

## 逻辑卷LVM实战

### 1、LVM原理

#### (1) 磁盘管理准备

#### (2) 硬盘分区：主分区、扩展分区和逻辑分区

第一种方法：使用gparted图形工具分区，比较直观。

第二种方法：使用fdisk命令分区，使用mkfs进行格式化

### 2、LVM实战

实战1：创建卷组vg

实战2：动态扩充卷组vg的容量

实战3：创建逻辑卷并格式化，挂载到一个指定的目录

实战4：动态扩充逻辑卷

实战6：删除lvm

### 3、总结

# 逻辑卷LVM实战

## 1、LVM原理

Lvm 的概念和基本使用详解 - 知乎 (zhihu.com)

LVM可以实现用户不停机的情况下，随意调整分区的大小。

### (1) 磁盘管理准备

按照RAID0实战中的方法在虚拟机中添加两块新硬盘。这里以新硬盘的设备文件名为sdb，sdc为例进行讲解。

### (2) 硬盘分区：主分区、扩展分区和逻辑分区

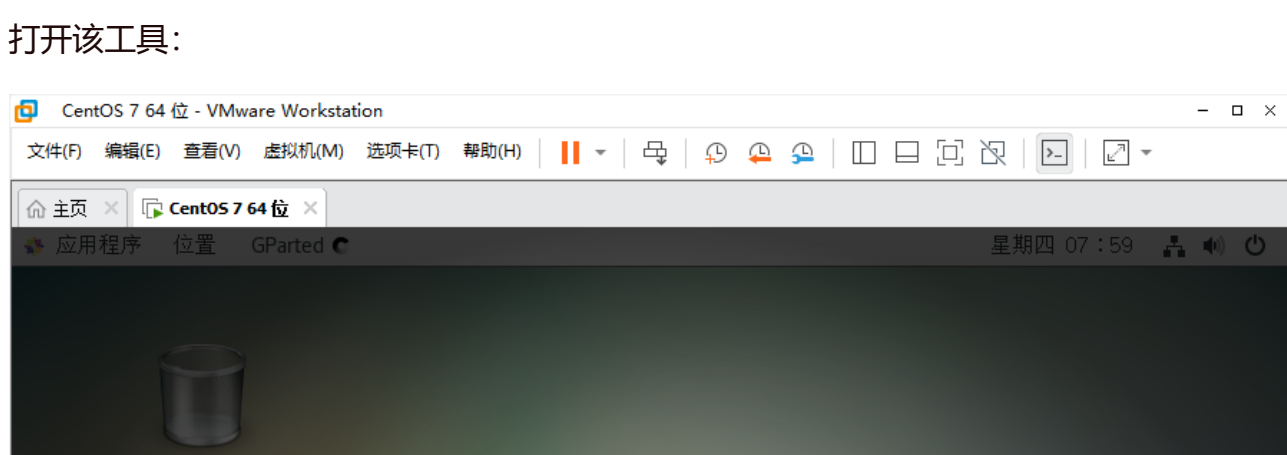
接下来，对上述两块硬盘进行分区。

提示：本次实战的环境都需要以root身份登录Linux服务器。另外要求安装虚拟机的Windows系统能够连接网络。

第一种方法：使用gparted图形工具分区，比较直观。

```
1 # 能够连接外网的情况下，
2 # 在本次实验中，虚拟机用的是NAT，所以只要Windows能连接外网，Linux就能够连接外网。
3 # 安装gparted图形分区工具。
4 [root@bogon ~]# yum install epel-release.noarch
5 [root@bogon ~]# yum install gparted
```

