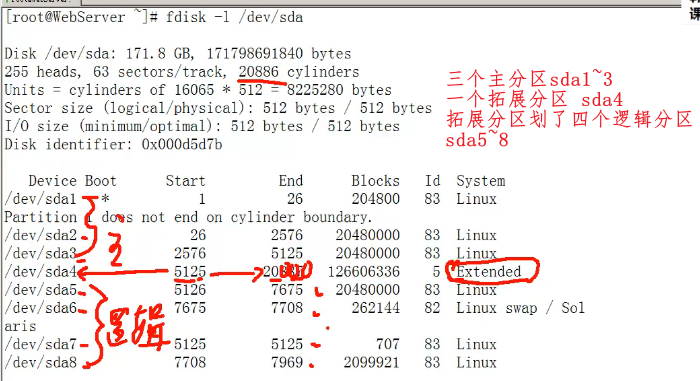


分区的边界使用柱面来标识

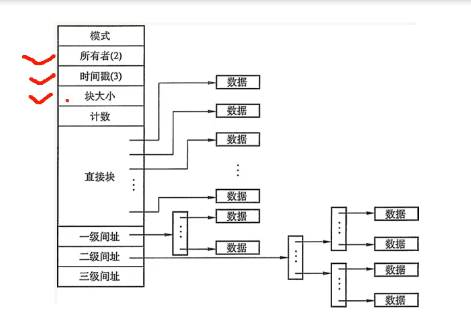


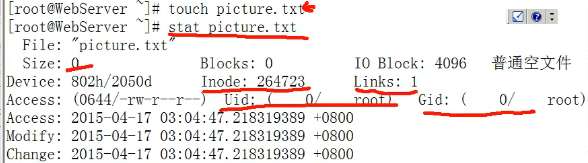
高级格式化：创建文件系统

低级格式化：划分磁道创建扇区，厂家进行的



inode df -i





文件系统在储存文件时，会根据文件大小来划定块大小，所以文件大小必定会小于等于占用空间大小。linux与windows逻辑一致



[04 Linux文件夹和文件名\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1RW4y1H7Tt?spm_id_from=333.788.player.switch&vd_source=08c34b0f73c4926b3009b7bcf284e2d1&p=4)

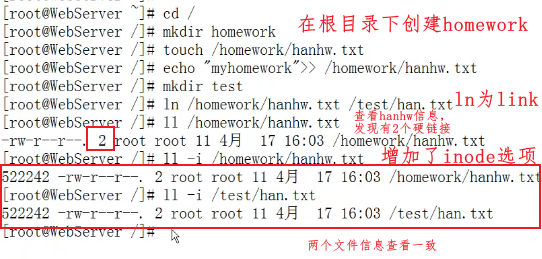
文件夹也是文件，会占用磁盘块，也有inode编号

查找文件会在各个目录的磁盘块dentry表里以文件路径和文件名为索引进行反复查找，

window和linux对文件路径也是有限制的



[05 硬连接和软连接OK\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1RW4y1H7Tt?spm_id_from=333.788.videopod.episodes&vd_source=08c34b0f73c4926b3009b7bcf284e2d1&p=5)



[06 设备文件OK\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1RW4y1H7Tt?spm_id_from=333.788.player.switch&vd_source=08c34b0f73c4926b3009b7bcf284e2d1&p=6)

b表示块设备，c表示字符设备，l是link链接的意思，dev下设备不占用磁盘大小，只是在文件系统的inode表里填入了设备信息

mknod 命令在 Linux 和 Unix 系统中并没有一个特定的英文全称，它本身就是一个命令名称。mknod 命令用于创建一个指定的文件，该文件可以是一个特殊文件，如字符设备、块设备或 FIFO（命名管道）。这个命令的名称直接来源于它所执行的操作：make a node（创建一个节点）。在 Unix 和类 Unix 系统中，文件系统中的每个文件和目录都被表示为一个节点（inode），因此 mknod 就是用来创建这些特殊类型的节点的命令。

