

硬盘分区&LVM逻辑卷

2020lrf227

一、硬盘分区

1.Linux虚拟机关机状态下添加一个100G和一个20G的SCSI接口硬盘

2.检测并确认新硬盘 fdisk -l 命令

3.规划硬盘中的分区 fdisk命令

4.创建文件系统mkfs、mkswap命令

5.创建XFS文件系统

6.创建ext4文件系统

8.mkswap命令

9.基于分区扩展SWAP分区

10.基于文件扩展SWAP分区

10.挂载文件系统、ISO镜像、mount命令

11.卸载已挂载的文件系统 umount命令

12.设置文件系统的自动挂载 /etc/fstab 文件-----面试题

13.查看磁盘使用情况 df命令

14.查看分区的UUID号 blkid命令

15.基于设备UUID挂载

16.大小超过2T的分区规划 parted命令

用法： parted [选项]...[设备[命令][参数]...]

帮助选项：

操作命令：

1.Parted工具详解：

2.开始通过parted工具来对磁盘进行操作

二、LVM逻辑卷管理

1.LVM概述--Logical Volume Manager

2.LVM基本概念

(1) PV (Physical Volume) 物理卷

(2) VG (Volume Group) 卷组

(3) LV (Logical Volume) 逻辑卷

3.LVM管理命令

4.物理卷管理

4.1 pvscan命令

4.2 pvcreate命令

4.3 pvdisplay命令

4.4 pvremove命令

5.卷组管理

5.1 vgscan命令

5.2 vgcreate命令

5.3 vgdisplay命令

5.4 vgremove命令

5.5 vgextend命令

6.逻辑卷管理

6.1 lvscan命令

6.2 lvcreate命令

6.3 lvdisplay命令

6.4格式化并使用逻辑卷

6.5 lvextend命令

6.6 执行xfs_growfs /dev/卷组名/逻辑卷名 命令以便重新识别文件系统的大小

6.7 lvremove命令

一、硬盘分区

1.Linux虚拟机**关机状态**下添加一个100G和一个20G的SCSI接口硬盘

2.检测并确认新硬盘 fdisk -l 命令

格式: fdisk -l [磁盘设备]

```
[root@localhost ~]# fdisk -l
```

磁盘 /dev/sda: 85.9 GB, 85899345920 字节, 167772160 个扇区

Units = 扇区 of 1 * 512 = 512 bytes

扇区大小(逻辑/物理): 512 字节 / 512 字节

I/O 大小(最小/最佳): 512 字节 / 512 字节

磁盘标签类型: dos

磁盘标识符: 0x000ab9f0

设备	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	2048	2099199	1048576	83	Linux
/dev/sda2		2099200	167772159	82836480	8e	Linux LVM

磁盘 /dev/sdb: 107.4 GB, 107374182400 字节, 209715200 个扇区

Units = 扇区 of 1 * 512 = 512 bytes

扇区大小(逻辑/物理): 512 字节 / 512 字节

I/O 大小(最小/最佳): 512 字节 / 512 字节

```
[root@localhost ~]# fdisk -l | grep "/dev/sd*"
```

磁盘 /dev/sda: 85.9 GB, 85899345920 字节, 167772160 个扇区

/dev/sda1 * 2048 2099199 1048576 83 Linux

/dev/sda2 2099200 167772159 82836480 8e Linux LVM

磁盘 /dev/sdb: 107.4 GB, 107374182400 字节, 209715200 个扇区

/dev/sdb1 2048 4196351 2097152 83 Linux

/dev/sdb2 4196352 8390655 2097152 83 Linux

/dev/sdb3 8390656 209715199 100662272 5 Extended

```

/dev/sdb5          8392704      10489855      1048576    82  Linux swap / Solaris
/dev/sdb6          10491904      20977663      5242880    83  Linux
磁盘 /dev/sdc: 107.4 GB, 107374182400 字节, 209715200 个扇区
磁盘 /dev/sdd: 107.4 GB, 107374182400 字节, 209715200 个扇区
磁盘 /dev/sde: 107.4 GB, 107374182400 字节, 209715200 个扇区

```

```
[root@localhost ~]# fdisk -l /dev/sda
```

```
磁盘 /dev/sda: 85.9 GB, 85899345920 字节, 167772160 个扇区
```

```
Units = 扇区 of 1 * 512 = 512 bytes
```

```
扇区大小(逻辑/物理): 512 字节 / 512 字节
```

```
I/O 大小(最小/最佳): 512 字节 / 512 字节
```

```
磁盘标签类型: dos
```

```
磁盘标识符: 0x000ab9f0
```

设备	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	2048	2099199	1048576	83	Linux
/dev/sda2		2099200	167772159	82836480	8e	Linux LVM

MBR (msdos dos) GPT(最多可以划分128个主分区)

带*号的是系统引导文件，对于系统启动很重要

对于已有的分区，将通过列表的方式输出以下信息：

- Device:分区的设备文件名称
- Boot: 是否是引导分区，是则有 “*”
- Start:该分区在硬盘中的起始位置（柱面数）
- End:该分区在硬盘中的结束（柱面数）
- Blocks:分区的大小，以Blocks（块）为单位，默认的块大小为1024字节
- Id:分区类型的ID标记号，对于EXT4分区为83，LVM分区为8e，swap分区为82，RAID分区为fd等
- System:分区类型

注意：新挂载的硬盘可能未包含任何分区会提示以下信息 (Disk/dev/sdb doesn't contain a valid partition table)

注意：分区类型的概念在windows中是没有的，分区类型与文件系统类型应一致，这样管理磁盘文件系统时才不容易引起混乱

3.规划硬盘中的分区 fdisk命令

作用：在交互式的操作环境中管理磁盘分区

格式：fdisk [磁盘设备]

常用选项：

m: 查看操作指令的帮助信息

p: 列表查看分区信息

n: 新建分区

d: 删除分区

t: 变更分区类型

w: 保存分区设置并退出

q: 放弃分区设置并退出

Ctrl+退格键：删除输入的错误字符

示例：

为主机新增一块100G的SCSI硬盘

对该硬盘进行分区

划分两个主分区，各10G，剩余空间为扩展分区

在扩展分区中建立两个逻辑分区，第一个逻辑分区2G，类型为swap。第二个逻辑分区为20G

```
[root@localhost ~]# fdisk -l /dev/sdb
```

磁盘 /dev/sdb: 107.4 GB, 107374182400 字节, 209715200 个扇区

Units = 扇区 of 1 * 512 = 512 bytes

扇区大小(逻辑/物理): 512 字节 / 512 字节

I/O 大小(最小/最佳): 512 字节 / 512 字节

磁盘标签类型: dos

磁盘标识符: 0x398f19af

设备	Boot	Start	End	Blocks	Id	System
/dev/sdb1		2048	4196351	2097152	83	Linux
/dev/sdb2		4196352	8390655	2097152	83	Linux
/dev/sdb3		8390656	209715199	100662272	5	Extended
/dev/sdb5		8392704	10489855	1048576	82	Linux swap / Solaris
/dev/sdb6		10491904	20977663	5242880	83	Linux

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

欢迎使用 fdisk (util-linux 2.23.2)。
更改将停留在内存中，直到您决定将更改写入磁盘。
使用写入命令前请三思。

命令(输入 m 获取帮助): p

磁盘 /dev/sdb: 107.4 GB, 107374182400 字节, 209715200 个扇区

Units = 扇区 of 1 * 512 = 512 bytes

扇区大小(逻辑/物理): 512 字节 / 512 字节

I/O 大小(最小/最佳): 512 字节 / 512 字节

磁盘标签类型: dos

磁盘标识符: 0x398f19af

设备	Boot	Start	End	Blocks	Id	System
/dev/sdb1		2048	4196351	2097152	83	Linux
/dev/sdb2		4196352	8390655	2097152	83	Linux
/dev/sdb3		8390656	209715199	100662272	5	Extended
/dev/sdb5		8392704	10489855	1048576	82	Linux swap / Solaris
/dev/sdb6		10491904	20977663	5242880	0	Empty

