

# 分区规划及使用 **LVM** 逻辑卷 管理交换空间

磁道: track

扇区: sector

磁头: head

柱面: cylinder

每个扇区, 512 字节

- 识别硬盘 => 分区规划 => 格式化 => 挂载使用

毛坯楼层-----》打隔断-----》装修-----》入驻

- MBR/msdos 分区模式
  - 1~4 个主分区,或者 0~3 个主分区+1 个扩展分区(n 个逻辑分区)
  - 最大支持容量为 2.2TB 的磁盘
  - 扩展分区不能格式化

---

## 1. 查看硬盘

```
[root@server0 /]# lsblk
```

## 2. 修改硬盘的分区表

```
[root@server0 /]# fdisk /dev/vdb
```

- n** 创建新的分区----->回车----->回车----->回车----->在 **last** 结束时 **+2G**
- p** 查看分区表
- n** 创建新的分区----->回车----->回车----->回车----->在 **last** 结束时 **+2G**
- d** 删除分区
- w** 保存并退出

```
[root@server0 /]# lsblk
```

```
[root@server0 /]# ls /dev/vdb[1-2]
```

## 3. 格式化文件系统

### • mkfs 工具集

- mkfs.ext3 分区设备路径
- mkfs.ext4 分区设备路径
- mkfs.xfs 分区设备路径
- mkfs.vfat -F 32 分区设备路径

```
[root@server0 /]# mkfs.ext4 /dev/vdb1
```

```
[root@server0 /]# blkid /dev/vdb1
```

#查看分区文件系统命令

```
[root@server0 /]# mkfs.xfs /dev/vdb2
[root@server0 /]# blkid /dev/vdb2          #查看分区文件系统的命令
```

## 4. 挂载使用

```
[root@server0 ~]# mount /dev/vdb1 /part1
[root@server0 ~]# mount /dev/vdb2 /part2
```

```
[root@server0 ~]# df -h    #查看正在挂载的分区使用情况
```

## 5. 开机自动挂载

- 配置文件 **/etc/fstab** 的记录格式

- 设备路径	挂载点	类型	参数	备份标记	检测顺序
--------	-----	----	----	------	------

```
[root@server0 ~]# vim /etc/fstab
/dev/vdc5 /mnt/mypart ext4 defaults 0 0
[root@server0 ~]# tail -2 /etc/fstab
/dev/vdb1 /part1 ext4 defaults 0 0
/dev/vdb2 /part2 xfs defaults 0 0
```

```
[root@server0 ~]# df -h
```

验证:

```
[root@server0 ~]# mount -a
```

检测**/etc/fstab** 开机自动挂载配置文件,格式是否正确

检测**/etc/fstab** 中,书写完成,但当前没有挂载的设备,进行挂载

```
[root@server0 ~]# df -h
```

---

---

```
[root@server0 ~]# fdisk /dev/vdb  #划分第三个分区
```

```
[root@server0 ~]# partprobe    刷新    新的分区
```

---

---

总结:

- |              |         |
|--------------|---------|
| 1.lsblk      | 查看新的磁盘  |
| 2.fdisk      | 划分新的分区  |
| 3.partprobe  | 刷新      |
| 4.mkfs       | 格式化文件系统 |
| 5.mount      | 挂载使用    |
| 6./etc/fstab | 开机自动    |
- 
-

## 综合分区：

1.添加一个 80G 的虚拟硬盘

```
[root@server0 ~]# lsblk
```

2.划分区如下：

划分 3 个主分区，每个 10G

划分 3 个逻辑分区，每个 5G

```
[root@server0 ~]# fdisk /dev/vdc
```

**p** 查看分区表

**n** 创建新的分区----->回车----->回车----->回车----->在 **last** 结束时 **+10G**

**n** 创建新的分区----->回车----->回车----->回车----->在 **last** 结束时 **+10G**

**n** 创建新的分区----->回车----->回车----->回车----->在 **last** 结束时 **+10G**

**p** 查看分区表

**n** 创建新的分区

----->回车----->起始回车----->结束回车 将所有空间给扩展分区

**p** 查看分区表

**n** 创建新的分区----->起始回车----->结束**+5G**

**n** 创建新的分区----->起始回车----->结束**+5G**

**n** 创建新的分区----->起始回车----->结束**+5G**

**d** 删除分区

**w** 保存并退出

```
[root@server0 ~]# ls /dev/vdc[1-7]
```

---

---

## LVM 逻辑卷管理

作用： 1.整合分散的空间

2.LVM 逻辑卷空间可以动态的扩大或者缩减

零散空闲存储 ---- 整合的虚拟磁盘 ---- 虚拟的分区

将众多的物理卷，组成卷组，在从卷组中还分出逻辑卷

面粉 ----->大面团----->小面团----->蒸----->吃

砖----->房子----->打隔断-----> 装修----->入住

---

---

## 建立 LVM 逻辑卷

### 1. 创建卷组

命令格式: **vgcreate** 卷组名 分区路径

```
[root@server0 ~]# vgcreate myvg /dev/vdc[1-2]
```

```
[root@server0 ~]# vgs          #查看卷组基本信息
```

```
[root@server0 ~]# pvs          #查看物理卷基本信息
```