`mknod` 是一个在 Unix 和类 Unix 操作系统中用来创建特殊文件的命令，这些特殊文件可以是字符设备文件、块设备文件或 FIFO（先进先出队列，也称为命名管道）。

### 使用方法

`mknod` 命令的基本语法如下：

```bash

mknod 文件名 类型 [参数]

```

其中：

- `文件名`：是你想要创建的特殊文件的名称。

- `类型`：可以是 `c`（字符设备）、`b`（块设备）或 `p`（FIFO）。

- `参数`：根据类型不同，可以是主设备号和次设备号（对于字符和块设备），或者是没有参数（对于 FIFO）。

### 创建字符设备文件

```bash

mknod /dev/mychardev c 主设备号 次设备号

```

这条命令创建了一个指向字符设备的节点。

### 创建块设备文件

```bash

mknod /dev/myblockdev b 主设备号 次设备号

```

这条命令创建了一个指向块设备的节点。

### 创建 FIFO（命名管道）

```bash

mknod /path/to/myfifo p

```

这条命令创建了一个 FIFO 特殊文件，允许进程之间进行通信。

### 示例

1. \*\*创建一个字符设备文件\*\*：

```bash

sudo mknod /dev/mydevice c 10 5

```

这里 `10` 是主设备号，`5` 是次设备号。

2. \*\*创建一个块设备文件\*\*：

```bash

sudo mknod /dev/mydisk b 12 8

```

这里 `12` 是主设备号，`8` 是次设备号。

3. \*\*创建一个 FIFO\*\*：

```bash

sudo mknod /tmp/myfifo p

```

这条命令创建了一个 FIFO 文件，可以在 `/tmp` 目录下使用。

### 注意事项

- 使用 `mknod` 命令需要管理员权限，因此通常需要在命令前加上 `sudo`。

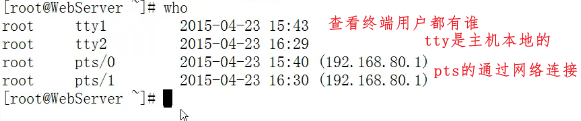
- 创建设备文件后，你可能需要修改文件的权限，以便用户可以正确地访问设备。

- `mknod` 命令创建的特殊文件只是文件系统中的一个节点，它们不会分配实际的物理设备或存储空间，它们只是提供了一个接口，允许操作系统和用户空间程序与相应的设备进行交互。

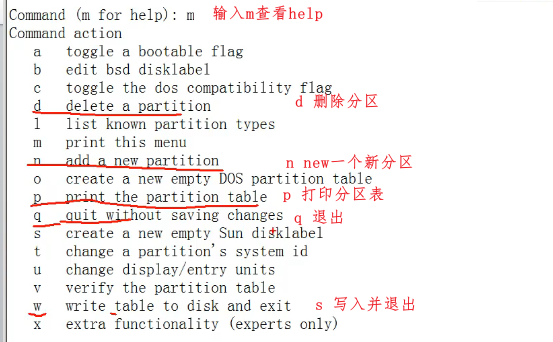
`mknod` 命令是系统管理员和开发人员在配置系统或开发驱动程序时常用的工具。