命令(输入 m 获取帮助): t

分区号 (1-3,5,6, 默认 6): 6

Hex 代码(输入 L 列出所有代码): L

0	空	24	NEC DOS	81	Minix / 旧 Linu	bf	Solaris
1	FAT12	27	隐藏的 NTFS Win	82	Linux 交换 / So	c1	DRDOS/sec (FAT-
2	XENIX root	39	Plan 9	83	Linux	c4	DRDOS/sec (FAT-
3	XENIX usr	3c	PartitionMagic	84	OS/2 隐藏的 C:	c6	DRDOS/sec (FAT-
4	FAT16 <32M	40	Venix 80286	85	Linux 扩展	c7	Syrinx
5	扩展	41	PPC PReP Boot	86	NTFS 卷集	da	非文件系统数据
6	FAT16	42	SFS	87	NTFS 卷集	db	CP/M / CTOS / .
7	HPFS/NTFS/exFAT	4d	QNX4. x	88	Linux 纯文本	de	Dell 工具
8	AIX	4e	QNX4. x 第2部分	8e	Linux LVM	df	BootIt
9	AIX 可启动	4f	QNX4. x 第3部分	93	Amoeba	e1	DOS 访问
a	OS/2 启动管理器	50	OnTrack DM	94	Amoeba BBT	e3	DOS R/0
b	W95 FAT32	51	OnTrack DM6 Aux	9f	BSD/OS	e4	SpeedStor
c	W95 FAT32 (LBA)	52	CP/M	a0	IBM Thinkpad 休	eb	BeOS fs
e	W95 FAT16 (LBA)	53	OnTrack DM6 Aux	a5	FreeBSD	ee	GPT
f	W95 扩展 (LBA)	54	OnTrackDM6	a6	OpenBSD	ef	EFI (FAT-12/16/
10	OPUS	55	EZ-Drive	a7	NeXTSTEP	f0	Linux/PA-RISC

11	隐藏的 FAT12	56	Golden Bow	a8	Darwin UFS	f1	SpeedStor
12	Compaq 诊断	5c	Priam Edisk	a9	NetBSD	f4	SpeedStor
14	隐藏的 FAT16 <3	61	SpeedStor	ab	Darwin 启动	f2	DOS 次要
16	隐藏的 FAT16	63	GNU HURD or Sys	af	HFS / HFS+	fb	VMware VMFS
17	隐藏的 HPFS/NTF	64	Novell Netware	b7	BSDI fs	fc	VMware VMKCORE
18	AST 智能睡眠	65	Novell Netware	b8	BSDI swap	fd	Linux raid 自动
1b	隐藏的 W95 FAT3	70	DiskSecure 多启	bb	Boot Wizard 隐	fe	LANstep
1c	隐藏的 W95 FAT3	75	PC/IX	be	Solaris 启动	ff	BBT
1e	隐藏的 W95 FAT1	80	旧 Minix				

Hex 代码(输入 L 列出所有代码): 83

已将分区 “Empty” 的类型更改为 “Linux”

命令(输入 m 获取帮助): P

磁盘 /dev/sdb: 107.4 GB, 107374182400 字节, 209715200 个扇区

Units = 扇区 of 1 * 512 = 512 bytes

扇区大小(逻辑/物理): 512 字节 / 512 字节

I/O 大小(最小/最佳): 512 字节 / 512 字节

磁盘标签类型: dos

磁盘标识符: 0x398f19af

设备	Boot	Start	End	Blocks	Id	System
/dev/sdb1		2048	4196351	2097152	83	Linux
/dev/sdb2		4196352	8390655	2097152	83	Linux
/dev/sdb3		8390656	209715199	100662272	5	Extended
/dev/sdb5		8392704	10489855	1048576	82	Linux swap / Solaris
/dev/sdb6		10491904	20977663	5242880	83	Linux

命令(输入 m 获取帮助): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

正在同步磁盘。

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

[root@localhost ~]# partx -u /dev/sdb

上面两个作用相同，使系统加载分区表，识别新的分区（有时需要重启操作系统才能识别）

4.创建文件系统mkfs、mkswap命令

mkfs命令

作用：创建文件系统（格式化）Make Filesystem

格式：mkfs -t 文件系统类型 分区设备

常用选项：

- -t:指定格式化文件类型
- -b:指定block大小，单位为字节
- -l:inode大小
- -U:设置UUID号
- -q:执行时不显示任何信息

```
[root@localhost ~]# mkfs    #两次Tab键
```

```
mkfs          mkfs.cramfs  mkfs.ext3    mkfs.fat      mkfs.msdos    mkfs.xfs
mkfs.btrfs    mkfs.ext2    mkfs.ext4    mkfs.minix    mkfs.vfat
```

5.创建XFS文件系统

创建文件夹才能用

```
[root@localhost ~]# mkfs -t xfs /dev/sdb1 或者 mkfs.xfs /dev/sdb1
meta-data=/dev/sdb1          isize=512    agcount=4, agsize=131072 blks
                        =                               sectsz=512   attr=2,
projid32bit=1
                        =                               crc=1      finobt=0, sparse=0
data                =                               bsize=4096  blocks=524288,
imaxpct=25
                        =                               sunit=0     swidth=0 blks
naming             =version 2                   bsize=4096  ascii-ci=0 ftype=1
log                =internal log                 bsize=4096  blocks=2560, version=2
                        =                               sectsz=512   sunit=0 blks, lazy-
count=1
realtime            =none                        extsz=4096  blocks=0, rtextents=0
[root@localhost ~]# mkdir /data1
[root@localhost ~]# mount /dev/sdb1 /data1
[root@localhost ~]# cd /data1/
[root@localhost data1]# ls
```

6.创建ext4文件系统


```
[root@localhost ~]# mkfs -t ext4 /dev/sdb2 = mkfs.ext4 /dev/sdb2
```

创建文件夹才能用

7. 创建FAT32文件系统

```
[root@localhost ~]# mkfs -t vfat -F 32 /dev/sdb6
```

```
mkfs.fat 3.0.20 (12 Jun 2013)
```

创建文件夹才能用

8.mkswap命令

作用：创建交换机文件系统

格式：mkswap分区设备

9.基于分区扩展SWAP分区

```
[root@localhost ~]# mkswap /dev/sdb5
```

正在设置交换空间版本 1，大小 = 1048572 KiB

无标签，UUID=5f657f98-556b-495d-a935-f3712cb2b805

```
[root@localhost ~]# swapon -s
```

文件名	类型	大小	已用	权限
/dev/dm-1	partition	2097148	0	-1

```
[root@localhost ~]# grep -i "swaptotal" /proc/meminfo
```

```
SwapTotal:      2097148 kB
```

```
[root@localhost ~]# cat /proc/meminfo
```

```
MemTotal:      997956 kB
```

```
MemFree:       203968 kB
```

```
MemAvailable:  403432 kB
```

```
Buffers:       22992 kB
```

```
Cached:        296116 kB
```

```
SwapCached:    0 kB
```

```
Mlocked:       0 kB
```

```
SwapTotal:     2097148 kB
```

```
SwapFree:      2097148 kB
```

基于分区扩展大小，需要创建分区，不是特别方便



i944



从已有的硬盘来弄



35-18044

10.基于文件扩展SWAP分区

```
[root@localhost ~]# dd if=/dev/zero of=/swapfile bs=1M count=200
```

记录了200+0 的读入

记录了200+0 的写出

209715200字节 (210 MB) 已复制, 10.9967 秒, 19.1 MB/秒

```
[root@localhost ~]# du -sh /swapfile
```

200M /swapfile

```
[root@localhost ~]# chmod 600 /swapfile
```

```
[root@localhost ~]# mkswap /swfile
```

正在设置交换空间版本 1, 大小 = 307196 KiB

无标签, UUID=47c02cc6-c3f3-4203-867d-42740e66b8eb

```
[root@localhost ~]# swapon /swfile
```

```
[root@localhost ~]# grep -i "swaptotal" /proc/meminfo
```

