linux 基础

磁盘管理

磁盘 IO设备

外部存储设备

机械式硬盘 造价低,廉价,

固态硬盘

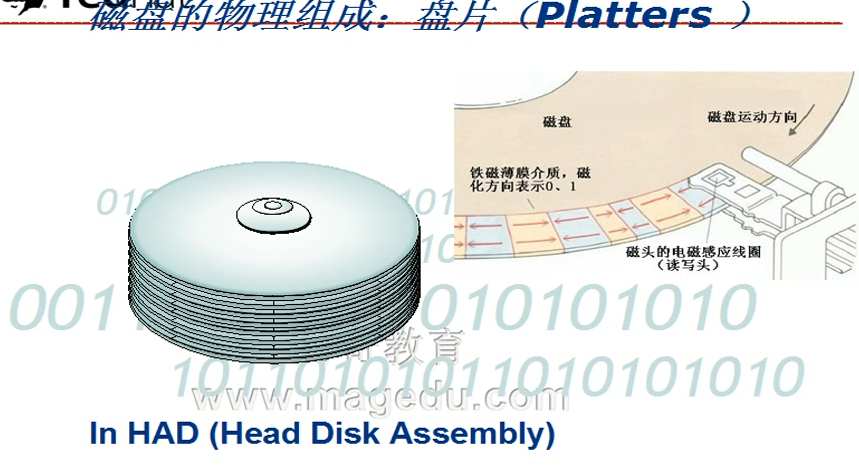
U盘

光盘

软盘

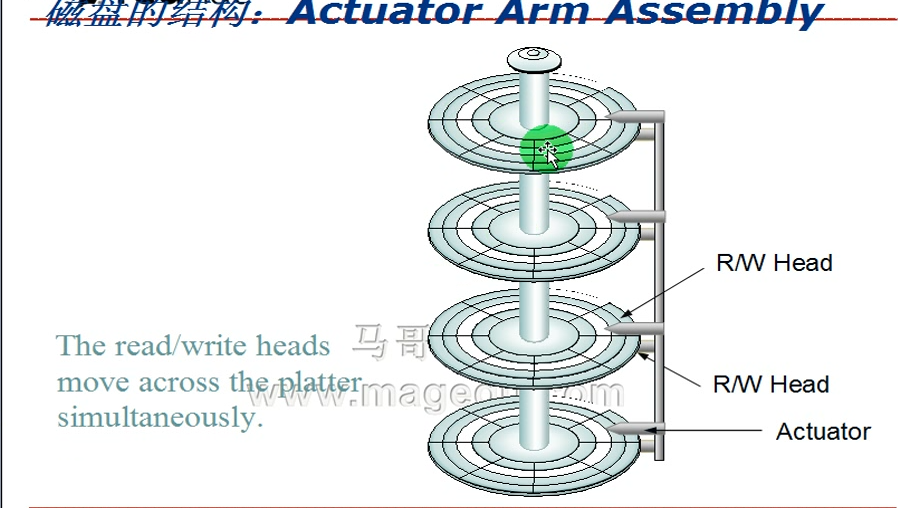
硬件

## 磁盘

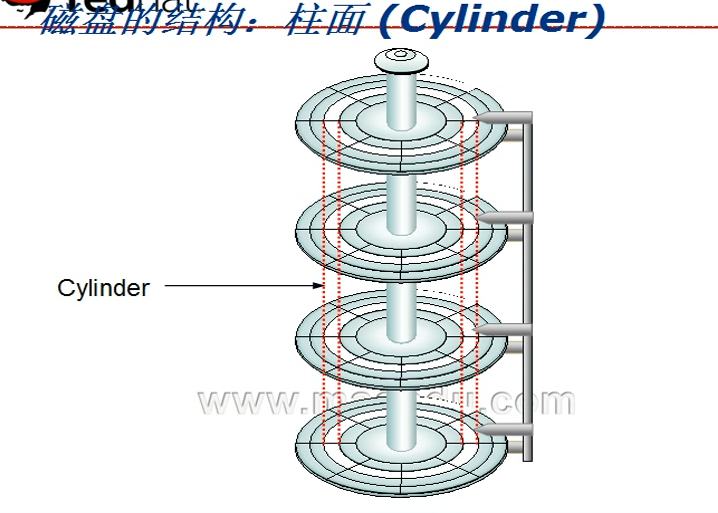












磁道划分成扇区,不同盘面上的相同编号磁道组成柱面

为什么划分分区: partition

可以建立不同的文件系统

逻辑边界

硬盘刚出厂的时候要低级格式化

## MBR

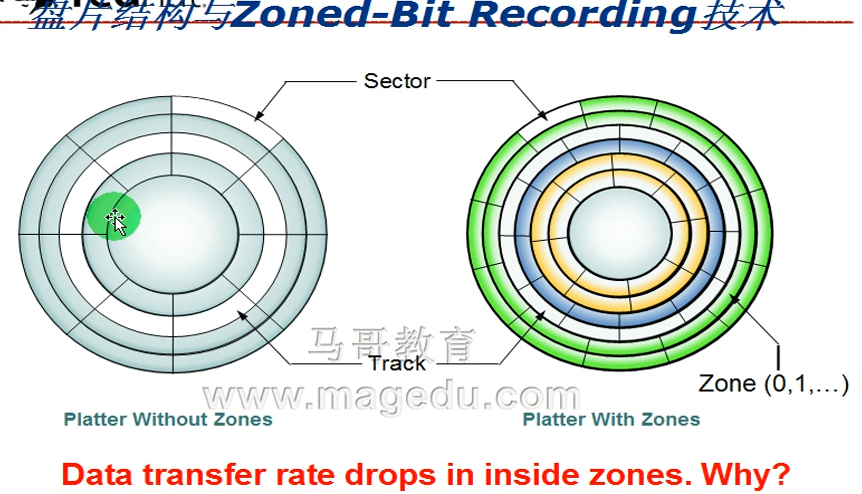
MBR Main Boot Record 主引导记录,

主引导记录放在0磁道的0扇区,512 byte 分3段 不是操作系统使用

前446字节 BootLoader 一段代码 引导加载器 去引导某个分区的操作系统,让系统启动起来

中间64字节每 16个字节标识一个分区可以标识4个分区

剩下的2个字节 magic number 标记MBR是否有效





## BIOS

BIOS Basic Input Output System 基本输入输出系统

BIOS**其实它是一组固化到**[**计算机**](http://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA)**内**[**主板**](http://baike.baidu.com/item/%E4%B8%BB%E6%9D%BF)**上一个**[**ROM**](http://baike.baidu.com/item/ROM)[**芯片**](http://baike.baidu.com/item/%E8%8A%AF%E7%89%87)**上的**[**程序**](http://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F)**，它保存着计算机最重要的基本输入输出的程序、开机后自检程序和系统自启动程序，它可从CMOS中读写**[**系统设置**](http://baike.baidu.com/item/%E7%B3%BB%E7%BB%9F%E8%AE%BE%E7%BD%AE)**的具体信息。** 其主要功能是为计算机提供最底层的、最直接的硬件设置和控制。此外，BIOS还向作业系统提供一些系统参数。系统硬件的变化是由BIOS隐藏，程序使用BIOS功能而不是直接控制硬件。现代作业系统会忽略BIOS提供的抽象层并直接控制硬件组件。

BIOS 程序可以设定boot的启动次序.装机中常修改启动次序.