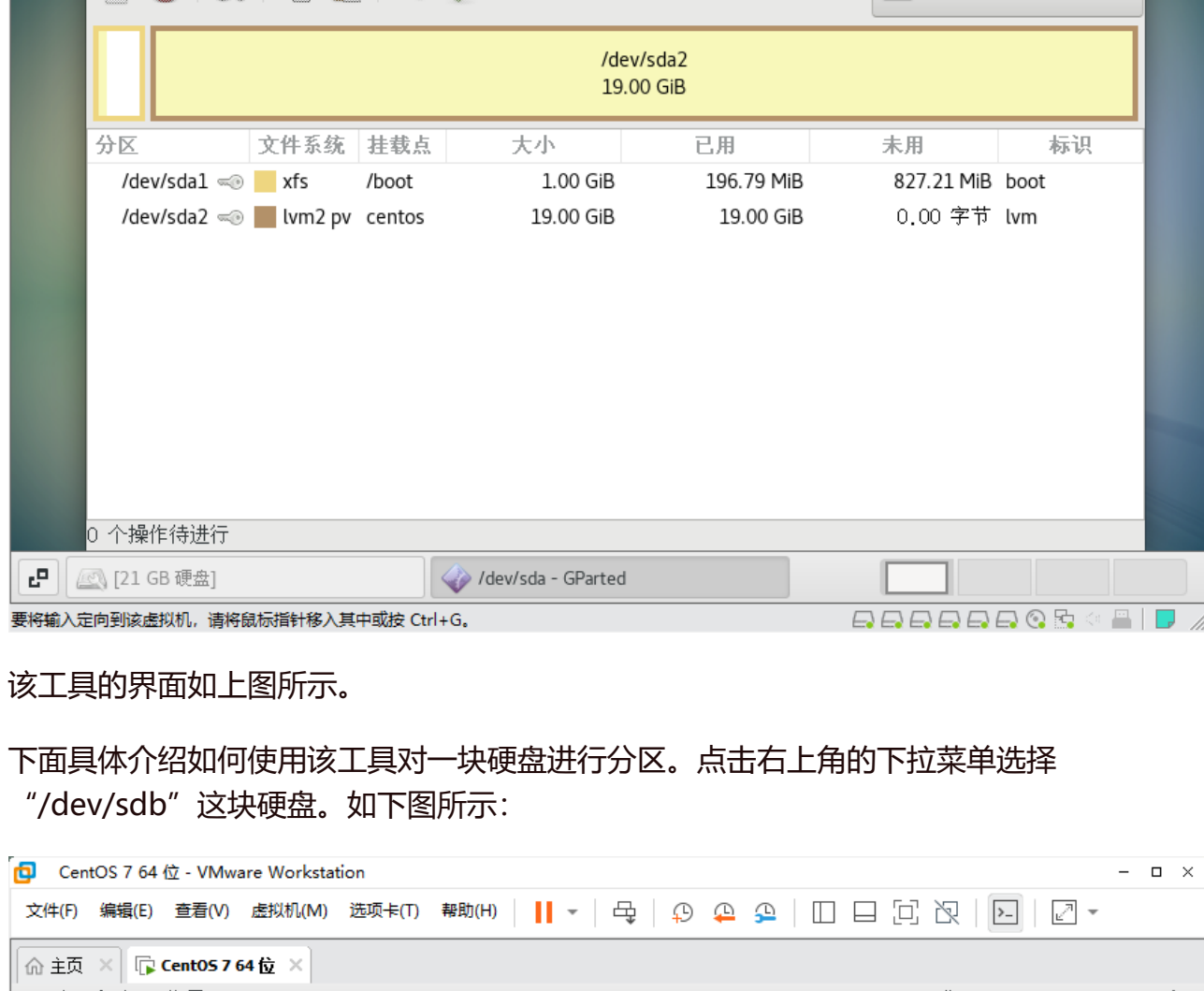
安装完成后，如下图，在“应用程序”->“系统工具”->“GParted”。



打开该工具：

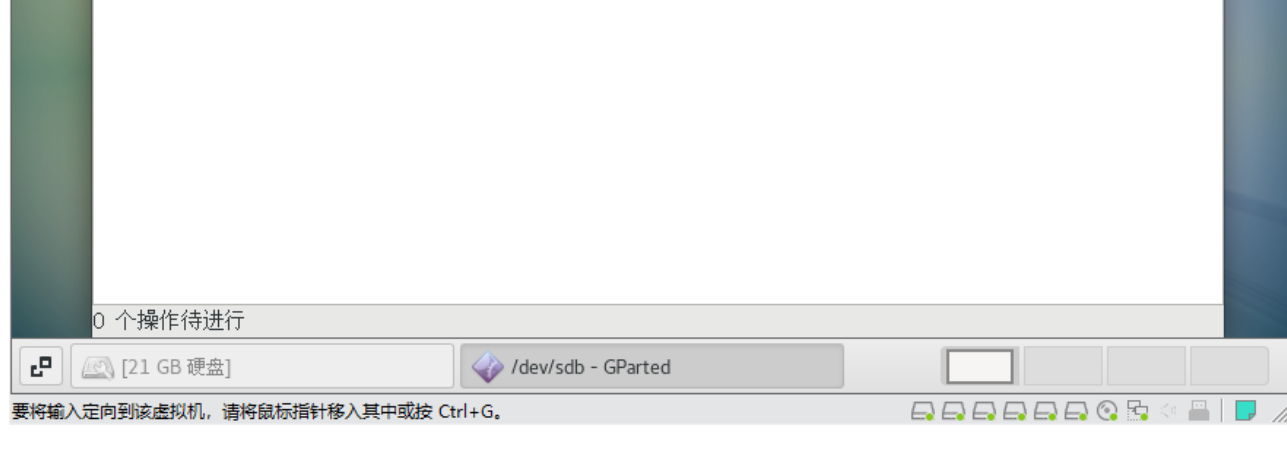


如果是普通用户，需要输入超级用户root的密码进行认证。

