1、在虚拟机上添加两块硬盘，容量任意，登录普通用户 user1、user2、user3 分别执行磁盘分区，格式化以及挂载任务，从标准输出中观察理解并记录结果

在虚拟机上添加两块硬盘，如图 1-1 所示

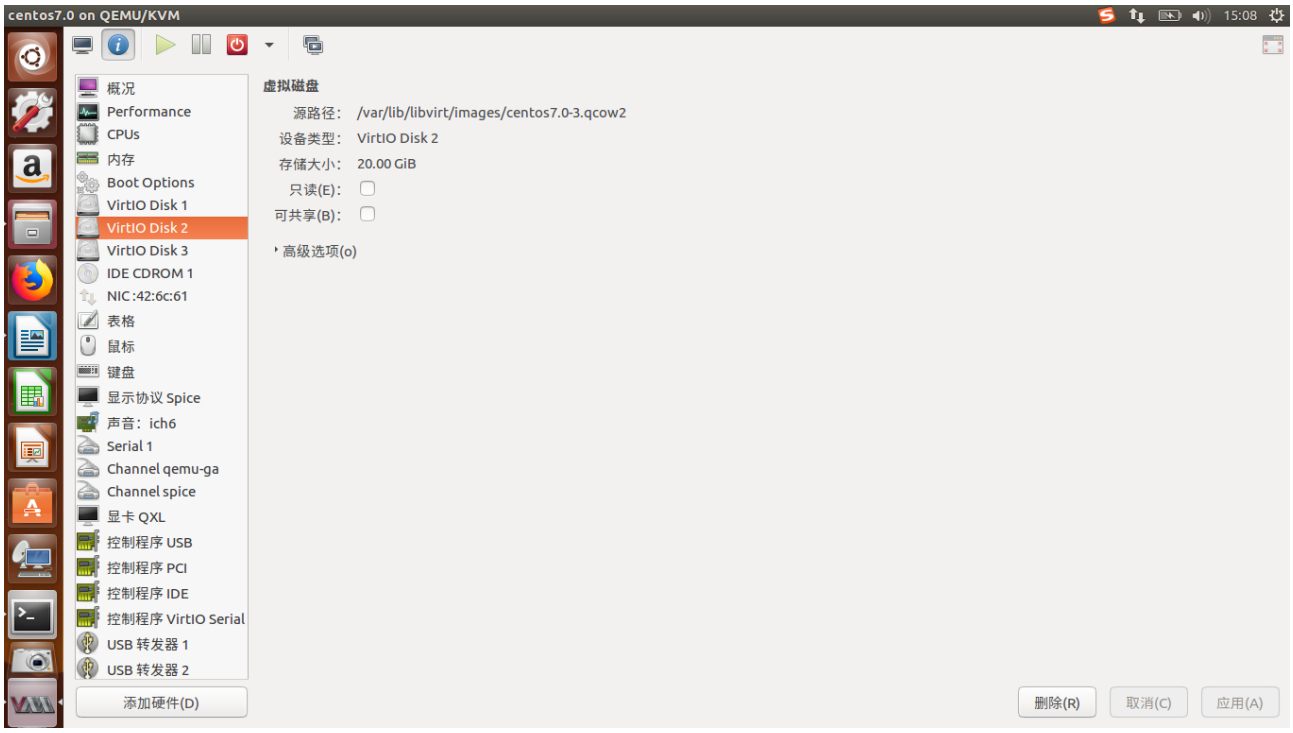

图 1-1

登录普通用户 user1 执行磁盘分区，如图 1-2

```
[user1@localhost ~]$ fdisk /dev/vdb
fdisk: cannot open /dev/vdb: Permission denied
```

图 1-2

登录普通用户 user2 执行格式化，如图 1-3

```
[user2@localhost ~]$ mkfs.xfs /dev/vdb
mkfs.xfs: cannot open /dev/vdb: Permission denied
```

图 1-3

登录普通用户 user3 执行挂载，如图 1-4

```
[user3@localhost ~]$ mount /dev/vdc /temp/msods
mount: only root can do that
```