### 2. 创建逻辑卷

命令格式: **lvcreate -L** 大小 **-n** 逻辑卷名称 基于卷组名

```
[root@server0 ~]# lvcreate -L 16G -n mylv myvg
```

Logical volume "mylv" created

```
[root@server0 ~]# lvs
```

```
[root@server0 ~]# mkfs.ext4 /dev/myvg/mylv
```

```
[root@server0 ~]# vim /etc/fstab
```

```
/dev/myvg/mylv /lvm ext4 defaults 0 0
```

```
[root@server0 ~]# mkdir /lvm
```

```
[root@server0 ~]# mount -a
```

```
[root@server0 ~]# df -h
```

---

---

---

## LVM 逻辑卷的扩展

### 一、卷组有足够的剩余空间

#### 1. 直接扩展逻辑卷的空间

```
[root@server0 ~]# vgs
```

```
[root@server0 ~]# lvs
```

```
[root@server0 ~]# lvextend -L 18G /dev/myvg/mylv
```

```
[root@server0 ~]# lvs
```

#### 2. 扩展逻辑卷文件系统

扩展 ext4: `resize2fs`

扩展 xfs: `xfs_growfs`

```
[root@server0 ~]# blkid /dev/myvg/mylv
```

```
[root@server0 ~]# df -h
```

```
[root@server0 ~]# resize2fs /dev/myvg/mylv
```

```
[root@server0 ~]# df -h
```

### 二、卷组没有足够的剩余空间

#### 1. 扩展卷组

```
[root@server0 ~]# vgs
```

```
[root@server0 ~]# vgextend myvg /dev/vdc3
[root@server0 ~]# vgs
```

## 2. 直接扩展逻辑卷的空间

```
[root@server0 ~]# lvs
[root@server0 ~]# lvextend -L 25G /dev/myvg/mylv
[root@server0 ~]# lvs
```

## 3. 扩展逻辑卷文件系统

扩展 ext4: `resize2fs`

扩展 xfs: `xfs_growfs`

```
[root@server0 ~]# blkid /dev/myvg/mylv
[root@server0 ~]# df -h
[root@server0 ~]# resize2fs /dev/myvg/mylv
[root@server0 ~]# df -h
```

---

---

## LVM 其他内容

1. LVM 的缩减（了解）
2. PE 单位：卷组划分空间单位 4M

```
[root@server0 ~]# vgdisplay myvg
[root@server0 ~]# vgchange -s 1M myvg #修改 PE 的大小
[root@server0 ~]# vgdisplay myvg
[root@server0 ~]# lvcreate -L 250M -n lvtest01 myvg
[root@server0 ~]# lvs
```

命令格式: `lvcreate -l PE 的个数 -n 逻辑卷的名字 基于的卷组`

```
[root@server0 ~]# lvcreate -l 86 -n lvtest03 myvg
Logical volume "lvtest03" created
[root@server0 ~]# lvs
```

---

---

## 交换分区的管理

### 一、分区

```
[root@server0 ~]# lsblk
```

`parted` 专用于 划分大空间 用于划分 GPT 分区方案  
GPT 分区模式 支持 128 个主分区  
支持 18EB 空间

```
[root@server0 ~]# parted /dev/vdb
```

```
(parted) mktable gpt      #指定分区的模式
```

```
(parted) print            #输出分区表
```

```
(parted) mkpart          #划分新的分区
```

```
分区名称?  []? haha      #分区的名称随意指定
```

```
文件系统类型?  [ext2]?    #文件系统直接回车
```

```
起始点?  0
```

```
结束点?  2G
```

警告: The resulting partition is not properly aligned  
for best performance.

```
忽略/Ignore/放弃/Cancel? Ignore    #选择忽略
```

```
(parted) unit  GB        #用 GB 作为显示单位
```

```
(parted) quit          #退出
```

## 二、交换分区的使用

- 相当于虚拟内存,
  - 当物理内存不够用时,使用磁盘空间来模拟内存
  - 在一定程度上缓解内存不足的问题
  - 交换分区:以空闲分区充当的交换空间

### 1.查看交换空间信息

```
[root@server0 ~]# swapon -s
```

### 2.制作交换空间

```
[root@server0 ~]# mkswap /dev/vdb1    #格式化交换文件系统
```

```
[root@server0 ~]# mkswap /dev/vdb2    #格式化交换文件系统
```

```
[root@server0 ~]# swapon /dev/vdb1    #启用交换分区
```

```
[root@server0 ~]# swapon -s
```

```
[root@server0 ~]# swapon /dev/vdb2
```

```
[root@server0 ~]# swapon -s
```

```
[root@server0 ~]# swapoff /dev/vdb1    #停用交换分区
```

```
[root@server0 ~]# swapon -s
```

```
[root@server0 ~]# swapoff /dev/vdb2    #停用交换分区
```

```
[root@server0 ~]# swapon -s
```