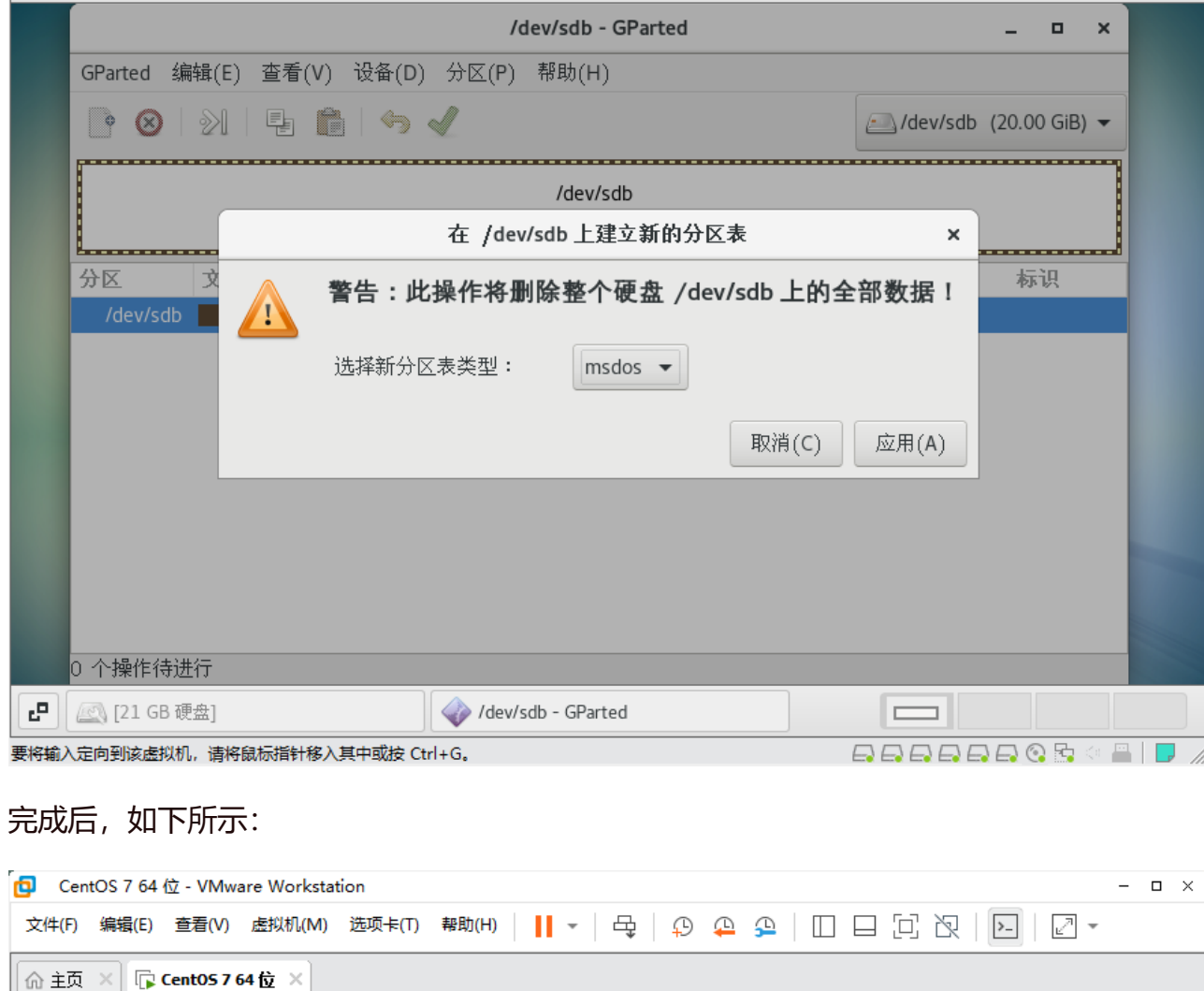


该工具的界面如上图所示。

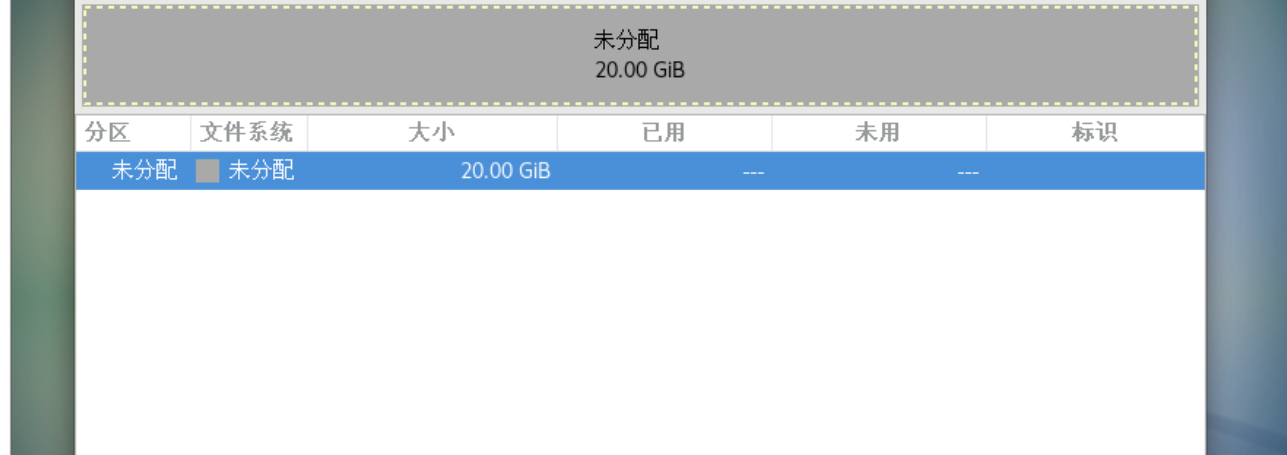
下面具体介绍如何使用该工具对一块硬盘进行分区。点击右上角的下拉菜单选择“/dev/sdb”这块硬盘。如下图所示：