图 1-4

因为都是普通用户登录，所以无权限执行分区、格式化、挂载。

2.现将基础磁盘管理任务分为：分区、格式化、挂载三项，分别分配给用户 user1、user2、user3 去做，每个用户只能做一项不得越权

每个用户只能做一项不得越权，user1 分区，user2 格式化，user3 挂载。打开 sudo 的配置文件分别给 user1，user2，user3 进行配置。如图 2-1

```
user1    ALL=/sbin/fdisk
user2    ALL=/sbin/mkfs.ext4, /sbin/mkfs.xfs
user3    ALL=/bin/mount
```

图 2-1

3、磁盘一分区要求：采用 msdos 分区表、划分 6 个分区、第一个分区需包含引导标识、第一分区格式化成 ext4、剩余分区格式化成 xfs、在根目录下创建公共目录 temp 并在其下创建 msdos1-msdos6 六个子目录、将六个分区依次挂载，观察记录操作过程和结果

现在对磁盘一进行分区，因为普通用户 user1 没有权限执行分区，所以加上 sudo 可以提权，如图 3-1。

```
[user1@localhost ~]$ sudo fdisk /dev/vdb
[sudo] password for user1:
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write the
m.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0x2d854bbc.

Command (m for help):
```

图 3-1

采用 msdos 分区表划分 6 个分区，下图是 3 个主分区，如图 3-2。

```
Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-41943039, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-41943039, default 41943039): +100M
Partition 1 of type Linux and of size 100 MiB is set

Command (m for help): n
Partition type:
   p   primary (1 primary, 0 extended, 3 free)
   e   extended
Select (default p): p
Partition number (2-4, default 2):
First sector (206848-41943039, default 206848):
Using default value 206848
Last sector, +sectors or +size{K,M,G} (206848-41943039, default 41943039): +100M
Partition 2 of type Linux and of size 100 MiB is set

Command (m for help): n
Partition type:
   p   primary (2 primary, 0 extended, 2 free)
   e   extended
Select (default p): p
Partition number (3,4, default 3):
```

图 3-2

现在创建第 4 个分区，第 4 个分区是扩展分区，逻辑分区是基于扩展分区创建的，所以除主分区占用的空间，剩余的空间全给扩展分区，如图 3-3。

```
Command (m for help): n
Partition type:
   p   primary (3 primary, 0 extended, 1 free)
   e   extended
Select (default e):
Using default response e
Selected partition 4
First sector (616448-41943039, default 616448):
Using default value 616448
Last sector, +sectors or +size{K,M,G} (616448-41943039, default 4
1943039):
Using default value 41943039
Partition 4 of type Extended and of size 19.7 GiB is set
```

图 3-3

现在创建第 5 个分区，大小 100M，命令的第二行提示所有的主分区已经使用，第三行提示创建逻辑分区 5，如图 3-4。（剩余的两个逻辑分区同逻辑分区 5）

```
Command (m for help): n
All primary partitions are in use
Adding logical partition 5
First sector (618496-41943039, default 618496):
Using default value 618496
Last sector, +sectors or +size{K,M,G} (618496-41943039, default 4
1943039): +100M
Partition 5 of type Linux and of size 100 MiB is set
```

图 3-4

创建好这 7 个分区以后保存退出，输入命令 sudo fdisk -l 验证是否采用的 msdos 分区表并且分区成功，如图 3-5。

```
Disk /dev/vdb: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x2d854bbc

   Device Boot      Start         End      Blocks   Id  System
/dev/vdb1            2048      206847      102400   83  Linux
/dev/vdb2          206848      411647      102400   83  Linux
/dev/vdb3          411648      616447      102400   83  Linux
/dev/vdb4          616448    41943039    20663296    5  Extended
/dev/vdb5          618496      823295      102400   83  Linux
/dev/vdb6          825344     1030143      102400   83  Linux
/dev/vdb7         1032192     1236991      102400   83  Linux
```

图 3-5

第一个分区需包含引导标识，如图 3-6。

```
Disk /dev/vdb: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x2d854bbc

   Device Boot      Start         End      Blocks   Id  System
/dev/vdb1   *        2048      206847      102400   83  Linux
/dev/vdb2          206848      411647      102400   83  Linux
/dev/vdb3          411648      616447      102400   83  Linux
/dev/vdb4          616448    41943039    20663296    5  Extended
/dev/vdb5          618496      823295      102400   83  Linux
/dev/vdb6          825344     1030143      102400   83  Linux
/dev/vdb7         1032192     1236991      102400   83  Linux
```