### 3.完成开机自动启用交换空间

```
[root@server0 ~]# blkid /dev/vdb1
```

```
/dev/vdb1: UUID="07de15a7-baa9-48ee-856a-dec6f2b2c1d4" TYPE="swap"  
PARTLABEL="haha" PARTUUID="ca01c66b-1a2f-443f-bc45-ceedb6f65a312"
```

```
[root@server0 ~]# vim /etc/fstab  
UUID="07de15a7-baa9-48ee-856a-dec6f2b2c1d4"    swap                swap  
defaults 0 0
```

```
[root@server0 ~]# swapon -a  
[root@server0 ~]# swapon -s
```

---

## 附加权限

### Set UID

- 附加在属主的 **x** 位上
  - 属主的权限标识会变为 **s**
  - 适用于可执行文件, **Set UID** 可以让使用者具有文件属主的身份及部分权限
  - 传递所有者身份

```
[root@server0 ~]# parted /dev/vdb  
GNU Parted 3.1  
使用 /dev/vdb  
Welcome to GNU Parted! Type 'help' to view a list of commands.  
(parted) mktable gpt  
(parted) print  
Model: Virtio Block Device (virtblk)  
Disk /dev/vdb: 10.7GB  
Sector size (logical/physical): 512B/512B  
Partition Table: gpt  
Disk Flags:  


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

  
(parted) mkpart  
分区名称? []? haha  
文件系统类型? [ext2]?  
起始点? 0  
结束点? 2G  
警告: The resulting partition is not properly aligned for best performance.  
忽略/Ignore/放弃/Cancel?  
忽略/Ignore/放弃/Cancel? ignore  
(parted) unit  
Unit? [compact]?  
(parted) unit GB  
(parted) print  
Model: Virtio Block Device (virtblk)  
Disk /dev/vdb: 10.7GB  
Sector size (logical/physical): 512B/512B  
Partition Table: gpt  
Disk Flags:  


| Number | Start  | End    | Size   | File system | Name | 标志 |
|--------|--------|--------|--------|-------------|------|----|
| 1      | 0.00GB | 2.00GB | 2.00GB |             | haha |    |


```

```
(parted) mkpart
分区名称?  []? xixi
文件系统类型?  [ext2]?
起始点?  2G
结束点?  4G
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 10.7GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
```

Number	Start	End	Size	File system	Name	标志
1	0.00GB	2.00GB	2.00GB		haha	
2	2.00GB	4.00GB	2.00GB		xixi	

```
(parted) quit
信息: You may need to update /etc/fstab.
```

格式化交换文件系统、启用交换分区

```
[root@server0 ~]# swapon -s
[root@server0 ~]# mkswap /dev/vdb1
正在设置交换空间版本 1, 大小 = 1953104 KiB
无标签, UUID=02e6dea6-801f-4250-9dc7-a4344d189a8b
[root@server0 ~]# mkswap /dev/vdb2
正在设置交换空间版本 1, 大小 = 1952764 KiB
无标签, UUID=930fa17f-b9ec-4e4c-a6d0-c6604e3a704e
[root@server0 ~]# swapon /dev/vdb1
[root@server0 ~]# swapon /dev/vdb2
[root@server0 ~]# swapon -s
```

文件名	类型	大小	已用	权限
/dev/vdb1	partition	1953104	0	- 1
/dev/vdb2	partition	1952764	0	- 2

停用交换分区

```
[root@server0 ~]# swapoff /dev/vdb1
[root@server0 ~]# swapoff /dev/vdb2
[root@server0 ~]# swapon -s
[root@server0 ~]#
```

完成开机自动启用交换空间

```
[root@server0 ~]# vim /etc/fstab
```

```
/dev/vdb1 swap swap defaults 0 0
UUID="930fa17f-b9ec-4e4c-a6d0-c6604e3a704e" swap swap defaults
0 0
```



```
[ root@server0 ~]# swapon - s
[ root@server0 ~]# swapon - a
[ root@server0 ~]# swapon - s
```

文件名	权限	类型	大小	已用
/dev/vdb1		partition		195310
4 0	- 1			

```
[ root@server0 ~]# blkid /dev/vdb2
/dev/vdb2: UUID="930fa17f- b9ec- 4e4c- a6d0- c6604e3a704e" TYPE="s
wap" PARTLABEL="xixi" PARTUUID="8483e75f- f0f7- 4093- a4e3- b2ae8e
c28739"
[ root@server0 ~]# vim /etc/fstab
```

```
/dev/vdb1 swap swap defaults 0 0
UUID="930fa17f- b9ec- 4e4c- a6d0- c6604e3a704e" swap swap defaults
0 0
```

```
[ root@server0 ~]# swapon - a
[ root@server0 ~]# swapon - s
```

文件名	权限	类型	大小	已用
/dev/vdb1		partition		195310
4 0	- 1			
/dev/vdb2		partition		195276
4 0	- 2			