## 电脑启动过程

MBR是启动系统的入口. 很重要

1.电脑刚开机的时候, BIOS完成加电自检 ,将一段 (固定到芯片中)的程序加入到内存,自身健康检查没问题了, 接下来根据BIOS根据设定的启动次序,挨个找设备的MBR(光盘设备,硬盘设备,软盘设备,网络设备,移动设备…)

2.如果找到硬盘的BMR, BIOS就会将BootLoader这段程序载入内存,BIOS就退出了,将控制权交给BootLoader.

3. BootLoader程序接下来找那**4个分区表**的分区,加载分区上的操作系统的内核到内存, 这4个分区表是主分区表.

4. 内核解压缩启动起来过后, BootLoader将控制权交给内核.并退出. 内核找到文件系统,各种驱动,启动上层应用.

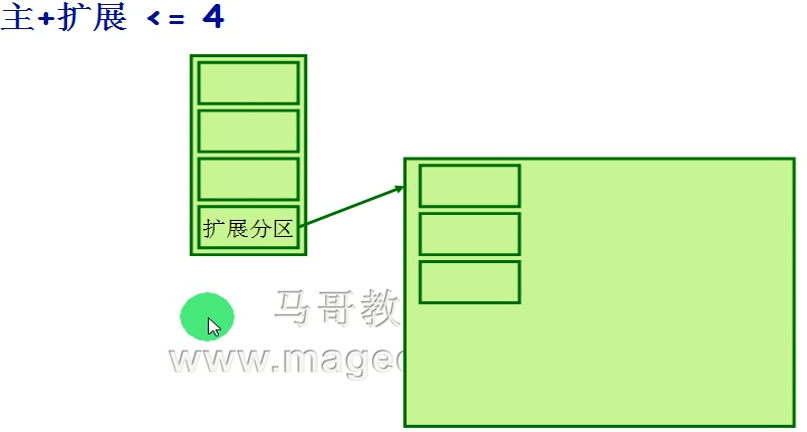
## 主分区

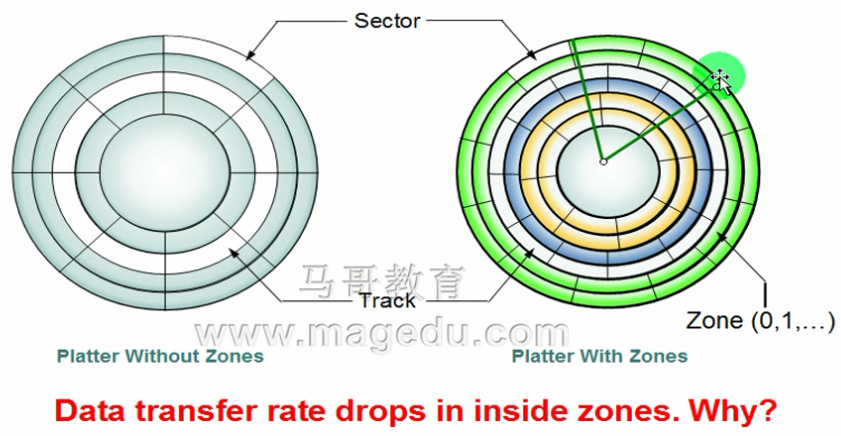
64字节标识 每16个字节标识一个分区 , 可以标识4个主分区

## 扩展分区

扩展只能一个,

主分区+扩展分区不能超过4个





RPM 5400 7200 15000… 每分钟多少转

角速度一定,越靠近外层的柱面,读写速度效率越高,一般C盘在最外层柱面,D盘次之,…

## 文件系统

是一个软件,为了高效的读写文件,查找文件.而存在

文件系统就是一个管理软件.

将磁盘分区从逻辑上划分两个区域: **元数据区 和 数据区**

**元数据区**存放 逻辑块的地址索引(索引节点, index node inode) 和 块位图 (bitmap)

**数据区**划分为一个一个的逻辑块(数据块) 逻辑块的地址以索引的方式保存在元数据区

写数据的时候,怎么知道那些块可以用那些块不可以用?

数据块存了数据为1 没有标识为0

所以在元数据区有一个区域存放**块位图**,用1bit存放这些逻辑块的使用情况

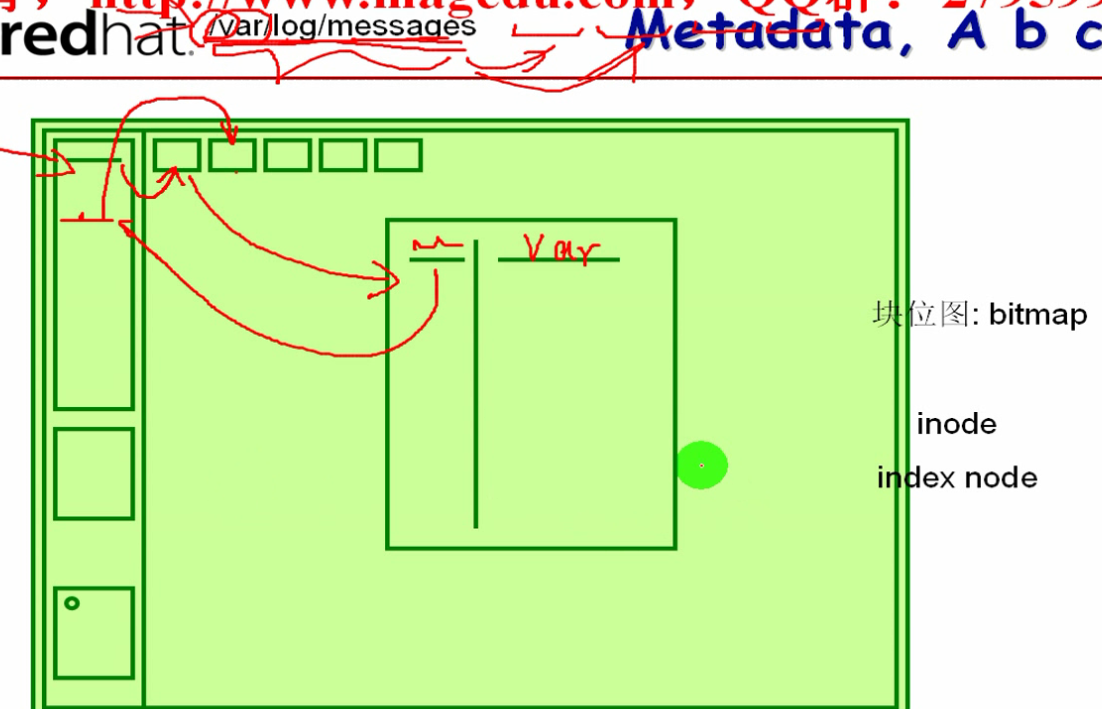
写的时候就遍历块位图,就不用扫描整个分区.

inode中存放 inode号,对应文件的权限,属组,属主,扩展属性,时间戳,大小,… 跟文件内容没有关系. 没有文件名

inode也有bitmap,对应到inode中的条目,哪些用,哪些没用,都标识清楚了

那么文件名存在哪儿?