图 3-6

第 1 分区格式化成 ext4，如图 3-7。

```
[user2@localhost ~]$ sudo mkfs.ext4 /dev/vdb1
[sudo] password for user2:
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
25688 inodes, 102400 blocks
5120 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=33685504
13 block groups
8192 blocks per group, 8192 fragments per group
1976 inodes per group
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

图 3-7

剩余分区格式化成 xfs，如图 3-8。

```
[user2@localhost ~]$ sudo mkfs.xfs /dev/vdb2
[sudo] password for user2:
meta-data=/dev/vdb2              isize=512    agcount=4, agsize=6400 blk
s
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=25600, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=855, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

图 3-8

在根目录下创建公共目录 temp 并在其下依次创建 msdos1～msdos6 六个子目录，如图 3-9。

```
[root@localhost temp]# mkdir msdos1
[root@localhost temp]# mkdir msdos2
[root@localhost temp]# mkdir msdos3
[root@localhost temp]# mkdir msdos4
[root@localhost temp]# mkdir msdos5
[root@localhost temp]# mkdir msdos6
[root@localhost temp]# mkdir msdos7
[root@localhost temp]# cd /
[root@localhost /]# ls
bin   d1   etc   lib    media  opt   root  sbin  sys   tmp  var
boot  dev  home  lib64  mnt    proc  run   srv   temp  usr
[root@localhost /]# ls temp
msdos1  msdos2  msdos3  msdos4  msdos5  msdos6  msdos7
```