SwapTotal: 2097148 kB

```
[root@localhost ~]# swapon -s
```

文件名	类型	大小	已用	权限
/dev/dm-1	partition	2097148 0	-1	

10.挂载文件系统、ISO镜像、mount命令

作用：挂载文件系统、ISO镜像

格式：

```
mount [-t类型] 存储设备 挂载点目录
mount -o loop ISO 镜像文件 挂载点目录
mount 查看当前的挂载
mount -a 挂载/etc/fstab中已记录的所有挂载
```

```
[root@localhost ~]# mkdir /{dianying,xuexi,youxi}
[root@localhost ~]# mount /dev/sdb1 /dianying/
[root@localhost ~]# mount /dev/sdb2 /xuexi/
[root@localhost ~]# mount /dev/sdb3 /youxi/
```

```
[root@localhost ~]# mount | grep sdb
/dev/sdb6 on /youxi type vfat
(rw,relatime,fmask=0022,dmask=0022,codepage=437,ioccharset=ascii,short
name=mixed,errors=remount-ro)
/dev/sdb1 on /dianying type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
/dev/sdb2 on /xuexi type ext4 (rw,relatime,seclabel,data=ordered)
```

#前提是上传一个测试IOS镜像文件到Linux系统中

```
[root@localhost ~]# vim /etc/fstab
[root@localhost ~]# rz
```

```
[root@localhost ~]# mkdir /vmtools
[root@localhost ~]# mount /root/linux.iso /vmtools
mount: /dev/loop0 写保护，将以只读方式挂载
[root@localhost ~]# mount | tail -1
/root/linux.iso on /vmtools type iso9660 (ro,relatime)
[root@localhost ~]# ls /vmtools/
manifest.txt      VMwareTools-10.3.2-9925305.tar.gz  vmware-tools-upgrader-64
run_upgrader.sh   vmware-tools-upgrader-32
```

11.卸载已挂载的文件系统 umount命令

作用：卸载已挂载的文件系统

格式：

umount 存储设备位置

umount 挂载点目录

umount -a 卸载所有/etc/fstab已记录的挂载 --不要轻易使用

光驱

eject = > eject -t -----弹出光驱

卸载

```
[root@localhost ~]# umount /dev/sdb6
```

```
[root@localhost ~]# umount /xuexi
```

```
[root@localhost ~]# mount | grep sdb
```

```
[root@localhost ~]# umount -a
```

umount 后面可以跟 设备 文件所在目录

umount: /: 目标忙。

(有些情况下通过lsf(8)或fuser(1)可以找到有关使用该设备的进程的有用信息)

umount: /sys/fs/cgroup: 目标忙

(有些情况下通过lsf(8)或fuser(1)可以找到有关使用该设备的进程的有用信息)

umount: /run:目标忙

(有些情况下通过lsf(8)或fuser(1)可以找到有关使用该设备的进程的有用信息)

umount: /dev:目标忙

(有些情况下通过lsf(8)或fuser(1)可以找到有关使用该设备的进程的有用信息)

```
[root@localhost ~]#mount -a
```

12.设置文件系统的自动挂载 /etc/fstab 文件-----面试题

```
[root@localhost ~]# vim /etc/fstab
```

#文件最后追加四行

/dev/sdb1	/data1	xfs	defaults	0 0
/dev/sdb2	/data2	ext4	defaults	0 0
UUID="0450-0157"	/data3	vfat	defaults	0 0
/dev/cdrom	/media/cdrom/	iso9660	defaults	0 0

```
[root@localhost ~]# mount -a
```

mount: /dev/sr0 写保护, 将以只读方式挂载

```
[root@localhost ~]# vim /etc/fstab
[root@localhost ~]# mount | grep sdb
/dev/sdb1 on /data1 type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
/dev/sdb2 on /data2 type ext4 (rw,relatime,seclabel,data=ordered)
/dev/sdb3 on /data3 type vfat
(rw,relatime,fmask=0022,dmask=0022,codepage=437,iocharset=ascii,short
name=mixed,errors=remount-ro)
```

常见参数:

```
/dev/sdb1                /dianying                xfs
defaults, noauto, ro      0 1
```

defaults: 默认

auto: 系统自动挂载, fstab默认就是这个选项

noauto: 开机不自动挂载

nouser: 只有超级用户可以挂载

ro: 按制度权限挂载 (read only)

rw: 按可读可写权限挂载

user: 任何用户都可以挂载

```
[root@localhost ~]# cat /etc/fstab
```

```
#
# /etc/fstab
# Created by anaconda on Mon Jan  6 00:04:38 2020
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos-root / xfs defaults 0 0
UUID=a981f4c4-dafc-4a70-afe5-f242b7662655 /boot xfs
defaults 0 0
/dev/mapper/centos-home /home xfs defaults 0 0
/dev/mapper/centos-swap swap swap defaults 0 0
```

/dev/sdb1	/data1	xfs	defaults	0 0
/dev/sdb2	/data2	ext4	defaults	0 0
UUID="0450-0157"	/data3	vfat	defaults	0 0
/dev/cdrom	/media/cdrom/	iso9660	defaults	0 0

dump备份设置 --第一列

/dev/sdb1	/dianying	ext4	defaults
1	2		

当其值为1时，将允许dump备份程序备份；设置为0时，忽略备份操作

fsck磁盘检查设置；检查文件设备的完整性 --第二列

/dev/sdb1	/dianying	ext4	defaults
1	2		

其值是一个顺序，当其值为0时，永远不检查；而/根目录分区永远都为1 优先检查；其他分区一般为2 次要检查。

13.查看磁盘使用情况 df命令

作用：查看磁盘使用情况

格式：df [选项] [文件]

常用选项：

- -h: 显示更易读的容量单位
- -T: 显示对应文件系统的类型
- -i: 显示inode数量

[root@localhost ~]# df -Th

文件系统	类型	容量	已用	可用	已用%	挂载点
/dev/mapper/centos-root	xfs	50G	4.9G	46G	10%	/
devtmpfs	devtmpfs	975M	0	975M	0%	/dev
tmpfs	tmpfs	992M	0	992M	0%	/dev/shm
tmpfs	tmpfs	992M	11M	982M	2%	/run
tmpfs	tmpfs	992M	0	992M	0%	/sys/fs/cgroup
/dev/mapper/centos-home	xfs	46G	33M	46G	1%	/home
/dev/sda1	xfs	1014M	157M	858M	16%	/boot
tmpfs	tmpfs	199M	12K	199M	1%	/run/user/42
tmpfs	tmpfs	199M	0	199M	0%	/run/user/0
/dev/sdb1	xfs	5.0G	33M	5.0G	1%	/data1

/dev/sdb2	ext4	4.8G	20M	4.6G	1%	/data2
/dev/sdb3	vfat	5.0G	4.0K	5.0G	1%	/data3
/dev/loop0	iso9660	55M	55M	0	100%	/vmtools
/dev/sr0	iso9660	4.2G	4.2G	0	100%	/media/cdrom

```
[root@localhost ~]# df -Th /dev/sdb1
```

文件系统	类型	容量	已用	可用	已用%	挂载点
/dev/sdb1	xfs	5.0G	33M	5.0G	1%	/data1

查看类命令很重要

```
[root@localhost ~]# who
```

```
root    tty2          2020-02-26 17:24
root    pts/0          2020-02-26 17:25 (192.168.200.128)
```

```
[root@localhost ~]# w
```

```
20:02:15 up 2:38, 2 users, load average: 0.02, 0.04, 0.05
USER      TTY      FROM          LOGIN@      IDLE   JCPU   PCPU WHAT
root      tty2                17:24      2:37m  0.07s  0.07s -bash
root      pts/0    192.168.200.128 17:25      7.00s  0.20s  0.03s w
```

```
[root@localhost ~]# uptime
```

```
20:02:29 up 2:38, 2 users, load average: 0.01, 0.04, 0.05
```

```
[root@localhost ~]# df -i
```

文件系统	Inode	已用 (I)	可用 (I)
已用 (I)%	挂载点		
/dev/mapper/centos-root	26214400 162275	26052125	1%
/			
devtmpfs	249476 404		249072
1%	/dev		
tmpfs	253768 1		253767
1%	/dev/shm		
tmpfs	253768 908		252860
1%	/run		
tmpfs	253768 16		253752
1%	/sys/fs/cgroup		

```

/dev/mapper/centos-home 23654400    10                23654390        1%
/home
/dev/sda1                524288          326                523962
1%                        /boot
tmpfs                    253768           9                  253759
1%                        /run/user/42
tmpfs                    253768           1                  253767
1%                        /run/user/0
/dev/sdb1                2621440          3                  2621437
1%                        /data1
/dev/sdb2                327680           11                 327669
1%                        /data2
/dev/sdb3                0                0                  0
-                        /data3
/dev/loop0               0                0                  0
-                        /vmtools
/dev/sr0                  0                0
0                        -                /media/cdrom