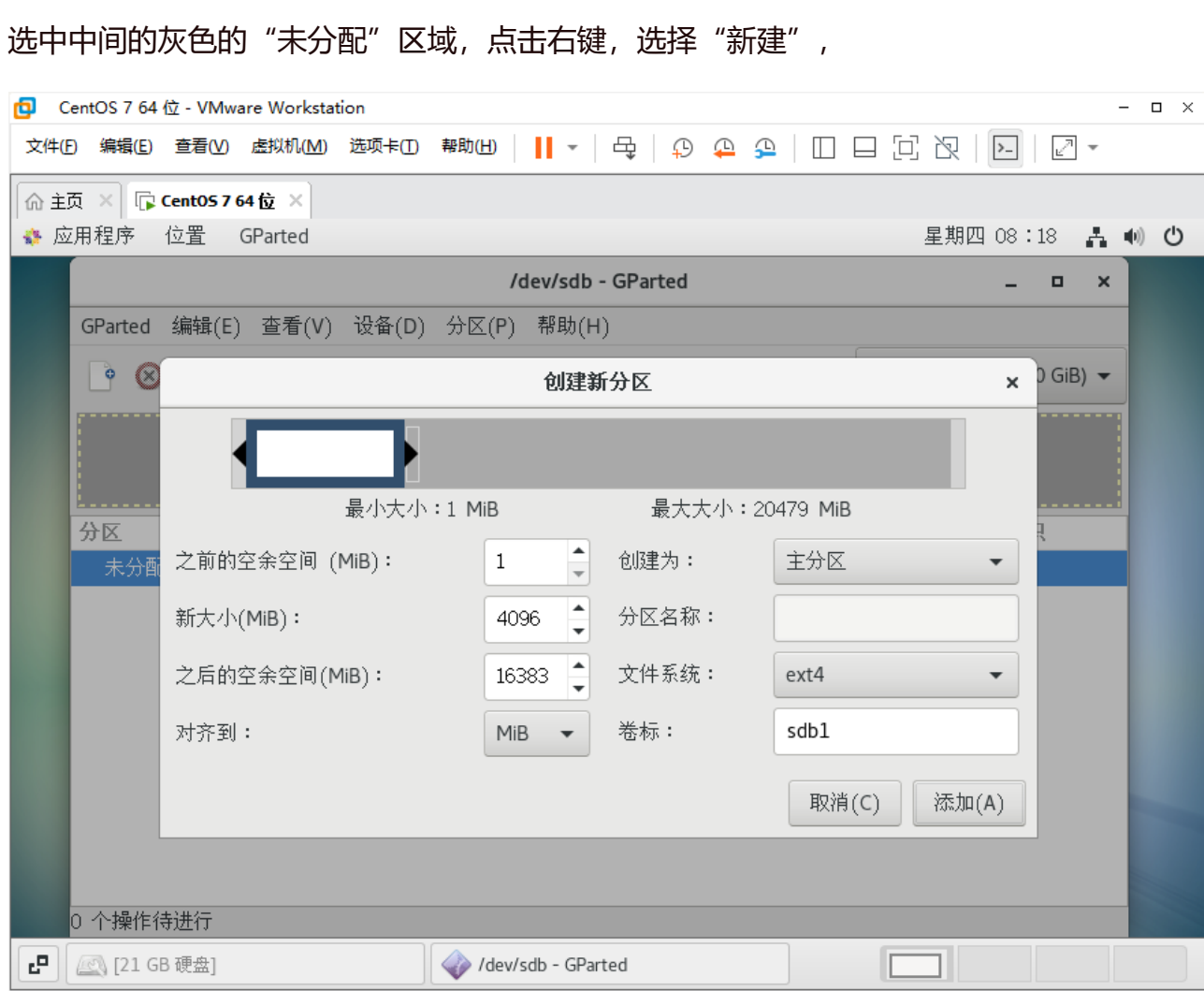
点击菜单中“设备”->“创建分区表”，如下图所示，选择新分区表类型，这里默认即可，点击“应用”。