图 3-9

将六个分区依次挂载，并查看挂载点，如图 3-10。

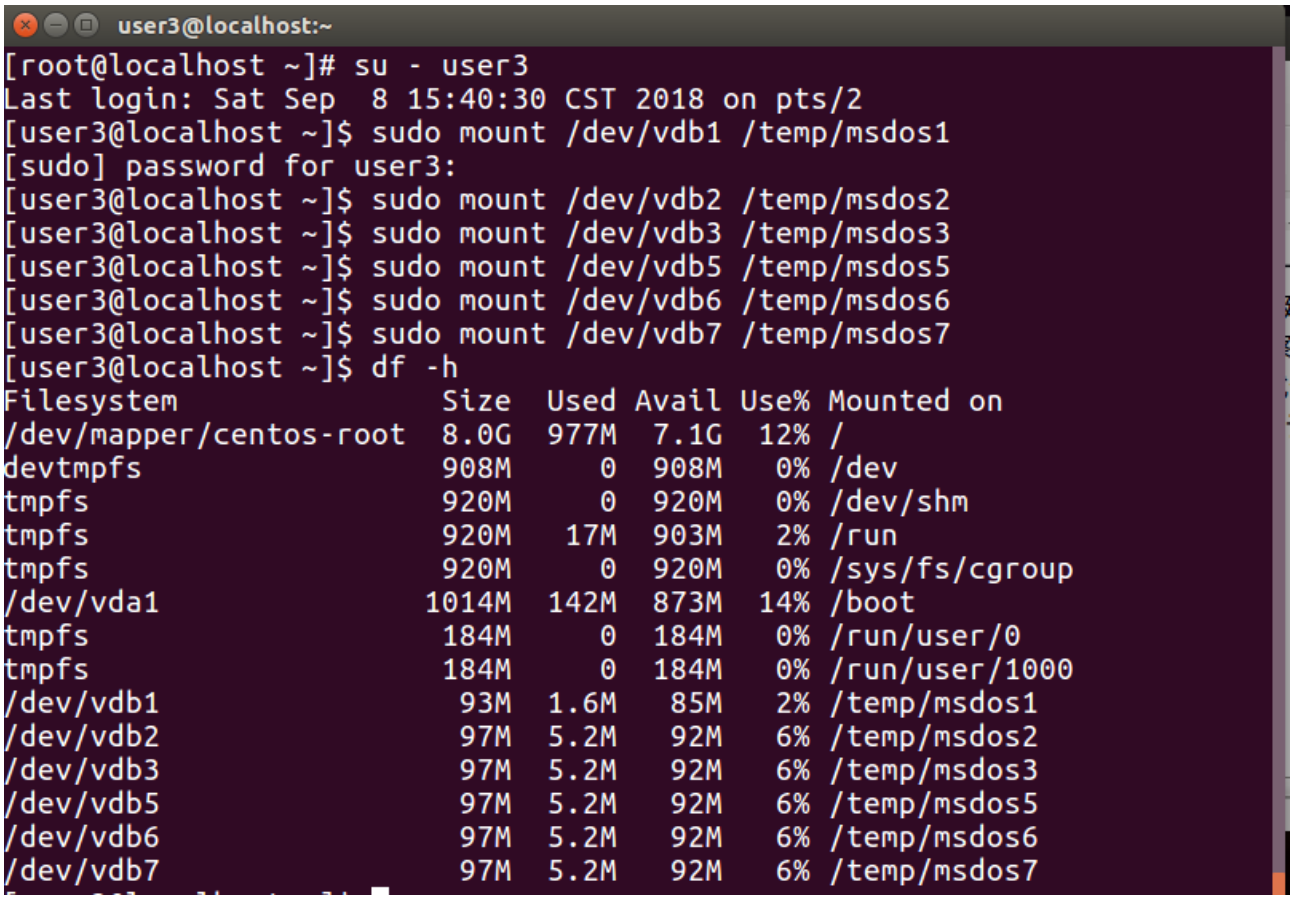

图 3-10

4、磁盘二分区要求：采用 gpt 分区表、划分 6 个分区、都格式化为 xfs、在上题所建公共目录下继续创建子目录 gpt1-gpt6、将六个分区依次挂载，观察记录操作过程和结果；

磁盘二分区要求采用 GPT 分区表，划分 6 个分区，如图 4-1。（剩余四个分区就不列示）

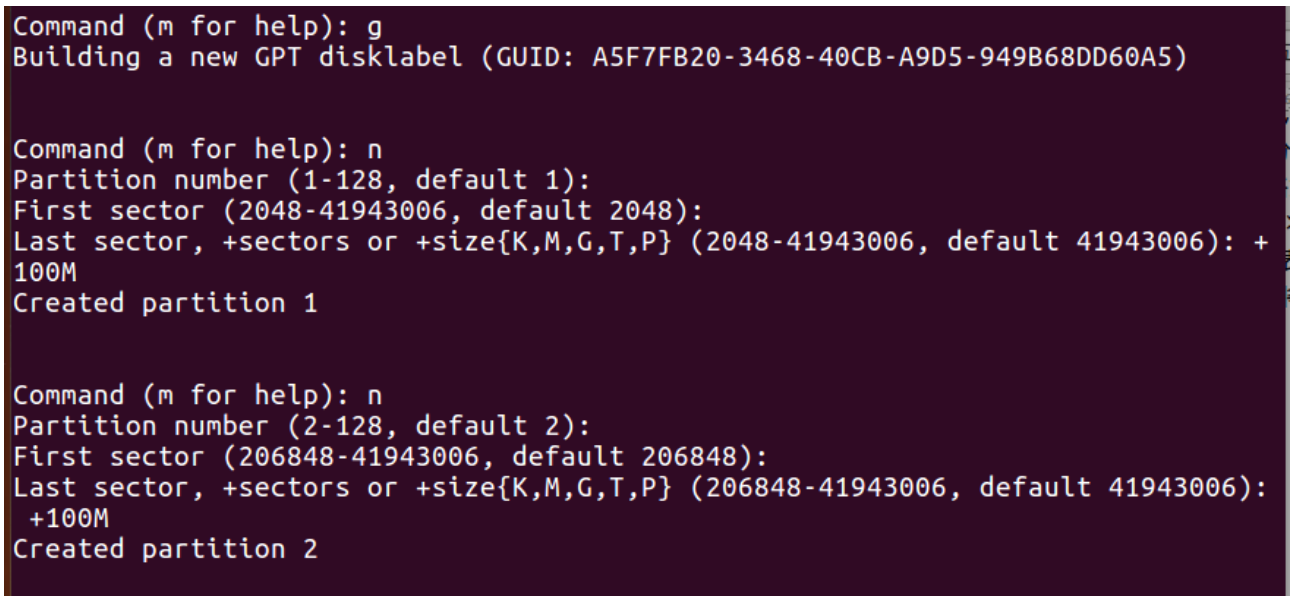

图 4-1

验证下是否是 GPT 分区表并且已经创建 6 个分区，如图 4-2。

```
Disk /dev/vdc: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: gpt
Disk identifier: A5F7FB20-3468-40CB-A9D5-949B68DD60A5


#         Start          End    Size  Type            Name
 1         2048       206847    100M  Linux filesyste
 2       206848       411647    100M  Linux filesyste
 3       411648       616447    100M  Linux filesyste
 4       616448       821247    100M  Linux filesyste
 5       821248      1026047    100M  Linux filesyste
 6      1026048      1230847    100M  Linux filesyste
```

