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]**

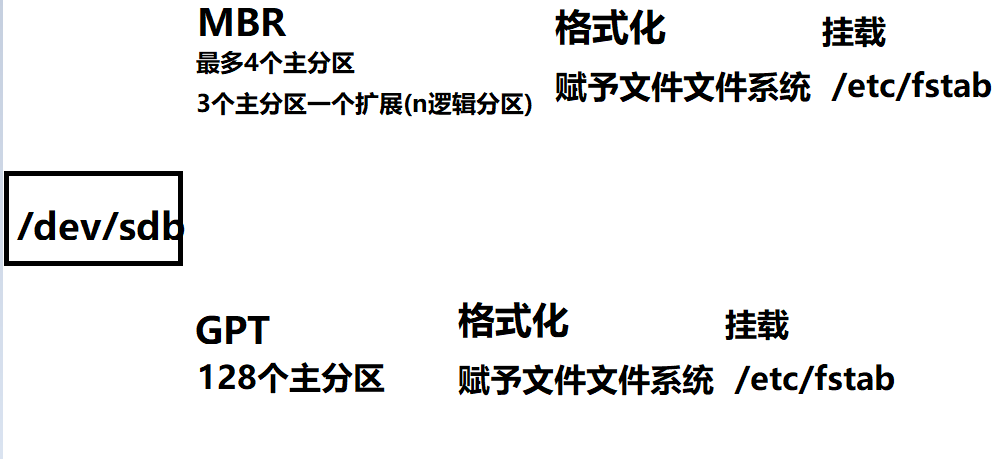
**name=xixi**

**baseurl=file:///centos7**

**enabled=1**

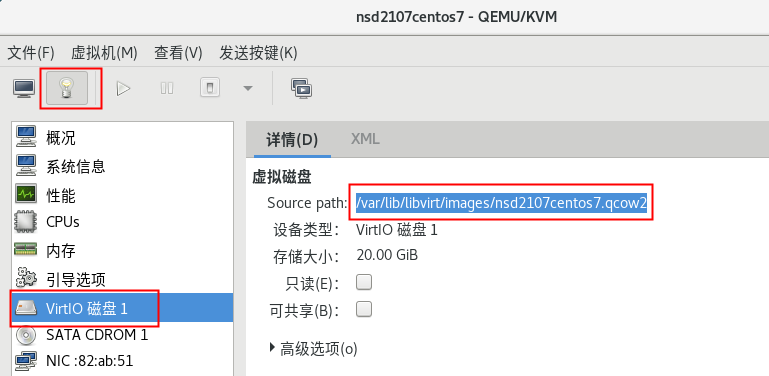
**gpgcheck=0**

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



KVM虚拟机缺点：

虚拟硬盘：真机创建一个大文件，利用大文件充当虚拟机的硬盘



[root@localhost ~]# **ls /var/lib/libvirt/images/**

[root@localhost images]# **df -h**

**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**

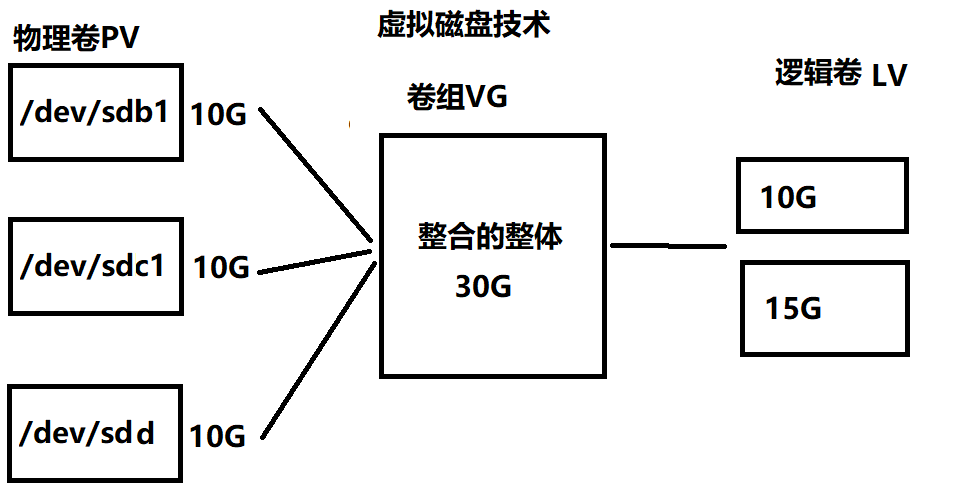
**p** 查看分区表

**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 | pvs | vgs | lvs |
| 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/sdb[5-6]**