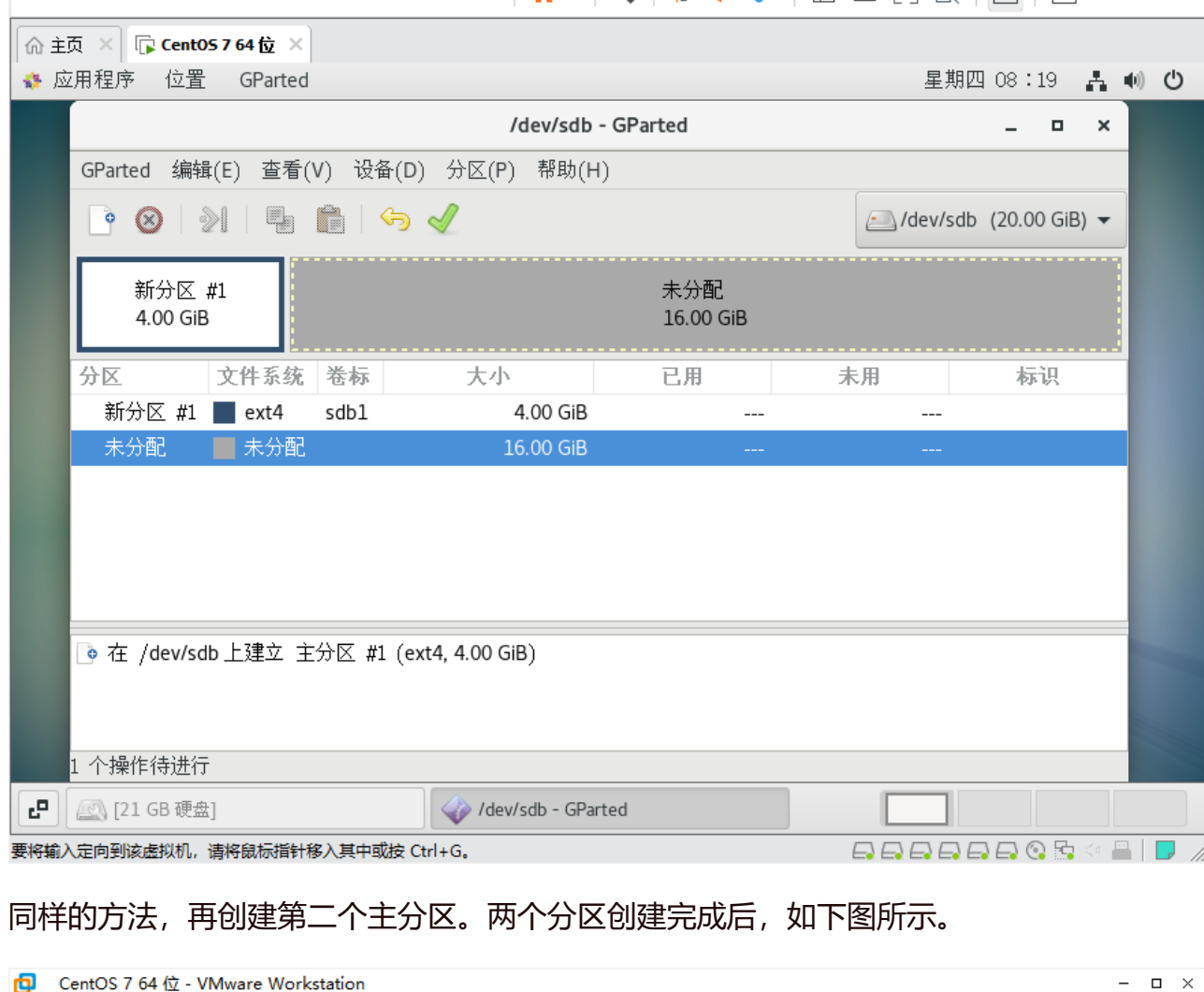
完成后，如下所示：



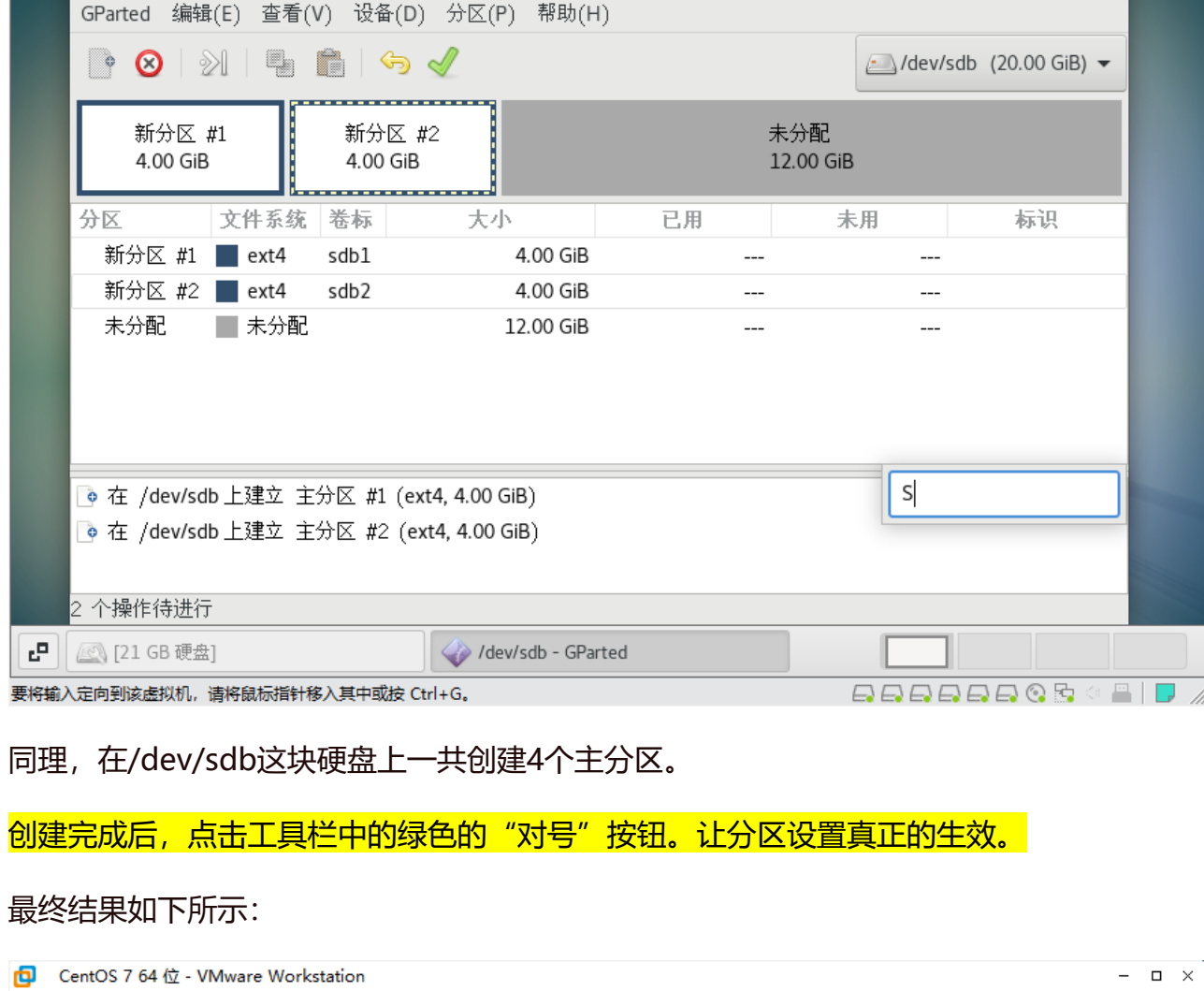
选中中间的灰色的“未分配”区域，点击右键，选择“新建”，



各个选项设置为如上图所示。点击“添加”。就可以完成了第一个主分区的创建。



同样的方法，再创建第二个主分区。两个分区创建完成后，如下图所示。



同理，在/dev/sdb这块硬盘上一共创建4个主分区。

创建完成后，点击工具栏中的绿色的“对号”按钮。让分区设置真正的生效。

最终结果如下所示：



这样就完成了4个主分区的创建。

1 # 在命令行下，确认/dev/sdb的分区情况，应该如下所示：

2 [root@bogon ~]# lsblk /dev/sdb

3 NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

4 sdb 8:16 0 20G 0 disk

5 └─sdb1 8:17 0 4G 0 part

6 └─sdb2 8:18 0 4G 0 part

7 └─sdb3 8:19 0 4G 0 part

8 └─sdb4 8:20 0 8G 0 part

9

第二种方法：使用fdisk命令分区，使用mkfs进行格式化

这两个命令的具体用法，可以参考如下的文章：

【linux】循序渐进学运维-基础篇-分区命令fdisk - 知乎 (zhihu.com)

陈涛•笨办法学Linux-磁盘分区哔哩哔哩bilibili



## 2、LVM实战

### 实战1：创建卷组vg

- 首先创建物理卷pv

```
1 #-----
2 #实战1：创建卷组
3 #-----
4
5 # 第一步，在上面的准备工作中，已经完成了第一步，添加了新硬盘sdb，分了四个分区
   sdb1, sdb2, sdb3, sdb4。
6 [root@bogon ~]# lsblk /dev/sdb
7      NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
8      sdb         8:16   0   20G  0 disk
9      └─sdb1      8:17   0    4G  0 part
10     └─sdb2      8:18   0    4G  0 part
11     └─sdb3      8:19   0    4G  0 part
12     └─sdb4      8:20   0    8G  0 part
13
14
15 #第二步，创建物理卷PV
16 [root@bogon ~]# pvcreate /dev/sdb1 /dev/sdb2
17 WARNING: ext4 signature detected on /dev/sdb1 at offset 1080. Wipe it?
18 [y/n]: y
19 Wiping ext4 signature on /dev/sdb1.
20 WARNING: ext4 signature detected on /dev/sdb2 at offset 1080. Wipe it?
21 [y/n]: y
22 Wiping ext4 signature on /dev/sdb2.
23 Physical volume "/dev/sdb1" successfully created.
24 Physical volume "/dev/sdb2" successfully created.
25
26 # 第三步，查看已经创建的物理卷，应该可以看到如下所示的信息。
27 [root@bogon ~]# pvdisplay /dev/sdb1
28      "/dev/sdb1" is a new physical volume of "4.00 GiB"