图 4-2

这 6 个分区都格式化为 xfs，如图 4-3。

```
[user1@localhost ~]$ su - user2
Password:
Last login: Sat Sep  8 16:41:57 CST 2018 on pts/2
[user2@localhost ~]$ sudo mkfs.xfs /dev/vdc1
[sudo] password for user2:
meta-data=/dev/vdc1              isize=512    agcount=4, agsize=6400 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=25600, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=855, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[user2@localhost ~]$ sudo mkfs.xfs /dev/vdc2
meta-data=/dev/vdc2              isize=512    agcount=4, agsize=6400 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=25600, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=855, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

图 4-3

在上题所建公共目录 temp 下继续创建子目录 gpt1～gpt6，如图 4-4。

```
[root@localhost ~]# cd /
[root@localhost /]# ls temp
msdos1  msdos2  msdos3  msdos4  msdos5  msdos6  msdos7
[root@localhost /]# cd /temp
[root@localhost temp]# mkdir gpt1
[root@localhost temp]# mkdir gpt2
[root@localhost temp]# mkdir gpt3
[root@localhost temp]# mkdir gpt4
[root@localhost temp]# mkdir gpt5
[root@localhost temp]# mkdir gpt6
[root@localhost temp]# cd /
[root@localhost /]# ls temp
gpt1  gpt3  gpt5  msdos1  msdos3  msdos5  msdos7
gpt2  gpt4  gpt6  msdos2  msdos4  msdos6
```

图 4-4

将 6 个分区依次挂载，如图 4-5。

```
[root@localhost ~]# su - user3
Last login: Sat Sep  8 17:01:50 CST 2018 on pts/2
[user3@localhost ~]$ sudo mount /dev/vdc1 /temp/gpt1
[sudo] password for user3:
[user3@localhost ~]$ sudo mount /dev/vdc2 /temp/gpt2
[user3@localhost ~]$ sudo mount /dev/vdc3 /temp/gpt3
[user3@localhost ~]$ sudo mount /dev/vdc4 /temp/gpt4
```

图 4-5

查看是否挂载，如图 4-6。

```
[user3@localhost ~]$ df -h
Filesystem               Size  Used Avail Use% Mounted on
/dev/mapper/centos-root  8.0G  977M  7.1G  12% /
devtmpfs                 908M     0  908M   0% /dev
tmpfs                    920M     0  920M   0% /dev/shm
tmpfs                    920M   17M  903M   2% /run
tmpfs                    920M     0  920M   0% /sys/fs/cgroup
/dev/vda1               1014M  142M  873M  14% /boot
tmpfs                    184M     0  184M   0% /run/user/0
tmpfs                    184M     0  184M   0% /run/user/1000
/dev/vdb1                 93M  1.6M   85M   2% /temp/msdos1
/dev/vdb2                 97M  5.2M   92M   6% /temp/msdos2
/dev/vdb3                 97M  5.2M   92M   6% /temp/msdos3
/dev/vdb5                 97M  5.2M   92M   6% /temp/msdos5
/dev/vdb6                 97M  5.2M   92M   6% /temp/msdos6
/dev/vdb7                 97M  5.2M   92M   6% /temp/msdos7
/dev/vdc1                 97M  5.2M   92M   6% /temp/gpt1
/dev/vdc2                 97M  5.2M   92M   6% /temp/gpt2
/dev/vdc3                 97M  5.2M   92M   6% /temp/gpt3
/dev/vdc4                 97M  5.2M   92M   6% /temp/gpt4
/dev/vdc5                 97M  5.2M   92M   6% /temp/gpt5
/dev/vdc6                 97M  5.2M   92M   6% /temp/gpt6
```

图 4-6