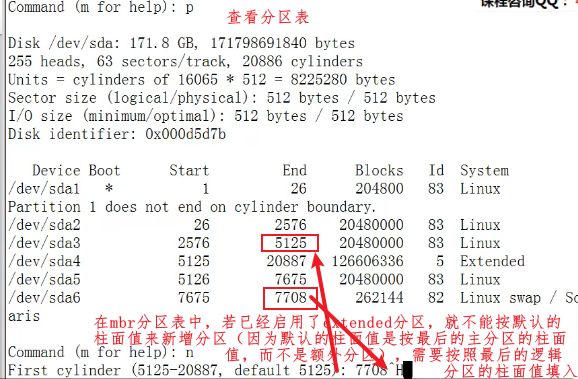
who am I 可以查看目前终端使用的身份的



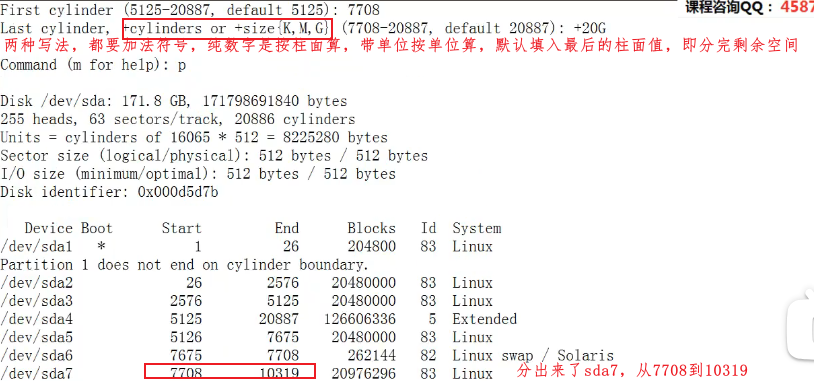
[07 创建磁盘分区OK\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1RW4y1H7Tt?spm_id_from=333.788.player.switch&vd_source=08c34b0f73c4926b3009b7bcf284e2d1&p=7)



使用fdisk对sda增加分区，查看分区表发现，sda已经分出了三个主分区，一个拓展分区，拓展分区从5125开始。所以第一个逻辑分区sd5从5126开始，后续增加逻辑分区应从末尾的逻辑分区的终止柱面开始，即7708.

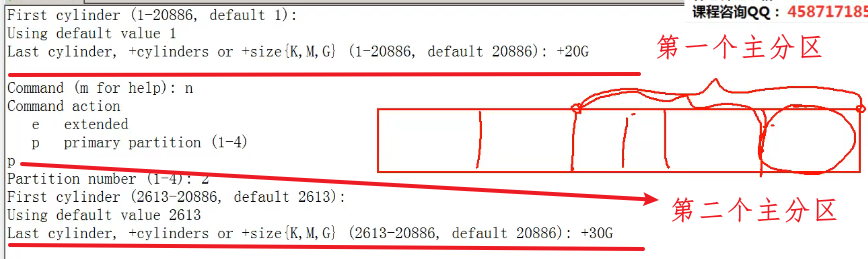


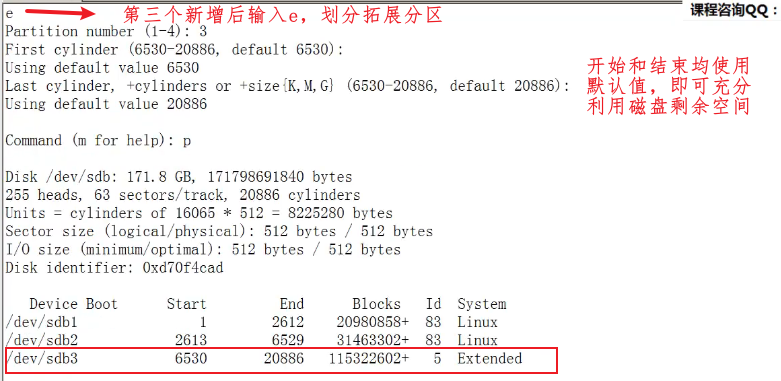
填入起始柱面值时，按下delete不会删掉数值，会出现^H，删除需要按下ctrl