29      --- NEW Physical volume ---
30      PV Name           /dev/sdb1
31      VG Name
32      PV Size            4.00 GiB
33      Allocatable        NO
34      PE Size            0
35      Total PE           0
36      Free PE            0
37      Allocated PE       0
38      PV UUID            v3j5vh-CZZp-8F6h-s2tP-3QTV-Q0TR-sGahad
39
40 [root@bogon ~]# pvdisplay /dev/sdb2
41      "/dev/sdb2" is a new physical volume of "4.00 GiB"
42      --- NEW Physical volume ---
43      PV Name           /dev/sdb2
44      VG Name
45      PV Size            4.00 GiB
46      Allocatable        NO
47      PE Size            0
48      Total PE           0
49      Free PE            0
50      Allocated PE       0
51      PV UUID            BS4ZF6-vMvy-2yZo-dFry-cBz9-bFFz-G0wifz
```

- 创建卷组vg

```
1 #第四步，创建卷组VG
2 [root@bogon ~]# vgcreate testVG /dev/sdb1 /dev/sdb2
3 Volume group "testVG" successfully created
4
5 #查看卷组信息，可以看到整个卷组的容量=sdb1+sdb2=8G
6 [root@bogon ~]# vgdisplay testVG
7      --- Volume group ---
8      VG Name           testVG
9      System ID
10     Format              lvm2
11     Metadata Areas      2
12     Metadata Sequence No 1
13     VG Access           read/write
14     VG Status            resizable
15     MAX LV              0
16     Cur LV              0
17     Open LV             0
18     Max PV              0
19     Cur PV              2
20     Act PV              2
21     VG Size              7.99 GiB
22     PE Size              4.00 MiB
23     Total PE            2046
24     Alloc PE / Size     0 / 0
25     Free PE / Size      2046 / 7.99 GiB
26     VG UUID              byVbZ2-p0h4-etQK-ue39-kt3x-Fx3E-sRng3h
27
```