Linux系统 文件名是存放在目录中的, **目录也是一个文件**,Linux启动时候,内核会引用到根.(自引用) 就像一棵树.从根开始,就可以找到需要找到的文件



**目录有两个区域: inode和 文件名列表( dentry)**

/var/log/messages

1.从根目录(/)开始,/是个文件,在/中找var名称对应的inode,根据inode到分区的元数据区查找var数据块,

2.在var目录数据块中查找log名称,然后找到log对应的inode,回到分区的元数据区查找log数据块,

3.在log数据块中查找messages名称对应的inode,回到分区的元数据查找messages数据块,就找到了.就读取messages的内容

如果路径太长,而又经常访问,可以把路径放入缓存.就不会耗费时间一级一级的去找

目录中创建文件过程: tmp/test.txt

1.在元数据去找一个空闲的inode (扫面bitmap为0的空闲)

2.找到tmp数据库块并在tmp数据库上新建test.txt的条目(文件名)并将空闲的inode号为test.txt的inode.在bitmap上标记为1 已使用,

3.存数据,先在分区元数据区找空闲的inode,( 扫面bitmap为0的空闲)比如要存10k,系统一下会申请超过10k的数据块,为了避免文件碎片. 如果每个块2k 可能会一下申请8个块,用不完的归为其他inode就可以了.

碎片意味着不连续,磁头要重新寻道,耗费时间.

删除文件 /tmp/test.txt

根据inode找到tmp数据块,删除test.txt条目,并根据test.txt对应的inode号将bitmap与其对应的bit重置为0即可,标识空闲.. 下次使用,覆盖数据块即可,所以文件删除只要不写入新数据,是可以找回来的.

复制cp很慢?

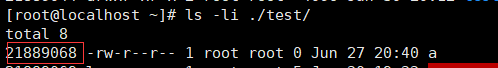
剪切mv很快?

复制就是创建的过程

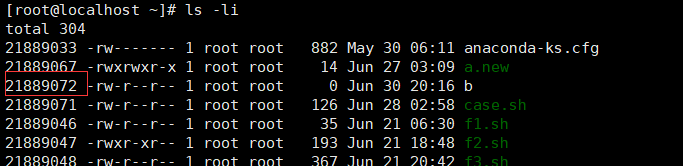
剪切过程只能在同一个文件系统,只是把路径条目改变了一下.inode与inode对应的数据块没有动.

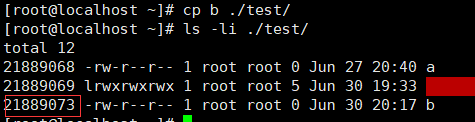






mv的时候, inode是不变的

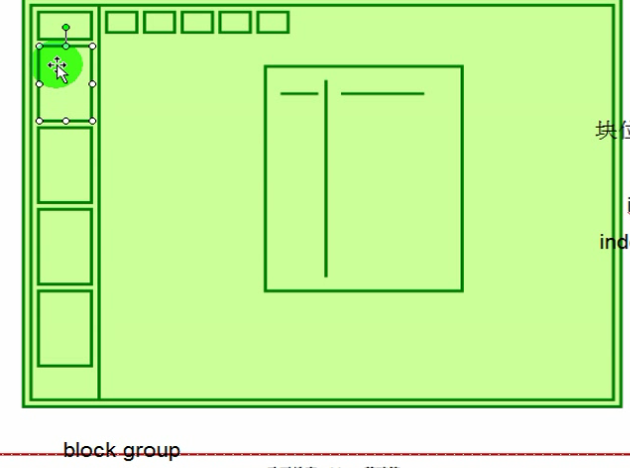




cp的时候 inode号变了

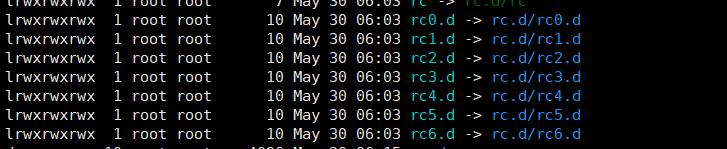
实际上由于磁盘是这样划分的

元数据按照逻辑块组,组下面有自己的bimtmap inodebitmap 等



## 链接

很多的链接文件都是777的

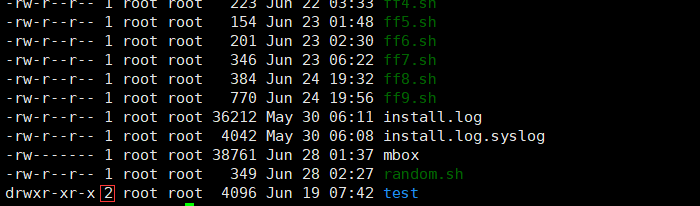


链接文件分为两类

硬链接 和 软连接(符号链接)

硬链接 两个不同的路径指向同一个inode称为硬链接

软连接 inode存的是一个指向另一个inode的路径. 文件大小是路径的字符个数 10个字符,他的权限是777 因为最终访问权限要取决另一个文件的权限.



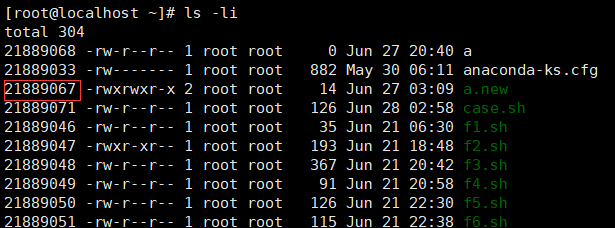
文件被硬链接的次数, 如果大于1 ,删除时,不会真正删除,只有小于等于1的时候会真正删除

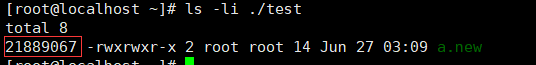
**创建链接**

**ln** [ -v -s ]

-s 创建符号链接

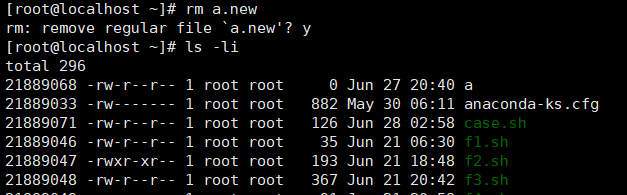
-v 显示创建过程



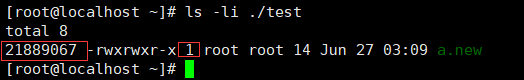


指向同一个inode 21889067

现在删除a.new



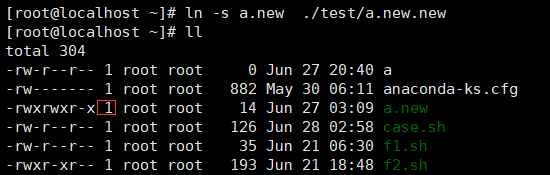
test目录下依然存在 不过文件硬链接次数变为1

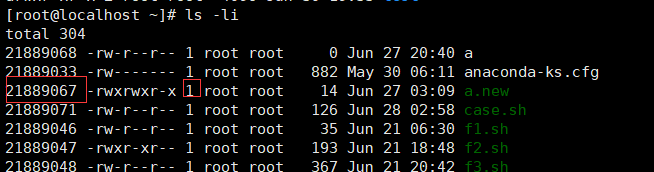


创建软连接

**ln -s**

ln -s a.new ./test/a.new.new

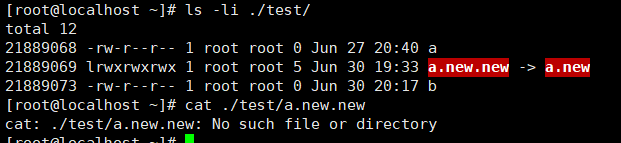




inode不同



如果我将a.new删除,那么 就会出现No such file or directory



硬链接:

1. 只能对文件创建,不能应用与目录

2. 不能跨文件系统

3. 创建文件硬链接会增加文件被链接的次数

符号链接:

1. 可应用与目录

2. 可以跨文件系统

3. 不会增加被链接的次数

4. 其大小为路径包含字符的个数

## du

du disk usage of each FILE

estimate file space usage 估计文件空间使用情况

**-h** 单位换算

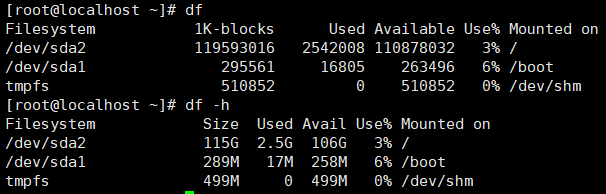
**-s** /lib 评估目录包含文件总共大小 不带-s 评估目录下的每个文件大小



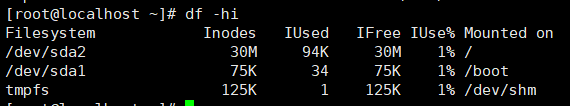
## df

df - report file system disk space usage 文件系统磁盘空间使用情况

**-h** 单位换算



**-i 显示inodes**



**-p**

不用换行

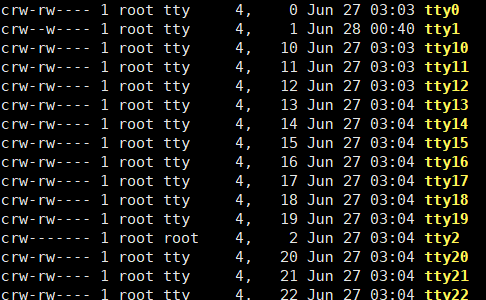
设备文件:

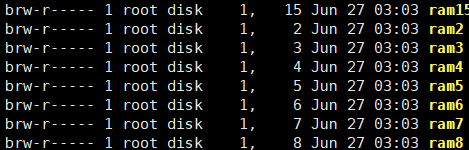
块设备 按块为单位,随机访问的设备 硬盘

字符设备 按字符为单位 线性设备 键盘 硬盘

4 , 0

**分别表示主设备号,和次设备号**





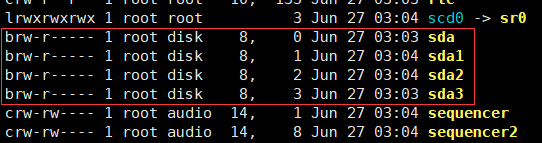
/dev

主设备号 (major number)

标识设备类型

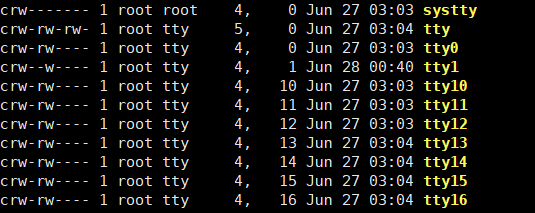
次设备号 (minor number)

标识同一种类型的的不同设备



sda1 sda2 sda3 就是3个主分区 设备号8 次设备号从1-3

字符设备 终端



之所以是特殊文件,他们并不占用磁盘空间,

**主设备号和次设备号存储在inode当中**,就像符合链接一样,inode中存储的是路径.

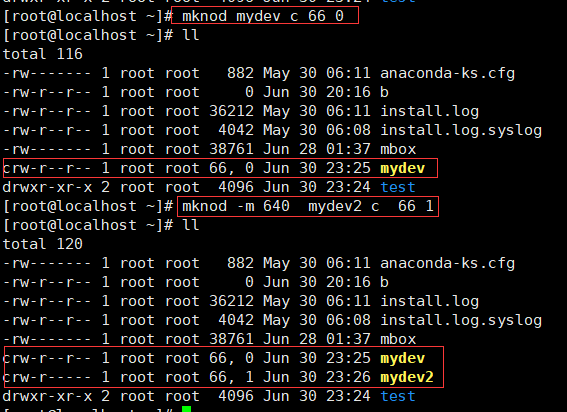
## mknod

创建特殊文件

mknod - make block or character special files 创建块或者字符特殊文件

mknod [OPTION]... NAME TYPE [MAJOR MINOR]

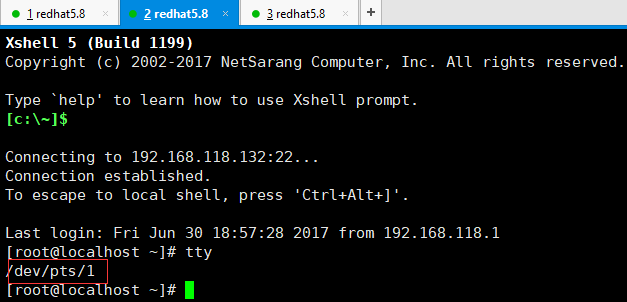
**-m** 指定权限

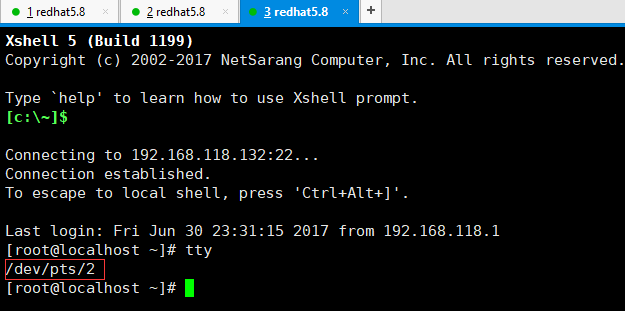


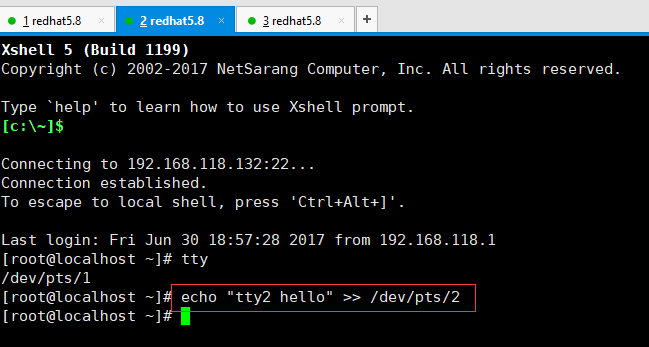
这个文件能干什么?