```

当我们创建一个文件的时候就会消耗一个inode

```

[root@localhost ~]# df -Th /dev/sdb1          #带单位
文件系统      类型  容量  已用  可用  已用% 挂载点
/dev/sdb1     xfs   5.0G   33M   5.0G    1% /data1
[root@localhost ~]# df -i /dev/sdb1
文件系统      Inode  已用(I)  可用(I)  已用(I)% 挂载点
/dev/sdb1     2621440      3 2621437      1% /data1

```

14.查看分区的UUID号 blkid命令

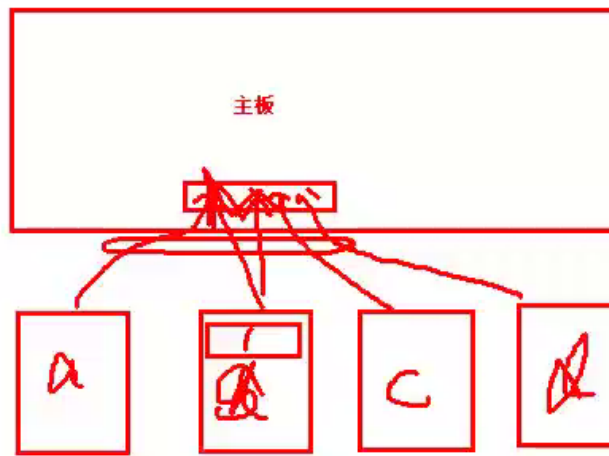
UUID号：分区必须格式化后才会有UUID号

格式：blkid 分区设备

15.基于设备UUID挂载

UUID号是唯一的，uuid 识别设备

写fskid的时候最好写上UID号



l" TYPE="xfs" ~~SDB1~~

```
[root@localhost ~]# umount /dev/sdb3
```

```
[root@localhost ~]# blkid /dev/sdb3
```

```
/dev/sdb3: UUID="0450-0157" TYPE="vfat"
```

```
[root@localhost ~]# vim /etc/fstab
```

```
UUID="0450-0157"          /data3                      vfat      defaults      0 0
```

```
[root@localhost ~]# mount /dev/sdb3
```

```
[root@localhost ~]# mount | grep sdb3
```

```
/dev/sdb3 on /data3 type vfat
```

```
(rw,relatime,fmask=0022,dmask=0022,codepage=437,ioccharset=ascii,short
name=mixed,errors=remount-ro)
```

inode知识补充:

inode包含文件的元信息（文件的属性），

一个文件有两类数据：文件的内容；文件的属性

具体来说有以下内容：

- 文件的字节数
- 文件拥有者的User ID
- 文件的Group ID
- 文件的读、写、执行权限
- 文件的时间戳：共有三个，**ctime**指inode上一次变动的时间，**mtime**指文件内容上一次变动的时间，**atime**指文件上一次打开的时间
- 链接数，即有多少文件名指向这个inode
- 文件数据block的位置

可以用stat命令，查看某个文件的inode信息：stat example.txt

inode的大小：128字节，其中最后4字节为记录block位置的数据

mke2fs -N 修改inode数

```
[root@localhost ~]# stat initial-setup-ks.cfg
```

文件: "initial-setup-ks.cfg"

大小: 1822 块: 8 IO 块: 4096 普通文件

设备: fd00h/64768d Inode: 100663386 硬链接: 1

权限: (0644/-rw-r--r--) Uid: (0/ root) Gid: (0/ root)

环境: system_u:object_r:admin_home_t:s0

最近访问: 2020-01-06 00:23:09.613996137 +0800

最近更改: 2020-01-06 00:23:09.614996137 +0800

最近改动: 2020-01-06 00:23:09.614996137 +0800

创建时间: -

16.大小超过2T的分区规划 parted命令

通常划分分区工具我们用的比较多是fdisk命令，但是现在由于磁盘越来越廉价，而且磁盘空间越来越大。而fdisk工具他对分区是有大小限制的，它只能划分小于2T的磁盘。现在的磁盘空间已经远远大于2T，有两个方法来解决这个问题：其一是通过卷管理来实现，其二就是通过Parted工具来实现对2T磁盘进行分区操作。

GPT格式的磁盘相当于原来MBR磁盘中原来保留4个partition table的4*16个字节，只留第一个16个字节，类似于扩展分区，真正的partition table在512字节之后，GPT分区方式没有四个主分区的限制，最多可达到128个主分区。

用法: parted [选项]...[设备[命令]][参数]...

将带有“参数”的命令应用于“设备”。如果没有给出“命令”，则以交互模式运行

帮助选项:

-h, --help	显示此求助信息
-l, ---list	列出所有识别的分区信息
-i, --interactive	在必要时，提示用户
-s, --script	从不提示用户
-v, --version	显示版本

操作命令:

检查MINOR #对文件
系统进行一个简单的检查

cp[FROM-DEVICE] FROM-MINOR TO-MINOR #将文件系统复制到另一个分区

help[COMMAND] #打印通用求助信息，或关于COMMAND的信息

mklabel 标签类型 #创建新的
磁盘标签（分区表）

mkfs MINOR 文件系统类型 #在MINOR创建类型为“文件系统类型”的文件系统

mkpart 分区类型[文件系统类型] 起始点 终止点 #创建一个分区

mkpartfs 分区类型 文件系统类型 起始点 终止点 #创建一个带有文件系统的分区

move MINOR 起始点 终止点 #移动编号为MINOR的分区

name MINOR 名称 #将编号为MINOR的分区命名为“名称”

print [MINOR] #打印分区表，或者分区

quit
#退出程序

rescue 起始点 终止点 #挽救临近“起始点”、“终止点”的遗失的分区

resize MINOR #改变位于编号为MINOR的分区中文件系统的大小

rm MINOR #删除编号为MINOR的分区

select 设备 #选择要编辑的设备

set MINOR 标志 状态 #改变编号为MINOR的分区标志

1.Parted工具详解:

1.1 进入Parted的方法（在命令行输入Parted命令即可）

```
[root@localhost ~]# parted
```

```
GNU Parted 3.1                                #Parted的软件版本号
使用 /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.    #欢迎信息
(parted)
```

类似fdisk一样，先要选择要分区的硬盘，此处为/dev/sdc：

```
[root@localhost ~]# parted
GNU Parted 3.1
使用 /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) select /dev/sdc
使用 /dev/sdc
(parted)
或者[root@localhost ~]# parted /dev/sdc
GNU Parted 3.1
使用 /dev/sdc
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

1.2获取parted工具帮助的方法：（只需输入help即可）

```
[root@localhost ~]# parted
(parted) help
或者[root@localhost ~]# parted --help
```

1.3退出parted工具（只需输入quit即可）

```
[root@localhost ~]# parted
(parted) quit
```

2.开始通过parted工具来对磁盘进行操作

2.1查看单个磁盘状态

```
[root@localhost ~]# parted /dev/sdc
GNU Parted 3.1
使用 /dev/sdc
Welcome to GNU Parted! Type 'help' to view a list of commands.
```

(parted) p #查看

磁盘分区状态

错误: /dev/sdc: unrecognised disk label #由于没有打上磁盘标签,
所以表现信息无法列出

Model: VMware, VMware Virtual S (scsi)

Disk /dev/sdc: 107GB

Sector size (logical/physical): 512B/512B

Partition Table: unknown

Disk Flags:

(parted) mklabel #创建磁

盘标签

新的磁盘标签类型? gpt #输入磁盘标签

类型 gpt msdos (MBR)

(parted) p #再

次查看磁盘分区状态

Model: VMware, VMware Virtual S (scsi)

Disk /dev/sdc: 107GB

Sector size (logical/physical): 512B/512B

Partition Table: gpt

Disk Flags:

Number	Start	End	Size	File system	Name	标志
--------	-------	-----	------	-------------	------	----

2.2查看所有磁盘状态

