一、环境准备

开启CentOS7虚拟机

构建Yum仓库

[root@localhost ~]# mkdir /centos7

[root@localhost ~]# mount /dev/cdrom /centos7

[root@localhost ~]# ls /centos7

[root@localhost ~]# rm -rf /etc/yum.repos.d/\*

[root@localhost ~]# vim /etc/yum.repos.d/mydvd.repo

[nsd]

name=xixi

baseurl=file:///centos7

enabled=1

gpgcheck=0

[root@localhost ~]# yum -y install xorg-x11-apps

二、制作交换空间（虚拟内存）

采用**文件**所占用空间，进行制作交换空间

1.生成一个较大的2G文件

* + dd if=源设备 of=目标设备 bs=块大小 count=块数

/dev/zero:永远都在产生数据

]# dd if=/dev/zero of=/opt/swap.txt bs=1M count=2048

]# du -sh /opt/swap.txt

]# mkswap /opt/swap.txt #格式化交换文件系统

]# swapon /opt/swap.txt #启用

swapon: /opt/swap.txt：不安全的权限 0644，建议使用 0600。

]# swapon #查看交换空间组成的成员

三、添加一块新的80G硬盘

[root@localhost ~]# poweroff



[root@localhost ~]# lsblk

80G硬盘进行（MBR分区模式）规划分区

划分2个10G的主分区；1个12G的主分区;2个20G的逻辑分区

[root@localhost ~]# fdisk /dev/sdb

**n** 创建主分区--->**回车**--->**回车**--->**回车**--->在last结束时 **+10G**

**n** 创建主分区--->**回车**--->**回车**--->**回车**--->在last结束时 **+10G**

**n** 创建主分区--->**回车**--->**回车**--->**回车**--->在last结束时 **+12G**

**p** 查看分区表

**n** 创建扩展分区 --->**回车**--->起始**回车**--->结束**回车** **将所有剩余空间给扩展分区**

**p** 查看分区表

**n** 创建逻辑分区----->起始**回车**------>结束**+20G**

**n** 创建逻辑分区----->起始**回车**------>结束**+20G**

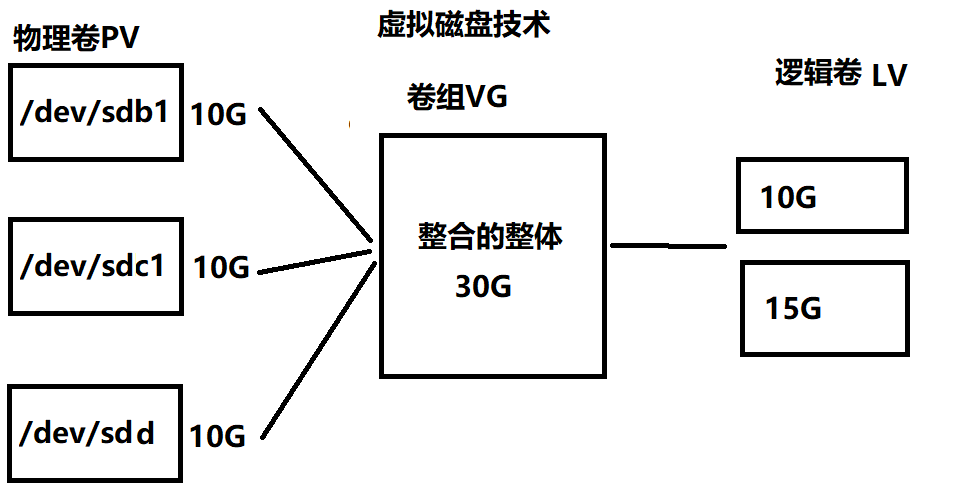
**p** 查看分区表

**w** 保存并退出

[root@localhost ~]# lsblk

四、逻辑卷

作用：1.整合分散的空间 2.空间支持扩大



逻辑卷制作过程：将众多的物理卷（PV）组建成卷组（VG），再从卷组中划分出逻辑卷（LV）

**LVM管理工具集**

|  |  |  |  |
| --- | --- | --- | --- |
| **功能** | **物理卷管理** | **卷组管理** | **逻辑卷管理** |
| Scan 扫描 | pvscan | vgscan | lvscan |
| Create 创建 | pvcreate | vgcreate | lvcreate |
| Display 显示 | pvdisplay | vgdisplay | lvdisplay |
| Remove 删除 | pvremove | vgremove | lvremove |
| Extend 扩展 | / | vgextend | lvextend |