]# **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**

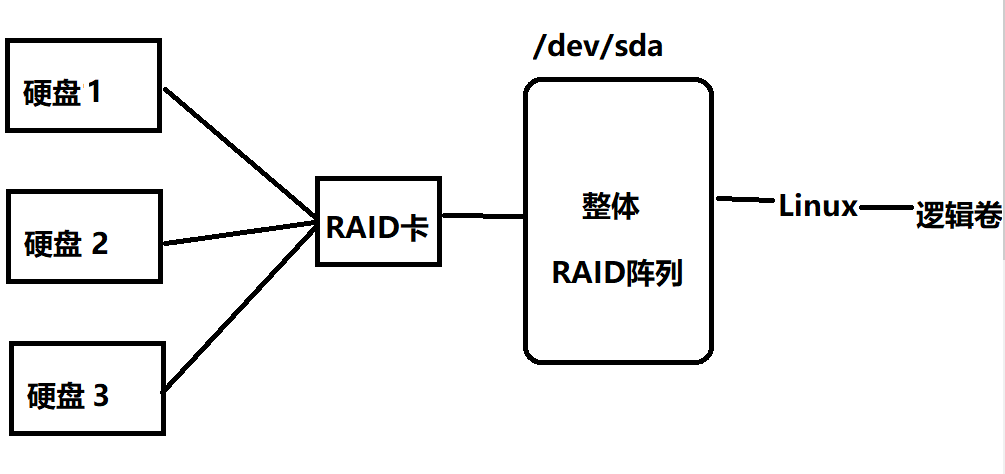
Do you really want to remove active logical volume systemvg/lvredhat? [y/n]: **y**

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**
* 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(PID永远为1)：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**

* **ps — Processes Snapshot**
  + 格式：ps [选项]...
* 常用命令选项
  + aux：显示当前终端所有进程（a）、当前用户在所有终端下的进程（x）、以用户格式输出（u）
  + -elf：显示系统内所有进程（-e）、以长格式输出（-l）信息、包括最完整的进程信息（-f）
* **ps aux** 操作
  + 列出正在运行的所有进程，显示进程信息非常详细

用户 进程ID %CPU %内存 虚拟内存 固定内存 终端 状态 起始时间 CPU时间 程序指令

* **ps -elf** 操作
  + 列出正在运行的所有进程，显示进程父进程信息
  + PPID为父进程的PID

请计算正在运行的进程有多少个？

[root@localhost ~]# wc -l /etc/passwd

[root@localhost ~]# ps aux | wc -l

[root@localhost ~]# ps -elf | wc -l

* **top 交互式工具（动态）**
  + 格式：top [-d 刷新秒数] [-U 用户名]

[root@localhost ~]# **top -d 1**

按大写P进行CPU排序

按大写M进行内存排序

* **pgrep — Process Grep**
  + 用途：pgrep [选项]... 查询条件
* 常用命令选项
  + **-l：输出进程名，而不仅仅是 PID**
  + -U：检索指定用户的进程
  + -x：精确匹配完整的进程名

]# **pgrep -l a #查找进程名字带有a字母**

]# **pgrep -l ab**

]# **pgrep -lU lisi #查找用户lisi开启的进程**

]# **pstree -p lisi #查找用户lisi开启的进程**

]# pgrep -x crond

]# pgrep -lx crond

**十、控制进程（进程前后台的调度）**

**进程的前后台调度**

* &符号：**正在运行**的状态放入后台
* Ctrl + z 组合键
  + 挂起当前进程（**暂停**并转入后台）
* jobs 命令
  + 查看后台任务列表
* fg 命令
  + 将后台任务恢复到前台运行
* bg 命令
  + 激活后台被挂起的任务

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

[root@localhost ~]# **xeyes**

^Z #按**Ctrl+z** 暂停放入后台

[1]+ 已停止 xeyes

[root@localhost ~]# **jobs**  #查看后台进程信息

[root@localhost ~]# **bg 1**  #让后台编号为1 的进程继续运行

[root@localhost ~]# **jobs** #查看后台进程信息

[root@localhost ~]# **fg 1**  #让后台编号为1 的进程恢复到前台

xeyes

^C #按**Ctrl+c 结束**

[root@localhost ~]#

* 干掉进程的不同方法
  + Ctrl+c 组合键，中断当前命令程序
  + kill [-9] PID... 、kill [-9] %后台任务编号
  + killall [-9] 进程名...
  + pkill [-9] 查找条件