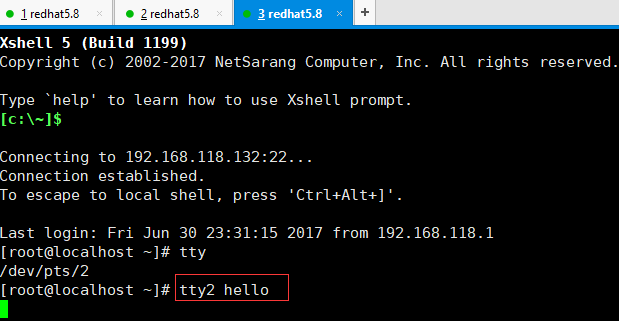
这个文前提是主设备号(66)与真正的设备进行关联











如果是硬盘sda,就惨了,发送的内容会覆盖sda的MBR, 所以不要轻易想设备上发送内容.

## 硬盘设备的设备文件名

硬盘类型

IDE ATA : hd开头

SATA : sd开头

SCSI : sd开头

USB : sd 开头

命名规则: a b c ……

IDE :

第一个IDE口 : 主盘 ,从盘

/dev/hda /dev/hdb

第二个IDE口 : 主盘 ,从盘

/dev/hdc /dev/hdd

sda sdb sdc ,,,, 具体怎么编比较麻烦 UDEV来标识剩下3种

扩展分区+主分区 <= 4

主分区最多只有4个,扩展分区可以是这4个当中任意一个

hda:

hda1 : 第一个主分区

hda2

hda3

hda4

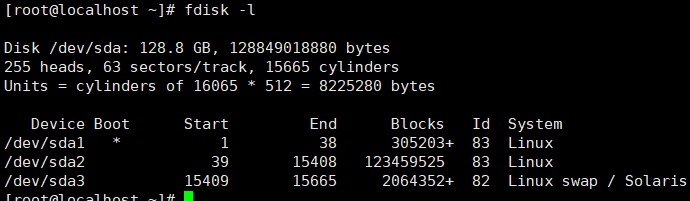
hda5:第一个逻辑分区

hda6:第二个逻辑分区

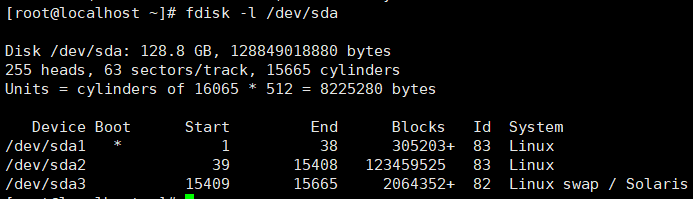
## fdisk

列出磁盘信息

fdisk -l



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



还可以管理磁盘分区

Linux支持的文件系统

文件系统是一个管理文件的软件,Linux内核提供此功能

## 格式化

低级格式化 磁盘出厂前,创建磁道

高级格式化 用户创建文件系统 比如我将磁盘格式化为ext3 ,windows系统将U盘格式化成FAT32 NTFS 等…

光盘 ISO9660

网络 CIFS

linux 支持的文件系统有很多种 ext3 ext4 ext5 xfs reiserfs jfs(IBM) jfs2 ……

linux 网络上支持的文件系统 nfs ocfs2(集群文件系统) gfs2(全局文件系统)