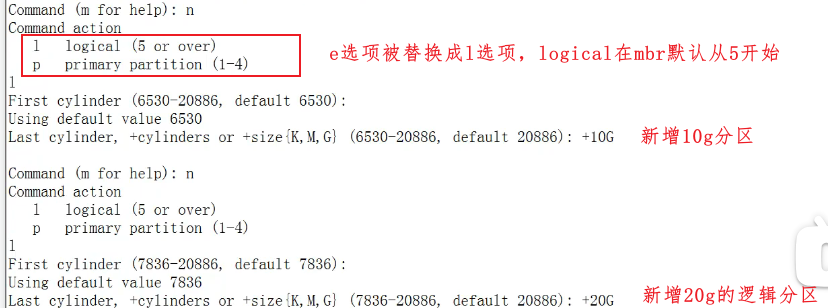
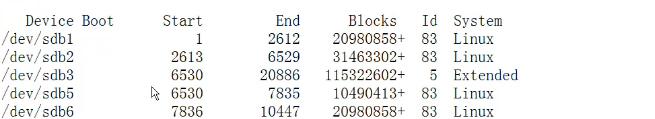


使用fdisk 对sdb硬盘进行分区

我们将sdb分为两个主分区+拓展分区，再在拓展分区里划分出逻辑分区







[08 介绍Linux虚拟文件系统VFSOK\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1RW4y1H7Tt?spm_id_from=333.788.player.switch&vd_source=08c34b0f73c4926b3009b7bcf284e2d1&p=8)

[09 超级块和块组描述符表(2)\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1RW4y1H7Tt?spm_id_from=333.788.player.switch&vd_source=08c34b0f73c4926b3009b7bcf284e2d1&p=9)

日志与非日志文件系统

查看linux支持的文件系统

[11使用mkfs和mke2fs创建文件系统\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1RW4y1H7Tt?spm_id_from=333.788.player.switch&vd_source=08c34b0f73c4926b3009b7bcf284e2d1&p=11)

mkfs

blkid

[数据恢复：详细介绍linux系统下Inode定位以及占用大小\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1eB4y1T76x/?spm_id_from=..search-card.all.click&vd_source=08c34b0f73c4926b3009b7bcf284e2d1)