五、制作逻辑卷

* **建立卷组（VG）**

格式：vgcreate 卷组名 设备路径…….

Successfully:成功

[root@localhost ~]# vgcreate systemvg /dev/sdb[1-2]

[root@localhost ~]# pvs #查看系统所有物理卷信息

[root@localhost ~]# vgs #查看系统卷组信息

* **建立逻辑卷（LV）**

格式: lvcreate -L 大小G -n 逻辑卷名字 卷组名

[root@localhost ~]# lvcreate -L 16G -n vo systemvg

[root@localhost ~]# vgs #查看卷组信息

[root@localhost ~]# lvs #查看逻辑卷信息

* **使用逻辑卷（LV）**

]# ls /dev/systemvg/vo

]# ls -l /dev/systemvg/vo

]# mkfs.xfs /dev/systemvg/vo #格式化xfs文件系统

]# blkid /dev/systemvg/vo #查看文件系统类型

]# vim /etc/fstab

/dev/systemvg/vo /mylv xfs defaults 0 0

]# mkdir /mylv

]# mount -a #检测fstab文件内容书写是否正确

]# df -h #查看查看正在挂载使用的设备

六、逻辑卷的扩展

* **卷组有足够的剩余空间**

1.扩展逻辑卷的空间

]# df -h | grep vo

]# vgs

]# lvextend -L 18G /dev/systemvg/vo

]# vgs

]# lvs

2.扩展逻辑卷的文件系统（刷新文件系统）

xfs\_growfs：刷新xfs文件系统

resize2fs：刷新ext4文件系统

]# xfs\_growfs /dev/systemvg/vo

]# df -h | grep vo

]# lvs

* **卷组没有足够的剩余空间**

1.扩展卷组的空间

]# vgextend systemvg /dev/sdb3 /dev/sdb5

]# vgs

2.扩展逻辑卷的空间

]# vgs

]# lvextend -L 25G /dev/systemvg/vo

]# vgs

]# df -h | grep vo

3.扩展逻辑卷的文件系统（刷新文件系统）

]# xfs\_growfs /dev/systemvg/vo

]# df -h | grep vo

七、逻辑卷的补充

* **逻辑卷支持缩减**

xfs文件系统：不支持缩减

ext4文件系统：支持缩减

* **卷组划分空间的单位 PE**

默认1个PE的大小为4M

]# vgdisplay systemvg

PE Size 4.00 MiB

请创建一个大小为250M的逻辑卷名字为lvredhat

]# vgchange -s 1M systemvg #修改PE大小

]# vgdisplay systemvg #查看卷组详细信息

]# lvcreate -L 250M -n lvredhat systemvg

]# lvs

* 创建逻辑卷的时候指定PE个数
  + lvcreate -l PE个数 -n 逻辑卷名 卷组名

[root@localhost ~]# lvcreate -l 108 -n lvhaha systemvg

[root@localhost ~]# lvs

* **逻辑卷的删除**

删除卷组的前提：基于此卷组创建的所有逻辑卷，要全部删除

删除逻辑卷的前提：不能删除正在挂载使用的逻辑卷

[root@localhost ~]# **lvremove /dev/systemvg/vo**

Logical volume systemvg/vo contains a filesystem in use.