```
[root@localhost ~]# parted -l /dev/sdc
```

2.3通过parted工具来创建分区

```
[root@localhost ~]# parted /dev/sdc
```

GNU Parted 3.1

使用 /dev/sdc

Welcome to GNU Parted! Type 'help' to view a list of commands.

(parted) mkpart #创建分区

分区名称? []? data1 #分区名称

文件系统类型? [ext2]? ext4 #文件系统类型

起始点? 1 #指定起始位置
结束点? 5G #指定结束位置
(parted) p #查看磁盘分区状态
Model: VMware, VMware Virtual S (scsi) #显示磁盘类型为SCSI
Disk /dev/sdc: 107GB #磁盘大小
Sector size (logical/physical): 512B/512B #扇区大小
Partition Table: gpt #分区类型为GPT
Disk Flags:

Number	Start	End	Size	File system	Name	标志
1	1049kB	5000MB	4999MB		data1	

用mkpart创建多个分区

(parted) p
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdc: 107GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number	Start	End	Size	File system	Name	标志
1	1049kB	5000MB	4999MB		data1	
2	5000MB	10.0GB	5001MB		data2	
3	10.0GB	15.0GB	5000MB		data3	

2.4通过parted工具删除一个分区

(parted) rm 3 #删除分区3
(parted) p
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdc: 107GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number	Start	End	Size	File system	Name	标志
1	1049kB	5000MB	4999MB		data1	
2	5000MB	10.0GB	5001MB		data2	

2.5通过parted工具来格式化分区创建文件系统

- 老的版本用 (parted)mkfs 1 ext4 ,现在用不了了
- 下面退出parted模式, 用mkfs来格式化

```
[root@localhost ~]# mkfs -t ext4 /dev/sdc1
```

mke2fs 1.42.9 (28-Dec-2013)

文件系统标签=

OS type: Linux

块大小=4096 (log=2)

分块大小=4096 (log=2)

Stride=0 blocks, Stripe width=0 blocks

305216 inodes, 1220352 blocks

61017 blocks (5.00%) reserved for the super user

第一个数据块=0

Maximum filesystem blocks=1249902592

38 block groups

32768 blocks per group, 32768 fragments per group

8032 inodes per group

Superblock backups stored on blocks:

32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: 完成

正在写入inode表: 完成

Creating journal (32768 blocks): 完成

Writing superblocks and filesystem accounting information: 完成

```
[root@localhost ~]# mkfs.xfs /dev/sdc2
```

meta-data=/dev/sdc2 isize=512 agcount=4, agsize=305216 blks

= sectsz=512 attr=2, projid32bit=1

= crc=1 finobt=0, sparse=0

data = bsize=4096 blocks=1220864, imaxpct=25

```

=                                sunit=0      swidth=0 blks
naming  =version 2              bsize=4096  ascii-ci=0 ftype=1
log      =internal log          bsize=4096  blocks=2560, version=2
=                                sectsz=512    sunit=0 blks, lazy-count=1
realtime =none                  extsz=4096   blocks=0, rtextents=0

```

```
[root@localhost ~]# mkfs.xfs /dev/sdc3
```

```

meta-data=/dev/sdc3            isize=512    agcount=4, agsize=305152 blks
=                                sectsz=512    attr=2, projid32bit=1
=                                crc=1        finobt=0, sparse=0
data      =                    bsize=4096    blocks=1220608, imaxpct=25
=                                sunit=0      swidth=0 blks
naming    =version 2          bsize=4096    ascii-ci=0 ftype=1
log        =internal log      bsize=4096    blocks=2560, version=2
=                                sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none                extsz=4096   blocks=0, rtextents=0

```

2.6挂载刚才格式化后的文件系统

```
[root@localhost ~]# mkdir /data1
```

```
[root@localhost ~]# mount /dev/sdc1 /data1
```

```
[root@localhost ~]# mount | tail -1
```

```
/dev/sdc1 on /data1 type ext4 (rw,relatime,seclabel,data=ordered)
```

```
[root@localhost ~]# mkdir /data1/dir{1..10}
```

```
[root@localhost ~]# ls /data1/
```

```
dir1 dir10 dir2 dir3 dir4 dir5 dir6 dir7 dir8 dir9 lost+found
```

2.7删除和恢复分区

```
[root@localhost ~]# umount /dev/sdc3
```

```
[root@localhost ~]# parted /dev/sdc
```

```
GNU Parted 3.1
```

```
使用 /dev/sdc
```

```
Welcome to GNU Parted! Type 'help' to view a list of commands.
```

```
(parted) rm 3
```



```
(parted) p
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdc: 107GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
```

Number	Start	End	Size	File system	Name	标志
1	1049kB	5000MB	4999MB	ext4	data1	
2	5000MB	10.0GB	5001MB	xfs	data2	

```
(parted) rescue #恢复分区
起始点? 10G
结束点? 15G
正在搜索文件系统... 45% (剩余时间 00:01)信息: A xfs primary partition was found
at 10.0GB -> 15.0GB
. Do you want to add it to the partitiontable?
是/Yes/否/No/放弃/Cancel? yes
(parted) quit
信息: You may need to update /etc/fstab.
```