至此，已经卷组testVG已经创建成功了。

### 实战2：动态扩充卷组vg的容量

在服务器的实用过程中，经常遇到，随着数据的增多。当初规划的卷组容量不够的情况。

那么，我们可以按照如下的方式对卷组的容量实施动态扩充。例如刚才创建的testVG卷组的容量为8G，现在需要对其进行扩充4G，使其容量为12G。扩充过程如下所示：

```
1 #-----
2 #实战2： 动态扩充卷组的容量
3 #-----
4 #创建一个新的PV，这里/dev/sdb3的容量为4G。
5 [root@bogon ~]# pvcreate /dev/sdb3
6 WARNING: ext4 signature detected on /dev/sdb3 at offset 1080. Wipe it?
7 [y/n]: y
8 Wiping ext4 signature on /dev/sdb3.
9 Physical volume "/dev/sdb3" successfully created.
10
11 #把新的PV添加进现有的testVG卷组，扩充容量
12 [root@bogon ~]# vgextend testVG /dev/sdb3
13 Volume group "testVG" successfully extended
14
15 #查看容量，新的容量应该是=原来卷组的容量+扩充的容量
16 [root@bogon ~]# vgdisplay testVG
17      --- Volume group ---
18      VG Name           testVG
19      System ID
20     Format              lvm2
21     Metadata Areas      3
22     Metadata Sequence No 2
23     VG Access           read/write
24     VG Status            resizable
25     MAX LV              0
26     Cur LV              0
27     Open LV             0
28     Max PV              0
29     Cur PV              3
30     Act PV              3
31     VG Size              <11.99 GiB          #这里可以看到卷组总的
32     PE Size              4.00 MiB          容量已经扩充了。
33     Total PE            3069
34     Alloc PE / Size     0 / 0
35     Free PE / Size      3069 / <11.99 GiB
36     VG UUID              byVbZ2-p0h4-etQK-ue39-kt3x-Fx3E-sRng3h
```

同样的方法，在服务器使用的过程中，可以随时按照这种方式添加新的pv，扩充卷组的大小。

那么创建出来的卷组testVG，如何去使用它呢？接下来通过下面的实战进行演示：

### 实战3：创建逻辑卷并格式化，挂载到一个指定的目录

比如，服务器想要规划一个目录/root/test\_lv，将来在该目录中存放客户端上传的文件，并且随着系统的运行，客户端存放的文件越来越多，那么应该怎么做呢？

接下来通过实战3和实战4，进行演示说明。

```
1 #-----
2 #实战3： 从卷组testVG中创建逻辑卷testlv1并格式化，挂载到一个指定的目录
3 #-----
4 # 第一步：新建一个逻辑卷testlv1，大小为2048M。
5 [root@bogon ~]# lvcreate -n testlv1 -L 2048M testVG
6 Logical volume "testlv1" created.
7
8 # 第二步：查看逻辑卷的信息
9 [root@bogon ~]# lvdisplay
10      --- Logical volume ---
11      LV Path              /dev/testVG/testlv1
12      LV Name              testlv1          # testlv1的逻辑卷
13      VG Name              testVG           # 从testVG卷组中新建
14      LV UUID              jqluQt-fdNd-rxf3-yEmA-r7NU-UEwP-LZay6n
15      LV Write Access       read/write
16      LV Creation host, time bogon, 2024-05-30 16:32:22 +0800
17      LV Status              available
18      # open                0
19      LV Size                2.00 GiB        # 大小为2048M
20      Current LE             512
21      Segments               1
22      Allocation             inherit
23      Read ahead sectors     auto
24      - currently set to    8192
25      Block device           253:2
26
27 # 第三步：格式化/dev/testVG/testlv1逻辑卷，类型为xfs文件系统
28 [root@bogon ~]# mkfs -t xfs /dev/testVG/testlv1
29 meta-data=/dev/testVG/testlv1  isize=512    agcount=4, agsize=131072
   blks
30      =                  sectsz=512   attr=2, projid32bit=1
31      =                  crcrc=1      finobt=0, sparse=0
32 data      =                  bsize=4096  blocks=524288, imaxpct=25
33      =                  sunit=0      swidth=0 blks
34 naming    =version 2              ascii-ci=0  ftype=1
35 log      =internal log          bsize=4096  blocks=2560, version=2
36      =                  sectsz=512   sunit=0 blks, lazy-count=1
37 realtime =none                extsz=4096  blocks=0, rtextents=0
38
39 # 第四步：新建一个目录/root/test_lv
40 [root@bogon ~]# mkdir /root/test_lv
41
42 # 第五步：把上面格式化好的/dev/testVG/testlv1和目录/root/test_lv挂载到一起
43 [root@bogon ~]# mount /dev/testVG/testlv1 /root/test_lv
44
45 # 第六步：查看/root/test_lv的容量，其实也就是/dev/testVG/testlv1逻辑卷的容量。
46 [root@bogon ~]# df -h /root/test_lv
47 文件系统              容量  已用  可用  已用% 挂载点
48 /dev/mapper/testVG-testlv1  2.0G  33M   2.0G    2% /root/test_lv
```

到这里，我们把新建的逻辑卷和目录/root/test\_lv挂载后，就可以作为服务器的一个存储区域来使用，例如，该目录规划为客户端上传文件的使用，但是随着客户端上传的文件越来越多，2G的容量不够用，怎么办呢？