linux 还支持windows的FAT32 叫做vfat

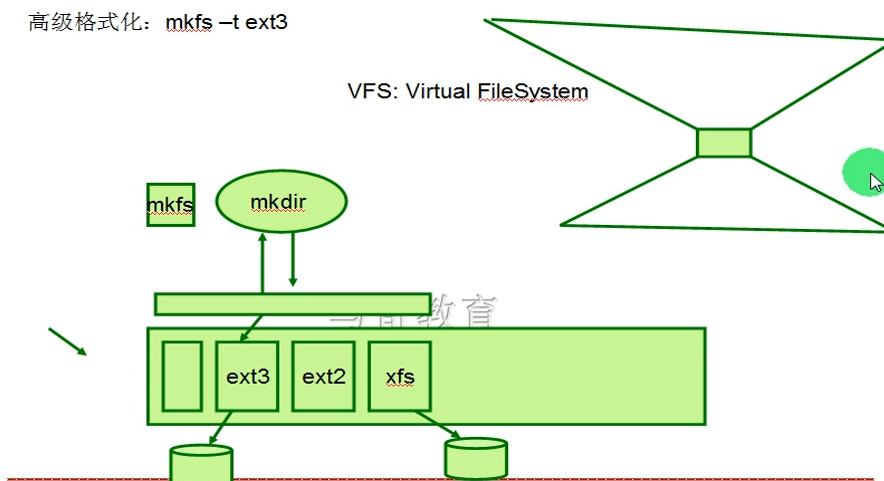
因为Linux底层还有VFS 虚拟文件系统功能



## mkfs

创建文件系统的命令

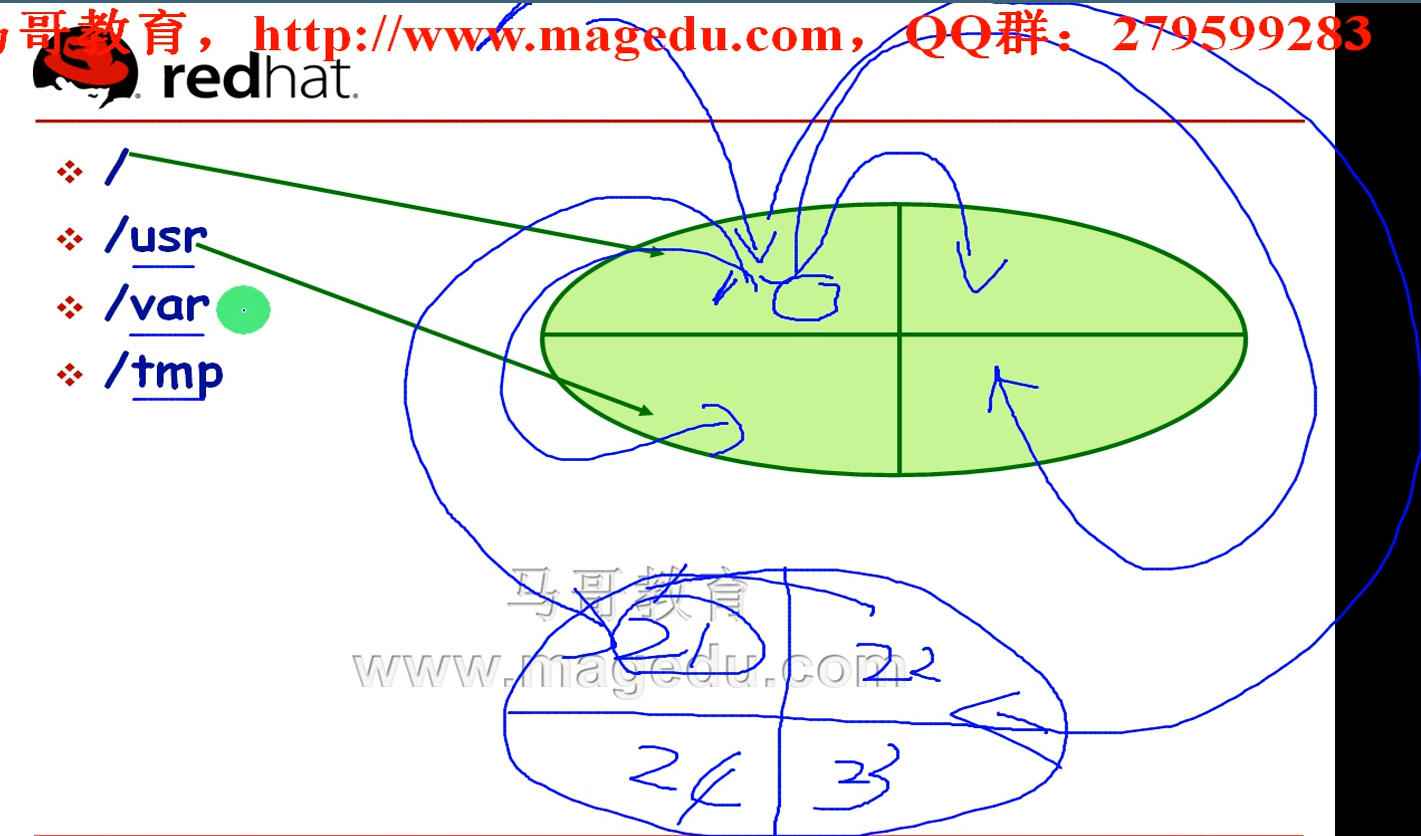
mkfs -t ext3



VFS 封装到内核中, 内核具有此功能.

一次每个分区都可以使用不同的文件系统

物理视角 和 逻辑视角



根是自引用的(只有根才能自引用),内核就知道根,根是高于硬盘的.

一切都从根开始,这是逻辑概念, 只有一个

只要对根分区的文件做了**访问入口**,比如/usr 那么usr 条目一定在根分区,但是数据在4分区上,如果没有访问入口,那么数据就会保存在根分区上.

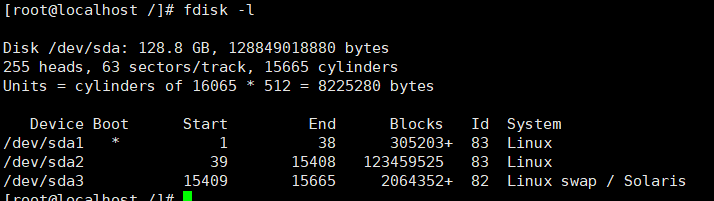
根文件系统,根分区有自己的规则,不是每一个目录的内容都可以独立出去 /bin /sbin /etc 是根分区比不可少的

如果又来了一块磁盘,这个磁盘上也有4个分区,这块磁盘上没有根了,根只能有一个,根是一个逻辑概念.只能在跟上建立一个文件与这4个分区进行挂载和关联,所有根很重要.