要将 sdb1 分区重置，你需要执行以下步骤：

1. **卸载分区**： 在进行任何操作之前，确保 sdb1 分区没有被挂载。可以使用 umount 命令来卸载它：

bash

sudo umount /dev/sdb1

1. **删除分区内容**： 你可以选择删除分区中的所有文件，或者完全格式化分区。删除所有文件可以使用 rm 命令，但请注意这不会删除分区本身：

bash

sudo rm -rf /dev/sdb1/\*

或者，你可以使用 dd 命令将分区内容完全重写为零，这将删除所有数据：

bash

sudo dd if=/dev/zero of=/dev/sdb1 bs=1M

**警告**：dd 命令非常强大，使用时必须非常小心，确保你指定了正确的设备文件，否则可能会覆盖其他分区或硬盘。

1. **格式化分区**： 如果你想要重新格式化 sdb1 分区，可以使用 mkfs 命令。例如，如果你想要格式化为 ext4 文件系统：

bash

sudo mkfs.ext4 /dev/sdb1

或者，如果你想要格式化为其他类型的文件系统，比如 xfs：

bash

sudo mkfs.xfs /dev/sdb1

1. **重新挂载分区**（如果需要）： 如果你想要挂载 sdb1 分区，可以使用 mount 命令：

bash

sudo mount /dev/sdb1 /mount/point