```
[root@localhost ~]# ls -l /dev/sdc*
brw-rw----. 1 root disk 8, 32 2月 27 02:57 /dev/sdc
brw-rw----. 1 root disk 8, 33 2月 27 02:57 /dev/sdc1
brw-rw----. 1 root disk 8, 34 2月 27 02:57 /dev/sdc2
brw-rw----. 1 root disk 8, 35 2月 27 02:57 /dev/sdc3
```

二、LVM逻辑卷管理

注意：逻辑分区和逻辑卷不是一个概念。

问题一：服务的日志存放在/var/log目录下，现在空间不够怎么办？

问题二：数据库软件安装在/usr/local/mysql目录下，现在空间不够怎么办？

答案：是普通分区吗？那没办法！

1.LVM概述--Logical Volume Manager

许多Linux操作系统的使用者在安装操作系统时都会遇到这样的困境：如何精确评估和分配各个硬盘分区的容量。如果当初估计不准确，一旦系统分区不够用就可能不得不备份，删除相关数据，甚至被迫重新规划分区并重装操作系统，以满足应用系统的需要。

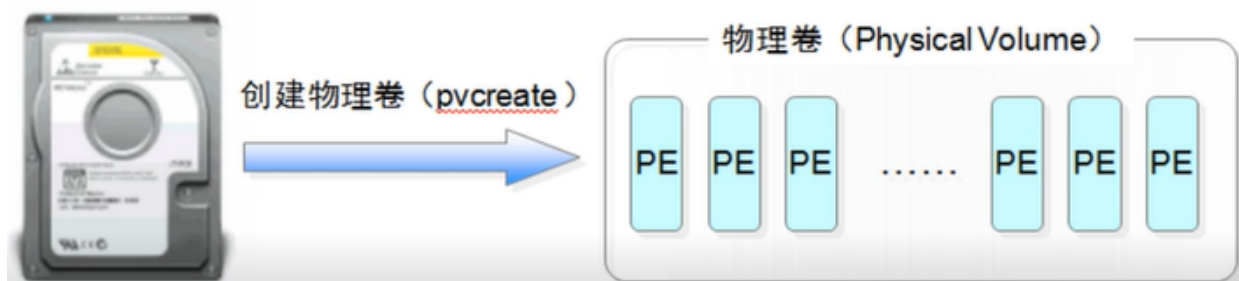
LVM是Linux操作系统中对磁盘的分区进行管理的一种逻辑机制，它是建立在硬盘和分区之上，文件系统之下的一个逻辑层，在建立文件系统时屏蔽了下层的磁盘分区布局，因此能够在保持现有数据不变的情况下动态调整磁盘容量，从而增强磁盘管理的灵活性。

在安装CentOS操作系统的过程中选择自动分区是，就会默认采用LVM分区方案，而不再需要进行手动配置。需要注意的是LVM逻辑卷的建立依赖于内核，因此/boot分区不能基于LVM创建，必须独立出来。

2.LVM基本概念

(1) PV (Physical Volume) 物理卷

物理卷是LVM机制的基本存储设备，通常对应为一个普通分区或整个硬盘，创建物理卷时，会在分区或硬盘头部创建一个保留区块，用于记录LVM属性，并把存储空间分割成默认大小为4MB的基本单元（Physical Exent, PE），构成物理卷。物理卷一般直接使用设备文件名称。



分区建议将ID 转为8e，若是整个硬盘将所有空间划分为一个主分区。



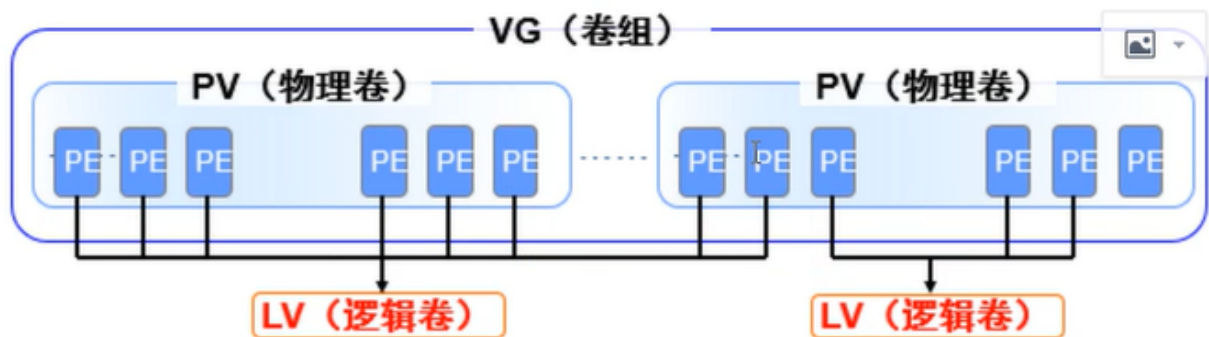
(2) VG (Volume Group) 卷组

由一个或多个物理卷组成一个整体，即成为卷组，在卷组中可以动态添加或移除物理卷可以分成不同的卷组，卷组的名称用户自己定义。



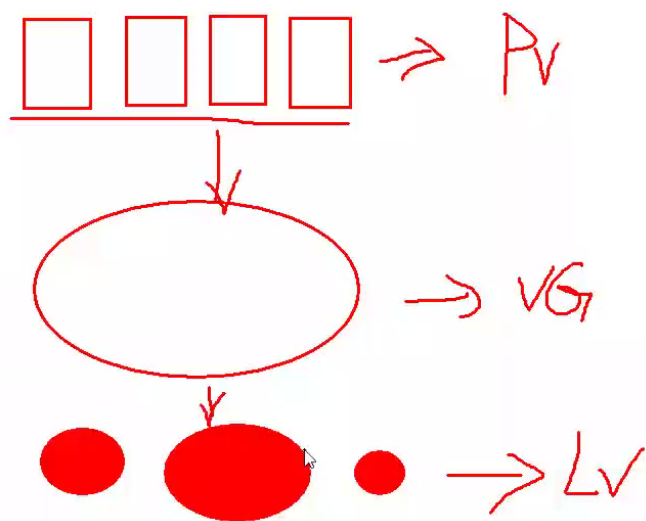
(3) LV (Logical Volume) 逻辑卷

逻辑卷建立在卷组之上，与物理卷没有直接关系，对于逻辑卷来说，每一个卷组就是一个整体，从这个整体中分割出一块空间，作为用户创建文件系统的基础，这一小块空间就称为逻辑卷。逻辑卷经过格式化创建了文件系统后，就可以挂载使用了。



逻辑卷创建步骤：准备分区-->创建物理卷-->创建卷组-->创建逻辑卷-->格式化（形成文件系统）-->挂载使用

面粉袋原理



3.LVM管理命令

功能	物理卷管理	卷组管理	逻辑卷管理
扫描scan	pvscan/pvs	vgscan/vgs	lvscan/lvs
建立create	pvcreate	vgcreate	lvcreate
显示display	pvdisk	vgdisplay	lvdisplay
删除remove	pvremove	vgremove	lvremove
扩展extend		vgextend	lvextend

减少reduce		vgreduce	lvreduce
----------	--	----------	----------

使用fdisk工具在磁盘设备“/dev/sdb”中划分出三个主分区/dev/sdb1,/dev/sdb21,/dev/sdb3每个分区大小为5G，将分区类型ID修改为8e。

ps:物理卷无法扩展和减少，它本身是一个分区一块硬盘

```
[root@localhost ~]# fdisk /dev/sdd
欢迎使用 fdisk (util-linux 2.23.2)。
```

更改将停留在内存中，直到您决定将更改写入磁盘。
使用写入命令前请三思。

Device does not contain a recognized partition table
使用磁盘标识符 0xd96ab1da 创建新的 DOS 磁盘标签。

```
命令(输入 m 获取帮助): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
分区号 (1-4, 默认 1):
起始 扇区 (2048-209715199, 默认为 2048):
将使用默认值 2048
Last 扇区, +扇区 or +size {K,M,G} (2048-209715199, 默认为 209715199): +5G
分区 1 已设置为 Linux 类型, 大小设为 5 GiB
```

```
命令(输入 m 获取帮助): n
Partition type:
   p   primary (1 primary, 0 extended, 3 free)
   e   extended
Select (default p): p
分区号 (2-4, 默认 2):
起始 扇区 (10487808-209715199, 默认为 10487808):
将使用默认值 10487808
Last 扇区, +扇区 or +size {K,M,G} (10487808-209715199, 默认为 209715199): +5G
分区 2 已设置为 Linux 类型, 大小设为 5 GiB
```

命令(输入 m 获取帮助): n

Partition type:

p primary (2 primary, 0 extended, 2 free)

e extended

Select (default p): p

分区号 (3, 4, 默认 3):

起始 扇区 (20973568-209715199, 默认为 20973568):

将使用默认值 20973568

Last 扇区, +扇区 or +size {K,M,G} (20973568-209715199, 默认为 209715199): +5G

分区 3 已设置为 Linux 类型, 大小设为 5 GiB

命令(输入 m 获取帮助): p

磁盘 /dev/sdd: 107.4 GB, 107374182400 字节, 209715200 个扇区

Units = 扇区 of 1 * 512 = 512 bytes

扇区大小(逻辑/物理): 512 字节 / 512 字节

I/O 大小(最小/最佳): 512 字节 / 512 字节

磁盘标签类型: dos

磁盘标识符: 0xd96ab1da

设备	Boot	Start	End	Blocks	Id	System
/dev/sdd1		2048	10487807	5242880	83	Linux
/dev/sdd2		10487808	20973567	5242880	83	Linux
/dev/sdd3		20973568	31459327	5242880	83	Linux

命令(输入 m 获取帮助): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

正在同步磁盘。

[root@localhost ~]# partx /dev/sdd

NR	START	END	SECTORS	SIZE	NAME	UUID
1	2048	10487807	10485760	5G		

```
2 10487808 20973567 10485760 5G
3 20973568 31459327 10485760 5G
```

4.物理卷管理

4.1 pvscan命令

pvscan命令用于扫描系统中所有的物理卷，并输出相关信息。

```
[root@localhost ~]# pvscan
PV /dev/sda2   VG centos           lvm2 [<99.00 GiB / 4.00 MiB free]
Total: 1 [<99.00 GiB] / in use: 1 [<99.00 GiB] / in no VG: 0 [0   ]
```

4.2 pvcreate命令

格式: pvcreate 设备1 [设备2...]

pvcreate命令用于将整个硬盘或分区转换为物理卷，主要是添加LVM属性信息并划分为PE存储单位，该命令需要使用硬盘或分区作为参数。

```
[root@localhost ~]# pvcreate /dev/sdd[123]
Physical volume "/dev/sdd1" successfully created.
Physical volume "/dev/sdd2" successfully created.
Physical volume "/dev/sdd3" successfully created.
[root@localhost ~]# pvscan
PV /dev/sda2   VG centos           lvm2 [<99.00 GiB / 4.00 MiB free]
PV /dev/sdd3   VG                 lvm2 [5.00 GiB]
PV /dev/sdd1   VG                 lvm2 [5.00 GiB]
PV /dev/sdd2   VG                 lvm2 [5.00 GiB]
Total: 4 [<114.00 GiB] / in use: 1 [<99.00 GiB] / in no VG: 3 [15.00 GiB]
[root@localhost ~]# pvs
PV          VG      Fmt  Attr  PSize   PFree
/dev/sda2   centos  lvm2 a--  <99.00g 4.00m
/dev/sdd1   myvg    lvm2 a--  <5.00g  <5.00g
/dev/sdd2   myvg    lvm2 a--  <5.00g  <5.00g
/dev/sdd3           lvm2 ---   5.00g   5.00g
```

4.3 pvdisplay命令