LVM的机制的最大的好处就是可以支持在线的动态的扩充容量，而且不影响原有的服务器上的业务和数据。

下面的实战4，来说明如何动态的扩充逻辑卷的容量。

### 实战4：动态扩充逻辑卷

```
1 #-----
2 #实战4： 动态扩充逻辑卷
3 #-----
4 #LVM支持在线对逻辑卷进行扩充，而且在线扩充不需要卸载正在使用的文件系统
5 #第一步：查看vg还有多少容量没有分配
6 [root@bogon ~]# vgdisplay testVG
7      --- Volume group ---
8      VG Name           testVG
9      ...
10     Alloc PE / Size     512 / 2.00 GiB   # 已经分配的空间
11     Free PE / Size      2557 / <9.99 GiB # 剩余的可以分配的空间
12
13
14 #第二步：扩充逻辑卷，由2G扩充到了2G+4G=6G
15 [root@bogon ~]# lvextend -L +4096M /dev/testVG/testlv1
16 Size of logical volume testVG/testlv1 changed from 2.00 GiB (512
   extents) to 6.00 GiB (1536 extents).
17 Logical volume testVG/testlv1 successfully resized.
18
19 #逻辑卷的容量扩充以后，还需要对扩展该逻辑卷的文件系统，如果是EXT类型的文件系统，使用
20 #resize2fs命令，XFS类型的文件系统使用xfs_growfs命令。
21 #第三步，扩充逻辑卷的文件系统
22 [root@bogon ~]# xfs_growfs /dev/testVG/testlv1
23 meta-data=/dev/mapper/testVG-testlv1  isize=512    agcount=4, agsize=131072
   blks
24      ...
25 data blocks changed from 524288 to 1572864
26
27 #第四步，查看扩充后的逻辑卷
28 [root@bogon ~]# lvdisplay
29      --- Logical volume ---
30      LV Path              /dev/testVG/testlv1
31      LV Name              testlv1
32      VG Name              testVG
33      ...
34      LV Size              6.00 GiB        #可以看出已经扩充为6G
35
36 # 查看/root/test_lv/, 该目录的容量现在是6G，扩充成功。
37 [root@bogon ~]# df -h /root/test_lv/
38 文件系统              容量  已用  可用  已用% 挂载点
39 /dev/mapper/testVG-testlv1  6.0G   33M   6.0G    1% /root/test_lv
```

按照此方法，可以随时扩充容量，更重要的是在扩充容量的时候，不需要停止服务器业务的运营。非常方便。

### 实战6：删除lvm

如果不想使用LVM的方式进行磁盘管理了。那么可以删除LVM，在删除的过程中，一定要遵循下面的顺序。

```
1 #-----
2 #实战6： 删除lvm
3 #-----
4 #首先查看目前的lv挂载到哪个目录
5 [root@localhost ~]# df -h
6 /dev/mapper/testVG-testlv1  6.0G   33M   6.0G    1% /root/test_lv  # 可以
   看到目前的逻辑卷挂载到了/root/test_lv目录
7
8 # 第一步：首先卸载LV
9 [root@bogon ~]# umount /root/test_lv
10
11 # 第二步：删除逻辑卷
12 [root@bogon ~]# lvremove /dev/testVG/testlv1
13 Do you really want to remove active logical volume testVG/testlv1? [y/n]:
14 y
15 Logical volume "testlv1" successfully removed
16
17 # 第三步：删除卷组
18 [root@bogon ~]# vgremove testVG
19 Volume group "testVG" successfully removed
20
21 # 第四步：删除物理卷
22 [root@bogon ~]# pvremove /dev/sdb1 /dev/sdb2 /dev/sdb3
23 Labels on physical volume "/dev/sdb1" successfully wiped.
24 Labels on physical volume "/dev/sdb2" successfully wiped.
25 Labels on physical volume "/dev/sdb3" successfully wiped.
26
27 #lvm都删除完毕后，使用lsblk命令确认一下，可以看到所有的磁盘已经恢复到没有建立逻辑
   卷之前的状态。
28 [root@bogon ~]# lsblk
29      NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
30      sda         8:0   0   20G  0 disk
31      └─sda1      8:1   0    1G  0 part /boot
32      └─sda2      8:2   0   19G  0 part
33      └─centos-root 253:0   0   17G  0 lvm  /
34      └─centos-swap 253:1   0    2G  0 lvm  [SWAP]
35      sdb         8:16   0   20G  0 disk
```

36	—sdb1	8:17	0	4G	0	part
37	—sdb2	8:18	0	4G	0	part
38	—sdb3	8:19	0	4G	0	part
39	—sdb4	8:20	0	8G	0	part
40	sdc	8:32	0	20G	0	disk
41	sdd	8:48	0	20G	0	disk
42	sde	8:64	0	20G	0	disk
43	sdf	8:80	0	20G	0	disk
44	sr0	11:0	1	9.6G	0	rom /run/media/lxr/CentOS 7 x86_64
45						

### 3、总结

- 建立逻辑卷过程如下：
  - (1) 添加硬盘，并分区
  - (2) pvcreate 创建物理卷pv
  - (3) vgcreate 创建卷组vg
  - (4) lvcreate 创建逻辑卷lv
  - (5) 格式化lv，并挂载到指定的目录即可使用
- 删除逻辑卷的过程相反：
  - (1) 卸载逻辑卷lv
  - (2) 删除逻辑卷lvremove
  - (3) 删除卷组vgremove
  - (4) 删除物理卷pvcreate
- 扩充卷组使用vgextend
- 扩充逻辑卷使用lvextend
- 提示：扩充逻辑卷后，还需要扩充逻辑卷的文件系统。（如果是ext类型的文件系统，使用resize2fs命令，xfs类型的文件系统使用xfs\_growfs命令。）

#### 逻辑卷LVM实战

- 1、LVM原理
  - (1) 磁盘管理准备
  - (2) 硬盘分区：主分区、扩展分区和逻辑分区
    - 第一种方法：使用gparted图形工具分区，比较直观。
    - 第二种方法：使用fdisk命令分区，使用mkfs进行格式化
- 2、LVM实战
  - 实战1：创建卷组vg
  - 实战2：动态扩充卷组vg的容量
  - 实战3：创建逻辑卷并格式化，挂载到一个指定的目录
  - 实战4：动态扩充逻辑卷
  - 实战6：删除lv
- 3、总结