘分区后的id为83，需要将id改为8e，才能创建物理卷，可用fdisk修改id

逻辑卷管理 - pv的使用举例

- 将磁盘hda的6-9分区创建成pv，注意大括号的使用。

- 在命令行中输入下面的命令：

```
[root@openEuler ~]#pvcreate /dev/hda{6,7,8,9}
```

- 使用pvdisk、pvscan、pvs查看物理卷信息。

- 如使用pvs查看：

```
[root@openEuler ~]#pvs #查看物理卷概要信息
```

- 输出信息说明

PV	VG	fmt	Attr	PSize	PFree
/dev/sdb1	vg1000	lvm2	--	100.00M	100.00M
/dev/sdb2		lvm2	--	101.98M	101.98M

逻辑卷管理 - vg的使用

- 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。
- 语法：vgcreate [选项] 卷组名 物理卷列表
 - 其中的命令选项说明如下：
 - - 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。
 - 其中的参数说明如下：
 - 卷组名：要创建的卷组名称。
 - 物理卷列表：要加入到卷组中的物理卷列表。

逻辑卷管理 - vg的使用举例

- 使用vgcreate命令创建卷组“vg1000”，并且将物理卷/dev/sdb1和/dev/sdb2添加到卷组中。
 - 在命令行中输入下面的命令：

```
[root@openEuler ~]#vgcreate vg1000 /dev/sdb1  
/dev/sdb2
```

- 使用vgdisplay、vgscan查看卷组信息。
 - 如使用vgdisplay:

```
[root@openEuler ~]#vgdisplay vg1000
```

注：不指定vg1000卷组则显示所有卷组信息

逻辑卷管理 - lv的使用

- lvcreate命令用于创建LVM的逻辑卷，逻辑卷是创建在卷组之上的。
- 语法：lvcreate [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。

注：逻辑卷创建完成后，同样需要格式化，挂载后才能使用，方式同2.4章节介绍的方法一致，即用mkfs格式化创建文件系统，mount挂载至目录。

逻辑卷管理 - lv的使用举例

- 使用lvcreate命令在卷组“vg1000” 上创建一个200MB的逻辑卷。
 - 在命令行中输入下面的命令：

```
[root@openEuler ~]#lvcreate -L 200M vg1000
```

- 使用lvdisplay、lvscan查看卷组信息。
 - 如使用lvscan：

```
[root@openEuler ~]#lvscan #扫描所有逻辑卷
```

目录

1. 文件系统基础概念
2. 磁盘存储挂载与使用
- 3. 逻辑卷管理**
 - 逻辑卷基础
 - 管理逻辑卷
 - 动态调整逻辑卷

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

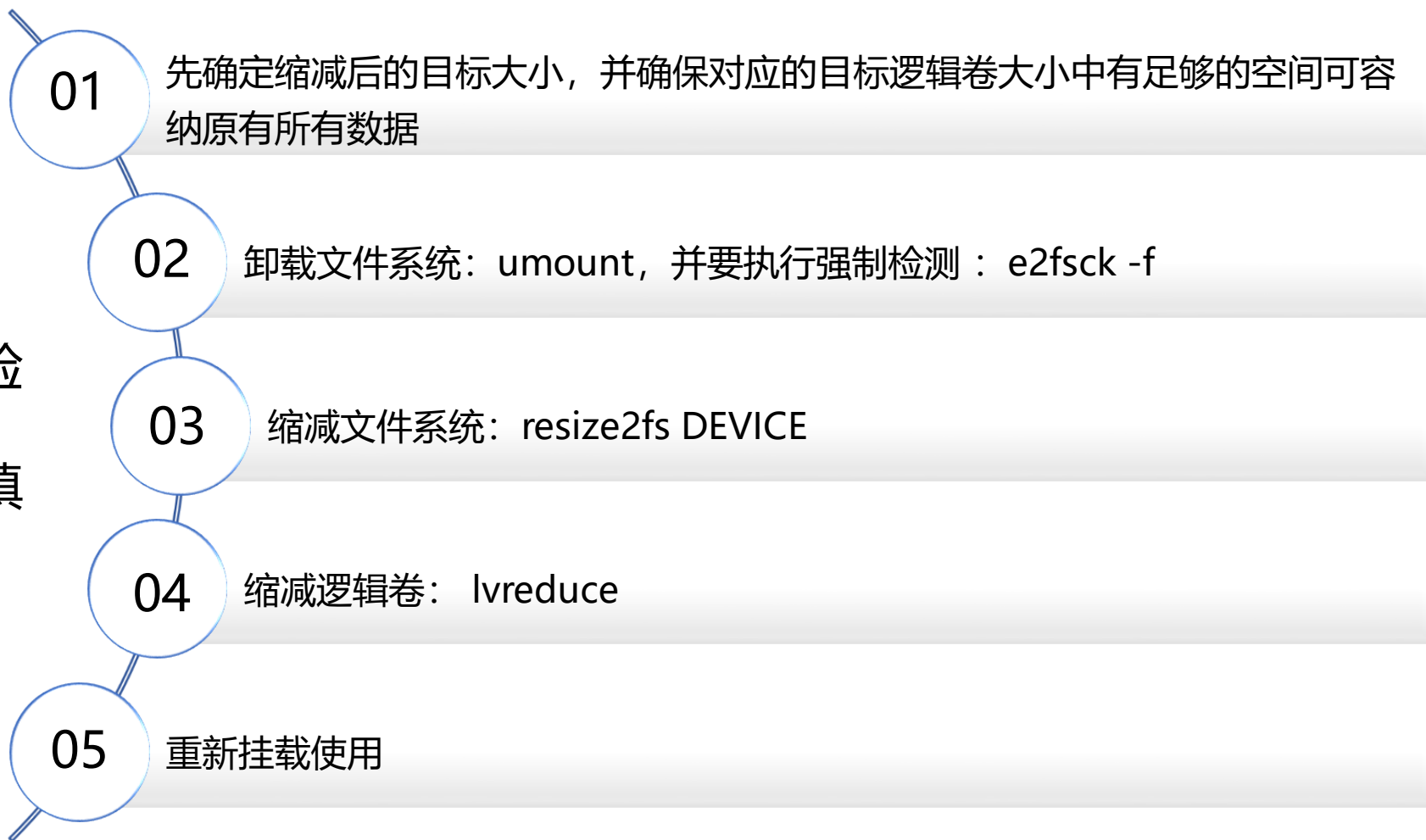
使用命令扩容: `lvextend -L +SIZE lv_device`

03

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

逻辑卷缩容

缩容有风险
操作需谨慎



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
```

思考题

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）

本章总结

- 本章内容主要介绍文件系统的概念，并对磁盘的挂载使用和逻辑卷的管理进行详细讲解。

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and
organization for a fully connected,
intelligent world.

**Copyright©2020 Huawei Technologies Co., Ltd.
All Rights Reserved.**

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