```
[root@localhost ~]# pvdisplay /dev/sdd1
"/dev/sdd1" is a new physical volume of "5.00 GiB"
--- NEW Physical volume ---
```

PV Name	/dev/sdd1
VG Name	
PV Size	5.00 GiB
Allocatable	NO
PE Size	0
Total PE	0
Free PE	0
Allocated PE	0
PV UUID	nLybFn-3qJx-qjSw-2IJp-GWNZ-001C-fzZWld

4.4 pvremove命令

pvremove命令用于将物理卷还原成普通的分区或磁盘，不在用于LVM体系，被移除的物理卷将无法被pvscan识别。

```
[root@localhost ~]# pvremove /dev/sdd3
```

5.卷组管理

5.1 vgscan命令

vgscan命令用于扫描系统中已建立的LVM卷组及相关信息。

```
[root@localhost ~]# vgscan
```

```
Reading volume groups from cache.
```

```
Found volume group "centos" using metadata type lvm2
```

```
[root@localhost ~]# vgs
```

VG	#PV	#LV	#SN	Attr	VSize	VFree
centos	1	3	0	wz--n-	<99.00g	4.00m
myvg	2	0	0	wz--n-	9.99g	9.99g

5.2 vgcreate命令

格式: vgcreate 卷组名 物理卷名1 物理卷名2... 选项-s 指定PE大小（单位M）

vgcreate命令用于将一个或多个物理卷创建一个卷组，第一个命令参数用于设置卷组名称，其后依次指定需要加入该卷组的物理卷作为参数。

```
[root@localhost ~]# vgcreate myvg /dev/sdd1 /dev/sdd2
```

```
Volume group "myvg" successfully created
```

5.3 vgdisplay命令

vgdisplay命令用于显示系统中各类卷组的详细信息，需要使用指定卷组名作为参数，未指定卷组默认

```
[root@localhost ~]# vgdisplay myvg
```

```
--- Volume group ---
```

```
VG Name                myvg
```

```
System ID
```

```
Format                 lvm2
```

```
Metadata Areas        2
```

```
Metadata Sequence No  1
```

```
VG Access              read/write
```

```
VG Status              resizable
```

```
MAX LV                0
```

```
Cur LV               0
```

```
Open LV              0
```

```
Max PV               0
```

```
Cur PV              2
```

```
Act PV               2
```

```
VG Size              9.99 GiB
```

```
PE Size              4.00 MiB
```

```
Total PE            2558
```

```
Alloc PE / Size      0 / 0
```

```
Free PE / Size       2558 / 9.99 GiB
```

```
VG UUID              KgOvSH-ERDf-tKb2-dTj1-UdQK-BejL-OIVdR0
```

5.4 vgremove命令

vgremove命令用于删除指定的卷组名称作为参数即可。删除时应确保该卷组中没有正在使用的逻辑卷

```
[root@localhost ~]# vgremove myvg
```

5.5 vgextend命令

vgextend命令用于扩展卷组的磁盘空间，当创建了新的物理卷，需要将其加入到已有的卷组中，就可以使用vgextend命令。该命令的第一个参数为需要扩展容量的卷组名称，其后为需要添加到该卷组中的各物理卷。

```
[root@localhost ~]# vgcreate myvg /dev/sdb1
```

```
[root@localhost ~]# vgextend myvg /dev/sdb2
```

6.逻辑卷管理

6.1 lvscan命令

lvscan命令用于扫描系统中已建立的逻辑卷及相关信息。

```
[root@localhost ~]# lvscan
ACTIVE          '/dev/centos/swap' [<3.88 GiB] inherit
ACTIVE          '/dev/centos/home' [<45.12 GiB] inherit
ACTIVE          '/dev/centos/root' [50.00 GiB] inherit
[root@localhost ~]# lvs
LV   VG      Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync
Convert
home centos -wi-ao----- <45.12g
root centos -wi-ao----- 50.00g
swap centos -wi-ao----- <3.88g
```

6.2 lvcreate命令

lvcreate命令用于从指定的卷组中分割空间，以创建新的逻辑卷。需要指定逻辑卷大小，名称及所在的卷组名作为参数。

格式：lvcreate -L 容量大小 （-l 指定PE个数）-n 逻辑卷名 卷组名

```
[root@localhost ~]# lvcreate -n lv001 -L 8G myvg
Logical volume "lv001" created.
[root@localhost ~]# ls -l /dev/myvg/lv001
lrwxrwxrwx. 1 root root 7 2月 26 23:48 /dev/myvg/lv001 -> ../dm-3
```

6.3 lvdisplay命令

lvdisplay命令用于像是逻辑卷的详细信息，可以指定逻辑卷的设备文件作为参数，也可以使用卷组名为参数，以显示该卷组中的所有逻辑卷的信息。

```
[root@localhost ~]# lvdisplay /dev/myvg/lv001
--- Logical volume ---
LV Path                /dev/myvg/lv001
LV Name                 lv001
VG Name                 myvg
LV UUID                 U1H1iq-QJM7-5N0o-eD3F-8EQy-D3pq-iPx1SQ
LV Write Access         read/write
LV Creation host, time localhost.localdomain, 2020-02-26 23:48:45 +0800
LV Status                available
# open                  0
```

LV Size	8.00 GiB
Current LE	2048
Segments	2
Allocation	inherit
Read ahead sectors	auto
- currently set to	8192
Block device	253:3

6.4格式化并使用逻辑卷

```
[root@localhost ~]# mkfs -t ext4 /dev/myvg/lv001
```

mke2fs 1.42.9 (28-Dec-2013)

文件系统标签=

OS type: Linux

块大小=4096 (log=2)

分块大小=4096 (log=2)

Stride=0 blocks, Stripe width=0 blocks

524288 inodes, 2097152 blocks

104857 blocks (5.00%) reserved for the super user

第一个数据块=0

Maximum filesystem blocks=2147483648

64 block groups

32768 blocks per group, 32768 fragments per group

8192 inodes per group

Superblock backups stored on blocks:

32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: 完成

正在写入inode表: 完成

Creating journal (32768 blocks): 完成

Writing superblocks and filesystem accounting information: 完成

强制由ext4文件系统转换为xfs文件系统

```
[root@localhost ~]# mkfs -t xfs -f /dev/myvg/lv001
```

meta-data=/dev/myvg/lv001	isize=512	agcount=4, agsize=524288 blks
=	sectsz=512	attr=2, projid32bit=1

```

=                                crc=1          finobt=0, sparse=0
data    =                        bsize=4096      blocks=2097152, imaxpct=25
=                                sunit=0         swidth=0 blks
naming  =version 2              bsize=4096      ascii-ci=0 ftype=1
log      =internal log          bsize=4096      blocks=2560, version=2
=                                sectsz=512     sunit=0 blks, lazy-count=1
realtime =none                  extsz=4096      blocks=0, rtextents=0

```

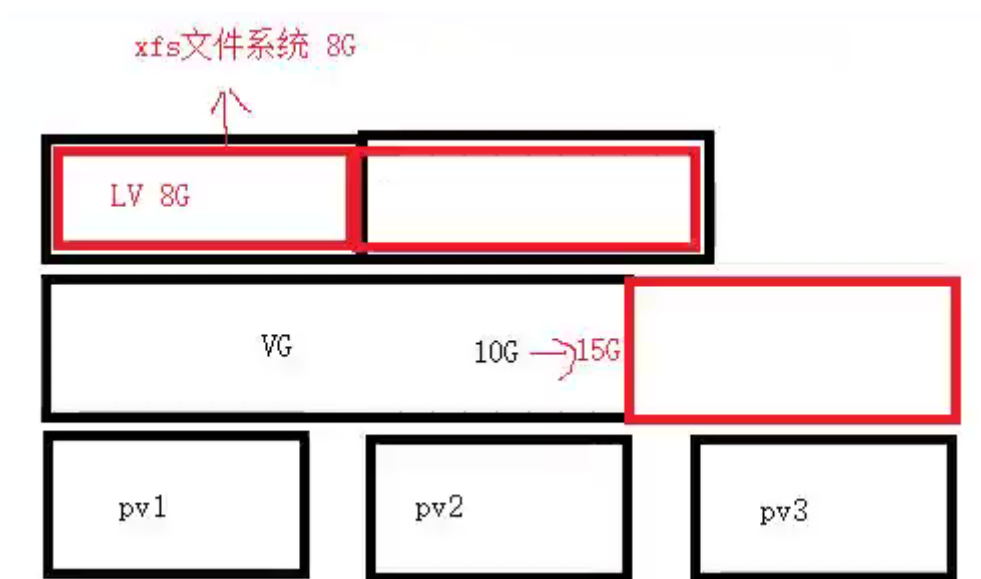
挂载: UUID号--->blkid /dev/myvg/lv001