[root@nb ~]# **xeyes &**

[1] 8332

[root@nb ~]# **xeyes &**

[2] 8339

[root@nb ~]# **xeyes &**

[3] 8346

[root@nb ~]# **killall -9 xeyes**

**十一、VDO 了解内容**

•Virtual Data Optimizer（虚拟数据优化器）

–一个内核模块，目的是通过重删减少磁盘的空间占用，以及减少复制带宽

–VDO是基于块设备层之上的，也就是在原设备基础上映射出mapper虚拟设备，然后直接使用即可

•重复数据删除

–输入的数据会判断是不是冗余数据

–判断为重复数据的部分不会被写入，然后对源数据进行更新，直接指向原始已经存储的数据块即可

•压缩

–对每个单独的数据块进行处理

[root@svr7 ~]# yum  -y  install  vdo  kmod-kvdo    #所需软件包

•制作VDO卷

•vdo基本操作：参考man vdo 全文查找/example

–vdo  create  --name=VDO卷名称  --device=设备路径 --vdoLogicalSize=逻辑大小

–vdo  list

–vdo  status  -n  VDO卷名称

–vdo  remove  -n  VDO卷名称

–vdostatus  [--human-readable] [/dev/mapper/VDO卷名称]

•VDO卷的格式化加速（跳过去重分析）：

–mkfs.xfs  –K   /dev/mapper/VDO卷名称

–mkfs.ext4  -E  nodiscard  /dev/mapper/VDO卷名称

前提制作VDO需要2G以上的内存

[root@nb ~]# vdo create --name=vdo0 --device=/dev/sdc --vdoLogicalSize=200G

[root@nb ~]# mkfs.xfs -K /dev/mapper/vdo0

[root@nb ~]# mkdir /nsd01

[root@nb ~]# mount /dev/mapper/vdo0 /nsd01

[root@nb ~]# df -h

[root@nb ~]# vdostats --hum /dev/mapper/vdo0 #查看vdo设备详细信息

[root@svr7 ~]# vim /etc/fstab

/dev/mapper/vdo0  /nsd01  xfs  defaults,\_netdev 0 0

课后习题：

案例1：复制、粘贴、移动

以root用户新建/exam/目录，在此目录下新建king.txt文件，并进一步完成下列操作

1）将“I Love hehe”写入到文件king.txt

2）将king.txt重命名为my.txt

3）将/etc/passwd、/boot、/etc/group同时拷贝到/exam/目录下

4）将ifconfig命令的前两行内容，追加写入king.txt

5）将主机名永久配置文件(/etc/hostname)，拷贝到/exam/目录下

6）将存放组基本信息的配置文件(/etc/group)，拷贝到/exam/目录下

7）将开机自动挂载配置文件，拷贝到/exam/目录下

案例2:虚拟机上操作,采用GPT分区模式，利用parted规划分区

添加一块30G的硬盘并规划分区：

划分2个2G的主分区；1个5G的主分区;

案例3:虚拟机上操作,交换分区使用

1、案例2中新添加30G硬盘的第一个主分区

– 格式化成交换文件系统，实现该分区开机自动启用

2、案例2中新添加30G硬盘的第二个主分区

– 格式化成交换文件系统，实现该分区开机自动启用

案例4:虚拟机上操作,文件扩展Swap空间

1. 使用dd命令创建一个大小为2048MB的交换文件，放在/opt/swap.db

2. 将swap.db文件格式化成swap文件系统

3. 启用swap.db文件，查看swap空间组成

4. 停用swap.db文件，查看swap空间组成

案例5:虚拟机操作，构建 LVM 存储

– 新建一个名为 systemvg 的卷组

– 在此卷组中创建一个名为 vo 的逻辑卷，大小为8G

– 将逻辑卷 vo 格式化为 xfs 文件系统

– 将逻辑卷 vo 挂载到 /vo 目录，并在此目录下建立一个测试文件 votest.txt，内容为“I AM KING.”

– 实现逻辑卷vo开机自动挂载到/vo

案例6:虚拟机操作，构建 LVM 存储(修改PE大小)

– 新的逻辑卷命名为 database，其大小为50个PE的大小，属于 datastore 卷组

– 在 datastore 卷组中其PE的大小为1M

– 使用 EXT4 文件系统对逻辑卷 database 格式化，此逻辑卷应该在开机时自动挂载到 /mnt/database 目录

案例7:虚拟机操作，扩展逻辑卷

– 将/dev/systemvg/vo逻辑卷的大小扩展到20G

案例8:进程管理

1.查看当前系统中整个进程树信息

2.利用pstree查看lisi开启的进程

3.显示当前系统正在运行的所有进程信息

4.检索当前系统中进程，进程名包含cron的PID是多少？