其中 /mount/point 是你想要挂载到的目录。

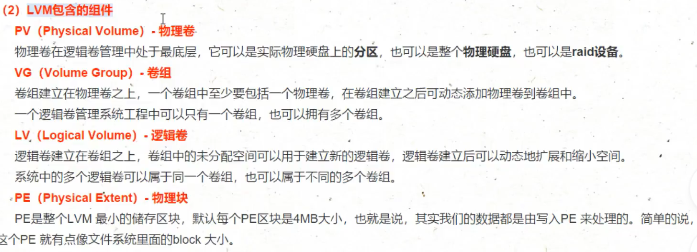
1. **检查分区**： 使用 fsck 命令检查分区的文件系统是否完好：

bash

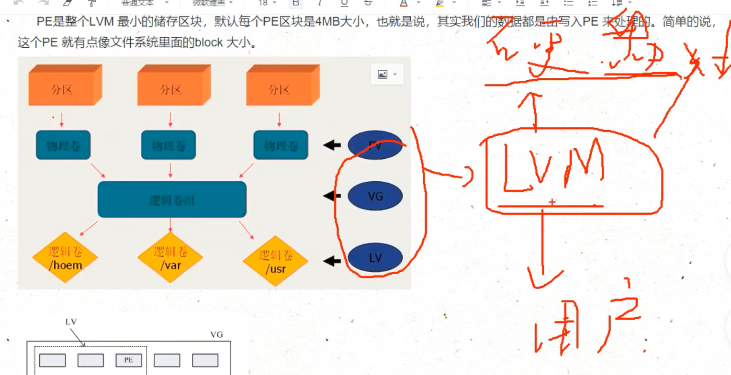
sudo fsck /dev/sdb1

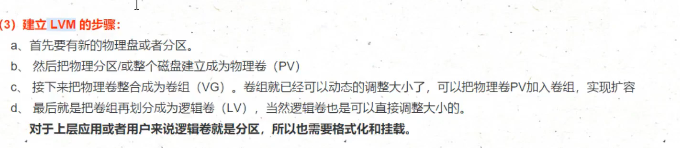
这个命令会检查并修复文件系统的错误。

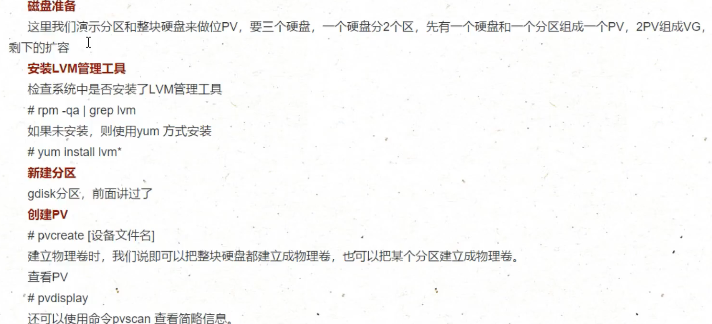
可以将分区设置为PV



物理卷是对文件系统的封装







rpm -qa | grep lvm

yum install -y lvm\*

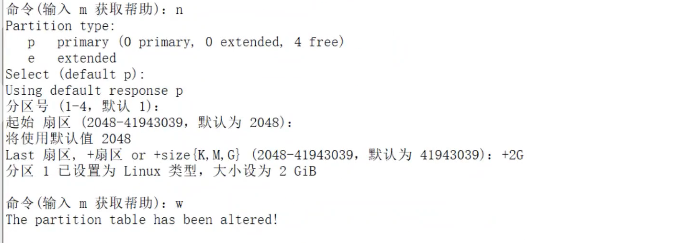
lsblk



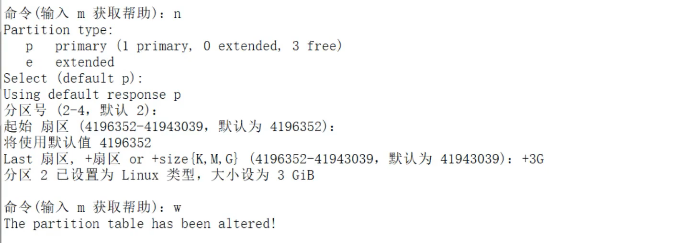


在sdb中划分两个磁盘分区2g、3g

fdisk /dev/sdb



fdisk /dev/sdb



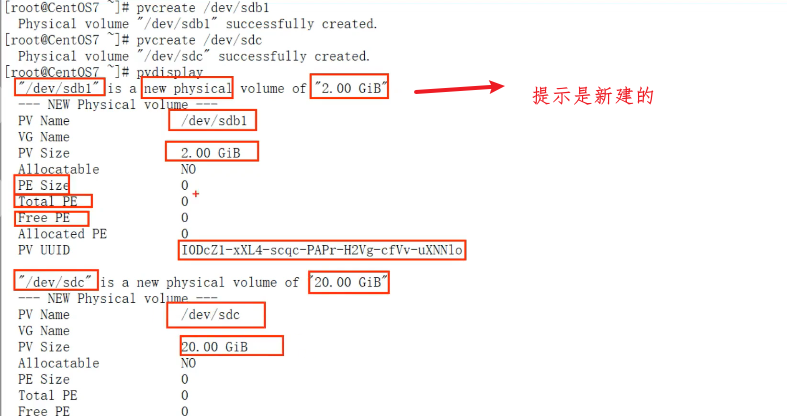
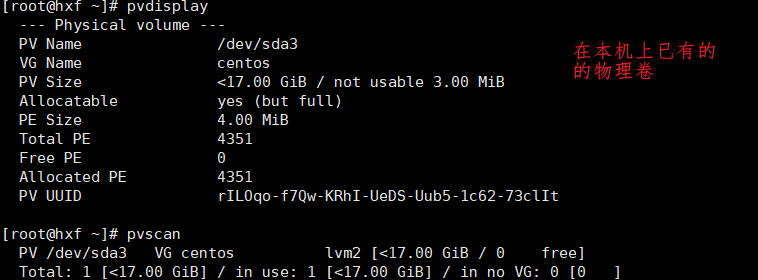
将sdb1和整个sdc作为pv，将这两个pv纳入同一个vg，建lv，挂载