[root@localhost ~]# **umount /mylv/**

[root@localhost ~]# **lvremove /dev/systemvg/vo**

Do you really want to remove active logical volume systemvg/vo? [y/n]: **y**

Logical volume "vo" successfully removed

[root@localhost ~]# **vim /etc/fstab #仅删除vo开机自动挂载**

[root@localhost ~]# **lvremove /dev/systemvg/lvredhat**

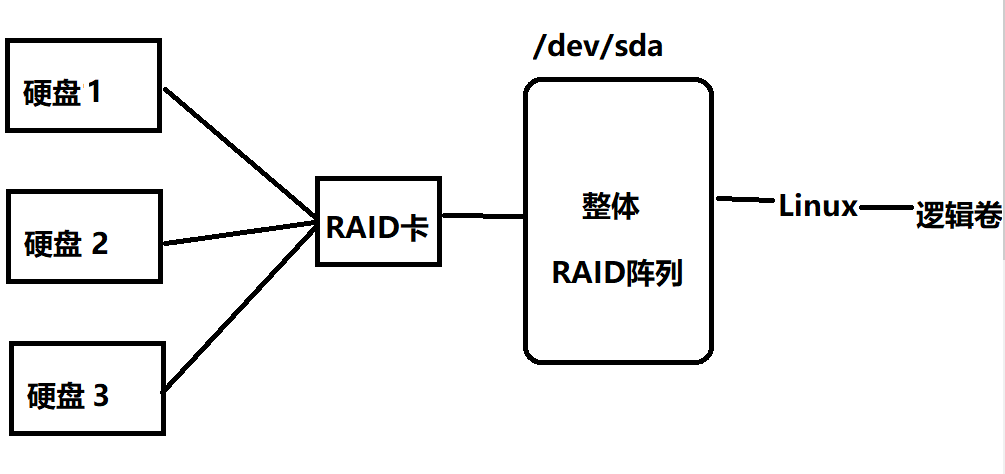
Do you really want to remove active logical volume systemvg/lvredhat? [y/n]: **y**

Logical volume "vo" successfully removed

八、RAID磁盘阵列

需要服务器硬件RAID卡

* 廉价冗余磁盘阵列
  + Redundant Arrays of Inexpensive Disks
  + 通过硬件/软件技术，将多个较小/低速的磁盘整合成一个大磁盘
  + 阵列的价值：提升I/O效率、硬件级别的数据冗余
  + 不同RAID级别的功能、特性各不相同



* **RAID 0**，条带模式
  + 同一个文档分散存放在不同磁盘
  + 并行写入以**提高效率**
  + 至少需要**两块**磁盘组成，磁盘利用率100%
* **RAID 1**，镜像模式
  + 一个文档复制成多份，分别写入不同磁盘
  + 多份拷贝**提高可靠性**，效率无提升
  + 至少需要**两块**磁盘组成，磁盘利用率50%
* **RAID5**，高性价比模式
  + 相当于RAID0和RAID1的折中方案
  + 需要至少一块磁盘的容量来存放校验数据
  + 至少需要**三块**磁盘组成，磁盘利用率n-1/n
* **RAID6**，高性价比/可靠模式
  + 相当于扩展的RAID5阵列，提供2份独立校验方案
  + 需要至少两块磁盘的容量来存放校验数据
  + 至少需要**四块**磁盘组成，磁盘利用率n-2/n
* **RAID 0+1**/**RAID 1+0**
  + 整合RAID 0、RAID 1的优势
  + 并行存取提高效率、镜像写入提高可靠性
  + 至少需要**四块**磁盘组成，磁盘利用率50%

九、进程管理

程序：静态没有执行的代码 硬盘空间

进程：动态执行的代码 CPU与内存资源

父进程与子进程 树型结构

进程编号：PID

* **pstree查看进程 课间嘻嘻：17：15上课**
* 常用命令选项
  + -a：显示完整的命令行
  + -p：列出对应进程的PID编号

systemd：所有进程的父进程（上帝进程）

[root@localhost ~]#**pstree -p lisi**

bash(9609)───vim(9656)

[root@localhost ~]# **pstree -a lisi**

bash

└─vim haha.txt

[root@localhost ~]# **pstree -ap lisi**