```
[root@localhost ~]# mkdir /data
```

```
[root@localhost ~]# mount /dev/myvg/lv001 /data
```

```
[root@localhost ~]# df -Th /dev/myvg/lv001
```

文件系统	类型	容量	已用	可用	已用%	挂载点
/dev/mapper/myvg-lv001	xfs	8.0G	33M	8.0G	1%	/data



6.5 lvextend命令

lvextend命令用于动态扩展逻辑卷的空间，当目前使用的逻辑卷空间不足时，可以从所在卷组中分割额外的空间进行扩展。只要指定需要增加的容量大小及逻辑卷设备文件位置即可。前提是该卷组中还有尚未分配的磁盘空间，否则需要先扩展组容量。

组扩展:

```
[root@localhost ~]# vgextend vg001 /dev/sdd3
```

```
Volume group "vg001" not found
```

```
Cannot process volume group vg001
```

```
[root@localhost ~]# vgs
```

```

VG      #PV #LV #SN Attr   VSize   VFree
centos   1   3   0 wz--n- <99.00g 4.00m
myvg     2   1   0 wz--n-   9.99g 1.99g
[root@localhost ~]# vgextend myvg /dev/sdd3
Volume group "myvg" successfully extended
[root@localhost ~]# vgs
VG      #PV #LV #SN Attr   VSize   VFree
centos   1   3   0 wz--n- <99.00g 4.00m
myvg     3   1   0 wz--n- <14.99g <6.99g

```

逻辑扩展:

```

[root@localhost ~]# lvextend -L +4G /dev/myvg/lv001
Size of logical volume myvg/lv001 changed from 8.00 GiB (2048 extents) to
12.00 GiB (3072 extents
).
```

Logical volume myvg/lv001 successfully resized.

```

[root@localhost ~]# lvdisplay
--- Logical volume ---
LV Path                /dev/centos/swap
LV Name                 swap
VG Name                 centos
LV UUID                 pUvmrf-VIwv-NtB9-wNCW-NKlV-uCOR-qros8L
LV Write Access         read/write
LV Creation host, time localhost.localdomain, 2020-01-06 00:04:37 +0800
LV Status                available
# open                  2
LV Size                 <3.88 GiB
Current LE              992
Segments                1
Allocation              inherit
Read ahead sectors      auto
- currently set to     8192
Block device            253:1

--- Logical volume ---
LV Path                /dev/centos/home
LV Name                 home

```

VG Name	centos
LV UUID	nKcyPo-xVQL-Ysk4-Kfrs-a1BZ-MTEp-dbcURk
LV Write Access	read/write
LV Creation host, time	localhost.localdomain, 2020-01-06 00:04:37 +0800
LV Status	available
# open	1
LV Size	<45.12 GiB
Current LE	11550
Segments	1
Allocation	inherit
Read ahead sectors	auto
- currently set to	8192
Block device	253:2

--- Logical volume ---

LV Path	/dev/centos/root
LV Name	root
VG Name	centos
LV UUID	CV1LoZ-5miO-NZoJ-20tk-5Unc-Vc02-dqCx7P
LV Write Access	read/write
LV Creation host, time	localhost.localdomain, 2020-01-06 00:04:37 +0800
LV Status	available
# open	1
LV Size	50.00 GiB
Current LE	12800
Segments	1
Allocation	inherit
Read ahead sectors	auto
- currently set to	8192
Block device	253:0

--- Logical volume ---

LV Path	/dev/myvg/lv001
LV Name	lv001
VG Name	myvg

```

LV UUID                U1H1iq-QJM7-5N0o-eD3F-8EQy-D3pq-iPx1SQ
LV Write Access         read/write
LV Creation host, time localhost.localdomain, 2020-02-26 23:48:45 +0800
LV Status               available
# open                  0
LV Size                 12.00 GiB
Current LE              3072
Segments                3
Allocation              inherit
Read ahead sectors     auto
- currently set to     8192
Block device            253:3

```

6.6 执行xfs_growfs /dev/卷组名/逻辑卷名 命令以便重新识别文件系统的大小

```
[root@localhost ~]# xfs_growfs /dev/myvg/lv001
```

```

meta-data=/dev/mapper/myvg-lv001 isize=512    agcount=4, agsize=524288 blks
         =                               sectsz=512    attr=2, projid32bit=1
         =                               crc=1        finobt=0 spinodes=0
data     =                               bsize=4096    blocks=2097152, imaxpct=25
         =                               sunit=0      swidth=0 blks
naming   =version 2                     bsize=4096    ascii-ci=0 ftype=1
log      =internal                     bsize=4096    blocks=2560, version=2
         =                               sectsz=512    sunit=0 blks, lazy-count=1
realtime =none                         extsz=4096    blocks=0, rtextents=0
data blocks changed from 2097152 to 3145728

```

```
[root@localhost ~]# df -Th
```

文件系统	类型	容量	已用	可用	已用%	挂载点
/dev/mapper/centos-root	xfs	50G	4.9G	46G	10%	/
devtmpfs	devtmpfs	975M	0	975M	0%	/dev
tmpfs	tmpfs	992M	0	992M	0%	/dev/shm
tmpfs	tmpfs	992M	11M	982M	2%	/run
tmpfs	tmpfs	992M	0	992M	0%	/sys/fs/cgroup
/dev/mapper/centos-home	xfs	46G	33M	46G	1%	/home
/dev/sdal	xfs	1014M	157M	858M	16%	/boot

tmpfs	tmpfs	199M	12K	199M	1%	/run/user/42
tmpfs	tmpfs	199M	0	199M	0%	/run/user/0
/dev/sdc1	ext4	4.5G	19M	4.2G	1%	/data1
/dev/sdc2	xfs	4.7G	33M	4.7G	1%	/data2
/dev/loop0	iso9660	55M	55M	0	100%	/vmtools
/dev/sr0	iso9660	4.2G	4.2G	0	100%	/media/cdrom
/dev/sdc3	xfs	4.7G	33M	4.7G	1%	/data3
/dev/mapper/myvg-lv001	xfs	12G	33M	12G	1%	/data

```
[root@localhost ~]# df -Th /dev/myvg/lv001
```

文件系统	类型	容量	已用	可用	已用%	挂载点
/dev/mapper/myvg-lv001	xfs	12G	33M	12G	1%	/data

```
ext4文件系统: [root@localhost ~]# resize2fs /dev/myvg/lv001
```

注意：xfs文件系统只支持增大分区空间的情况，不支持减小的情况（切记！！）

6.7 lvremove命令

lvremove命令用于删除指定的逻辑卷，直接使用逻辑卷的设备文件作为参数即可

```
[root@localhost ~]# umount /dev/myvg/lv001
```

```
[root@localhost ~]# lvremove /dev/myvg/lv001
```

```
Do you really want to remove active logical volume myvg/lv001? [y/n]: y
```

```
Logical volume "lv001" successfully removed
```

```
[root@localhost ~]# vgremove myvg
```

```
Volume group "myvg" successfully removed
```

```
[root@localhost ~]# pvremove /dev/sdd[123]
```

```
Labels on physical volume "/dev/sdd1" successfully wiped.
```

```
Labels on physical volume "/dev/sdd2" successfully wiped.
```

```
Labels on physical volume "/dev/sdd3" successfully wiped.
```

```
[root@localhost ~]# fdisk /dev/sdd
```

欢迎使用 fdisk (util-linux 2.23.2)。

更改将停留在内存中，直到您决定将更改写入磁盘。

使用写入命令前请三思。

命令(输入 m 获取帮助): d
分区号 (1-3, 默认 3): 1
分区 1 已删除

命令(输入 m 获取帮助): d
分区号 (2,3, 默认 3): 2
分区 2 已删除

命令(输入 m 获取帮助): d
已选择分区 3
分区 3 已删除
嗯~好啦, 回到解放前啦~