磁盘出厂后高级格式化并进行分区然后**创建好文件系统**,**才能被挂载或者关联到根或者根下的某个目录才能访问.**

## 磁盘管理分区

fdisk /dev/sda



/dev/sda 磁盘 128.8GB 128849018880 个字节 255个头 63个扇区 15665个柱面

磁盘 柱面范围

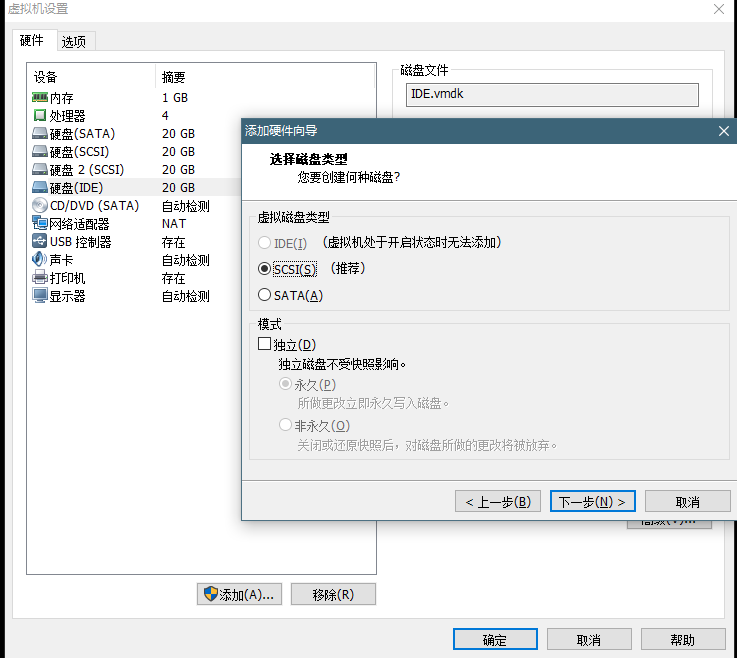
sda1 1-38

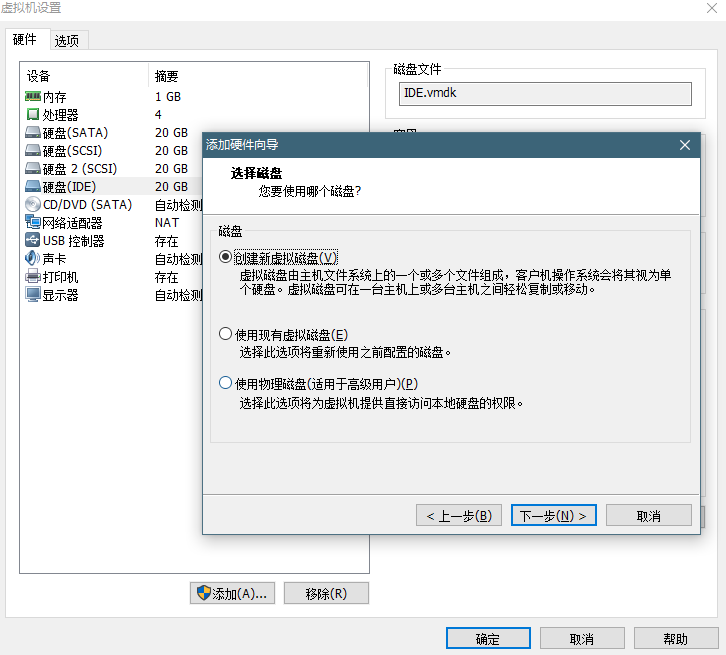
sda2 39-15408

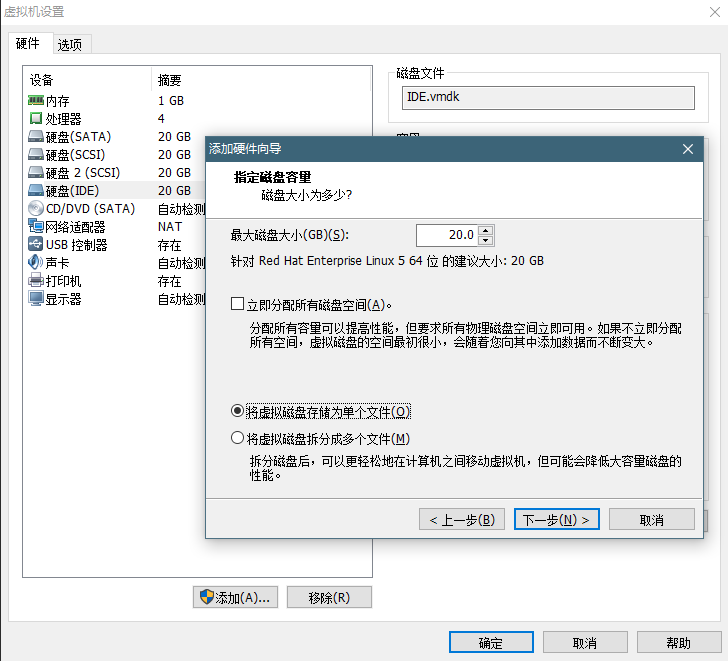
sda3 5409-15665

这3个主分区已经有数据了 ,而且柱面用完了

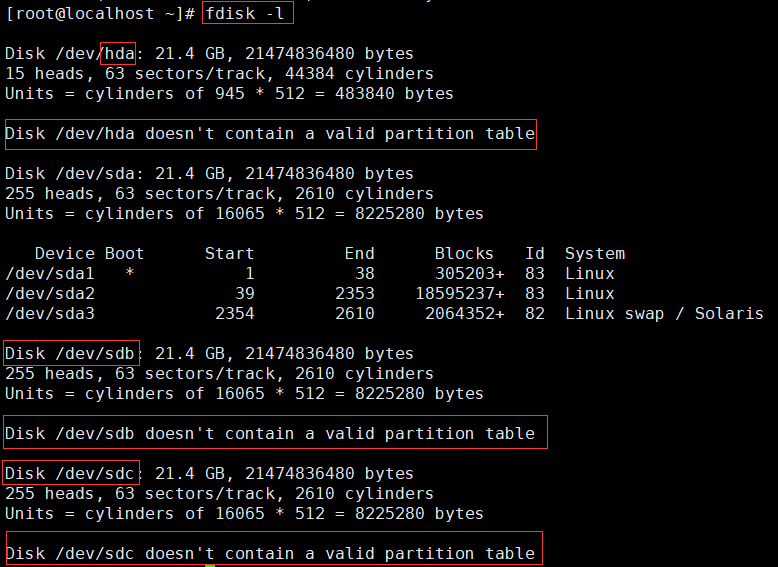
在虚拟机里面添加几个磁盘进来



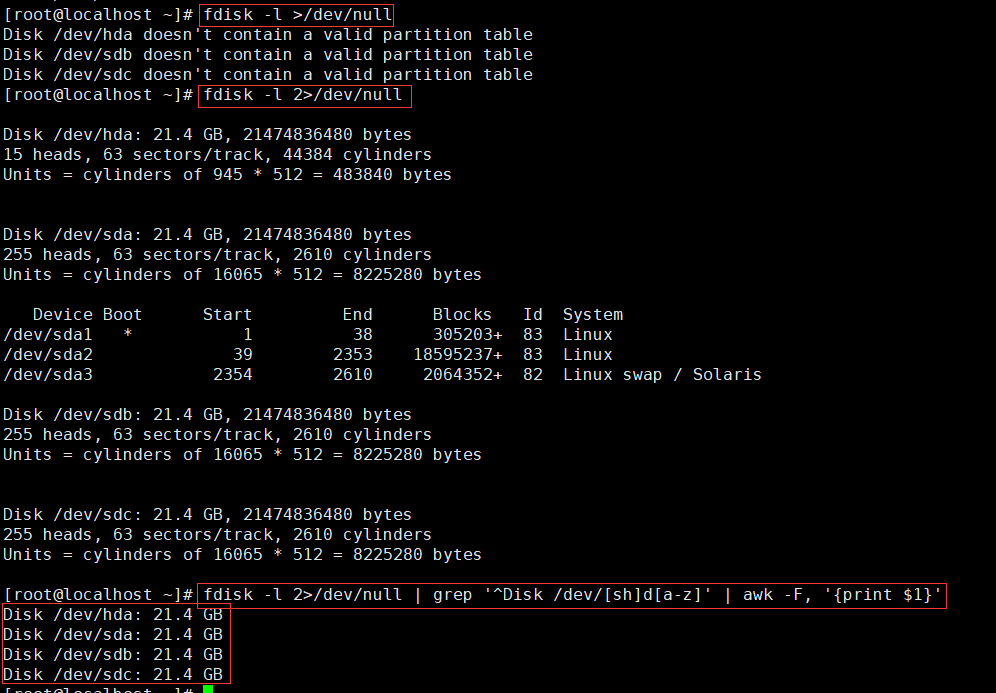




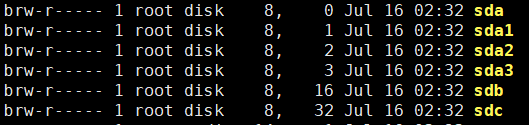
重启系统



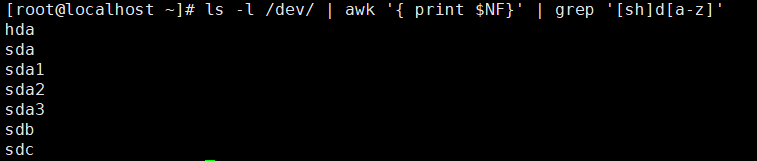
fdisk -l 2>/dev/null | grep '^Disk /dev/[sh]d[a-z]' | awk -F, '{print $1}'



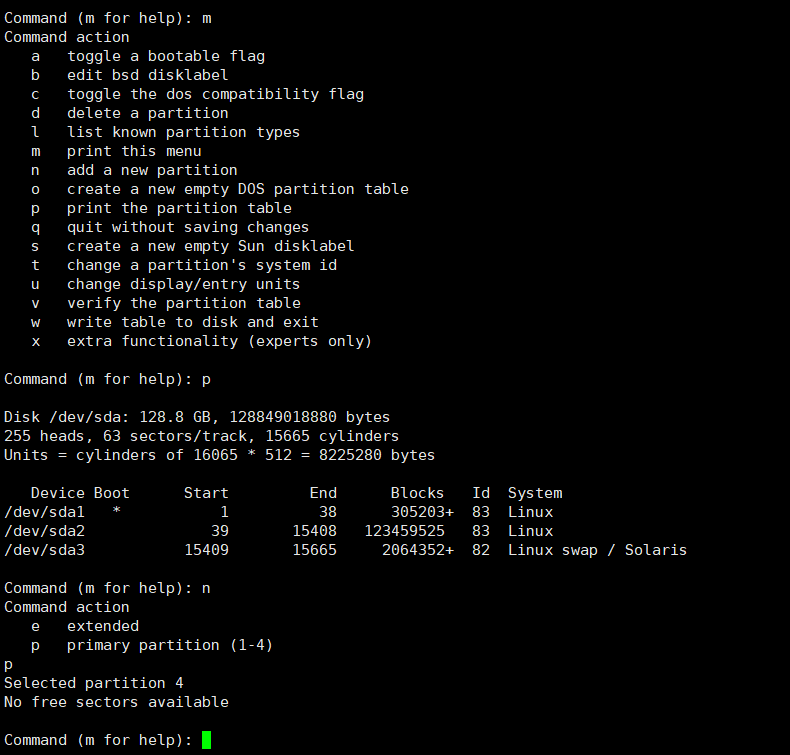
ls -l /dev/



ls -l /dev/ | awk '{ print $NF}' | grep '[sh]d[a-z]'