pvcreate /dev/sdb1

pvcreate /dev/sdc

pvdisplay

pvscan

创建逻辑卷组, 缩写+vg+1

vgcreate [xuanxiang] vgname pv1 pv2

vgcreate

vgcreate hxfvg1

vgs

vgscan

vgdisplay

创建逻辑卷lv指定名称，,指定大小，指定对应的逻辑卷组，缩写+lv+1,1g

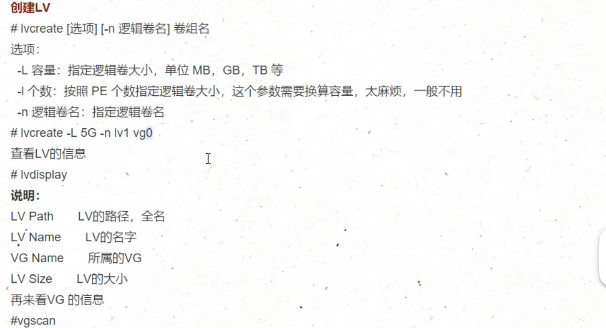
lvcreate -L 5G -n lv1 vg1

lvcreate -L 1G -n hxflv1 hxfvg1

lvdisplay

lvscan

lvs



格式化

mkfs.ext4 /dev/hxfvg1/hxflv1

mkfs.ext4 /dev/vg1/lv1

mkfs.xfs /dev/vg1/lv1

mkfs.xfs /dev/hxfvg1/hxflv1

挂载

创建挂载点mkdir /test/hxflv1

mount /dev/hxfvg1/hxflv1

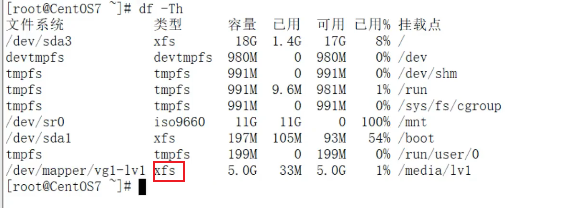
测试

cd /test/hxflv1

touch testlv.txt

echo ljljlfjalfjal > testlv.txt

df -Th



扩容xfs的逻辑卷lv1

扩容ext4的逻辑卷hxflv1

对lv1扩容1g，先检查vg1大小，vg1不够时需要通过增加pv先对vg1扩容，再在扩容后的vg1上扩容lv1

pvcreate /dev/

pvcreate /dev/

vgextend hxfvg1 /dev/

lvextend -L +1G -n hxflv1 hxfvg1

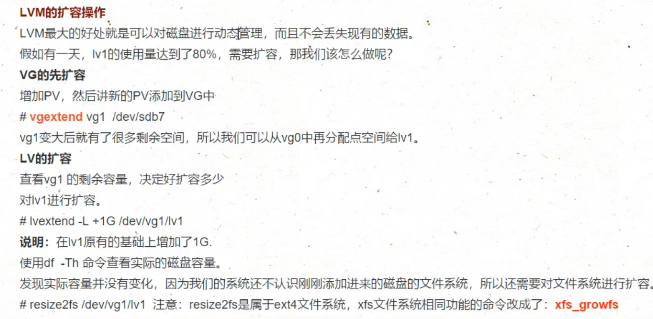
df -h 发现文件大小没有改变，文件系统当中还没改变，需要我们向文件系统扩容信息

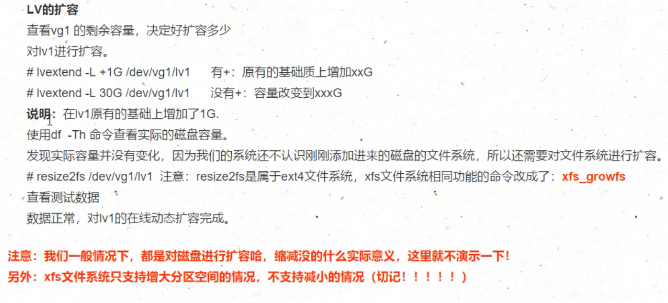
resize2fs /dev/hxfvg1/hxflv1

xfs\_growfs /dev/hxfvg1/hxflv1

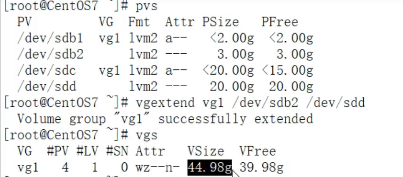
cd /test/hxflv1

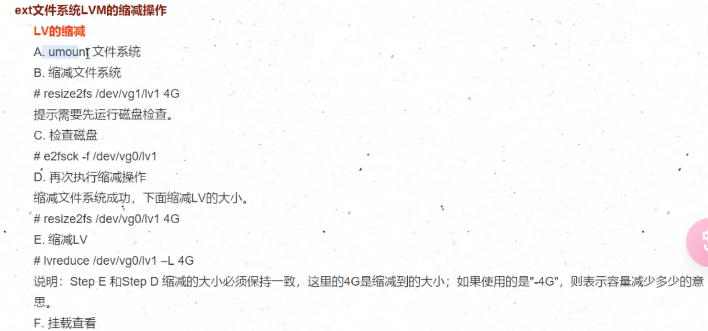
cat testlv.txt











[17 Linux逻辑卷管理LVMOK\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1RW4y1H7Tt?spm_id_from=333.788.videopod.episodes&vd_source=08c34b0f73c4926b3009b7bcf284e2d1&p=17)

原先用户使用硬盘需要进行如下操作：加装硬盘到本机、对硬盘分区、对分区高级格式化、挂载

lvm操作是在“对硬盘分区”与“对分区高级格式化”之间加入了一些步骤

[19 扩展卷组空间OK\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1RW4y1H7Tt?spm_id_from=333.788.player.switch&vd_source=08c34b0f73c4926b3009b7bcf284e2d1&p=19)

pvs

pvscan

pvdisplay的区别

pvs 命令的英文全称是 "Physical Volume Summary"