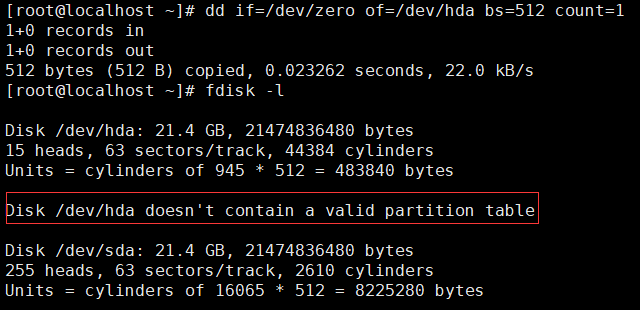


fdisk /dev/hda



dd if=/dev/zero of=/dev/hda bs=512 count=1

快速抹去一块硬盘上的所有分区



no free secotors avaliable

p 显示当前分区,包括没有保存的改动

n 创建新的分区

e 扩展分区

p 主分区

d 删除一个分区

w 保存并退出

q 不保存退出

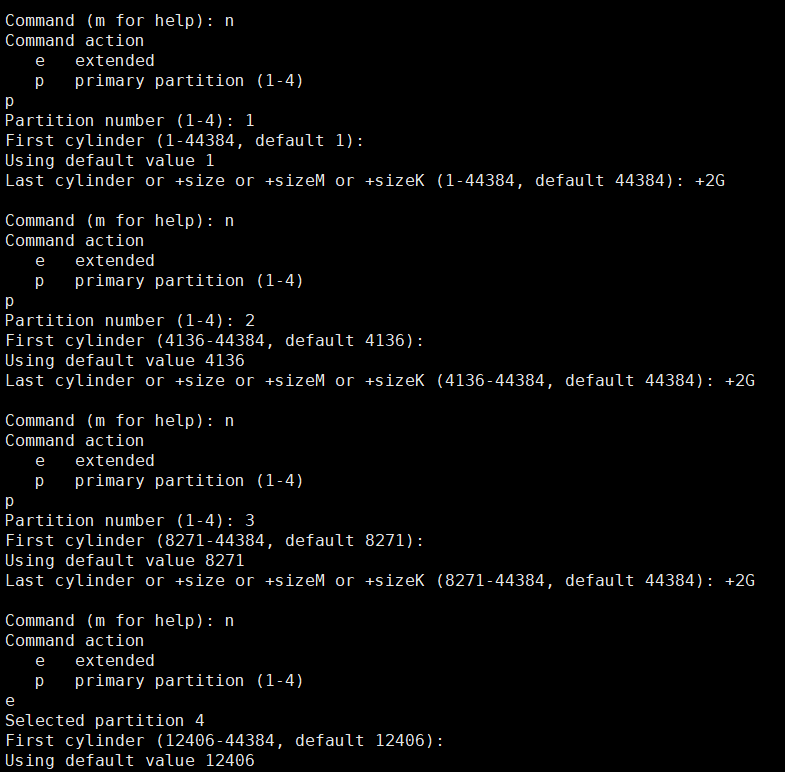
t 修改分区类型 默认ext#

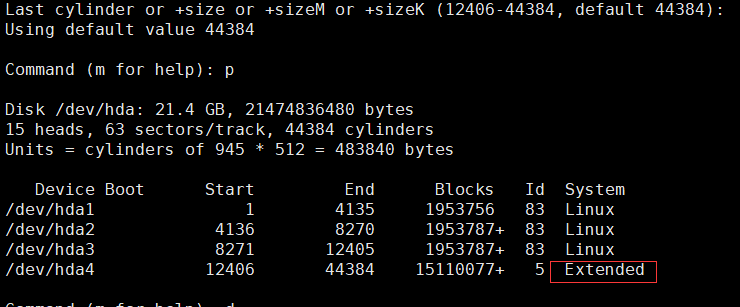
L: 显示支持的所有类型

l 显示支持的所有类型

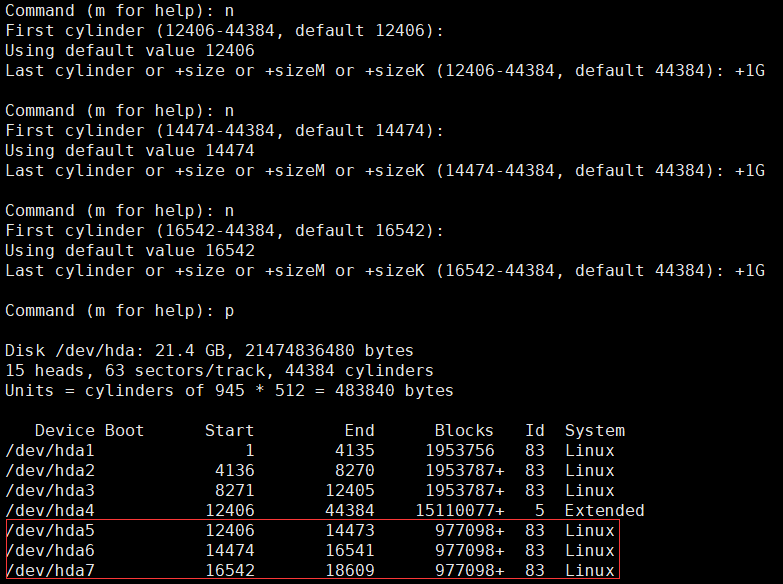
如果输入错误的命令使用Ctrl+u 撤销操作

把had分为3个主分区和1个扩展分区,主分区大小2G,剩下的给扩展分区

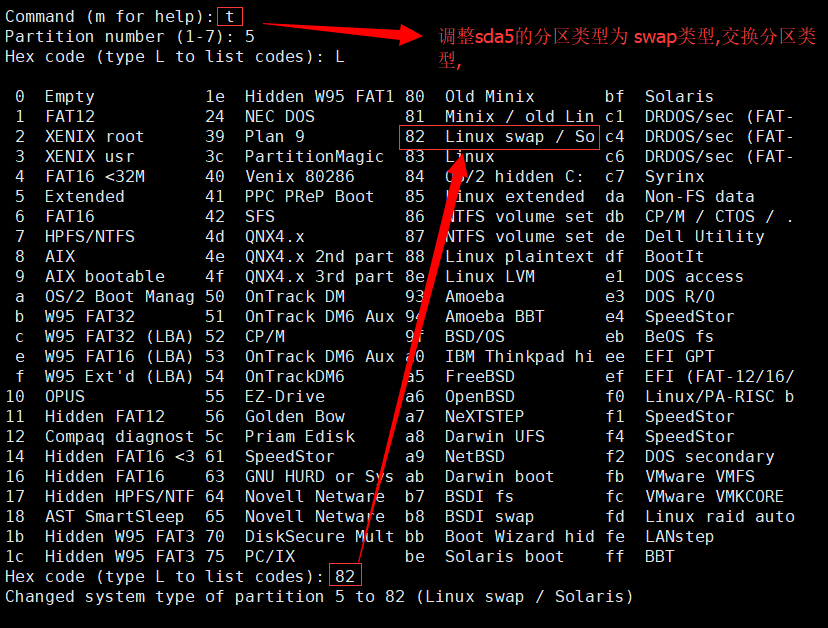


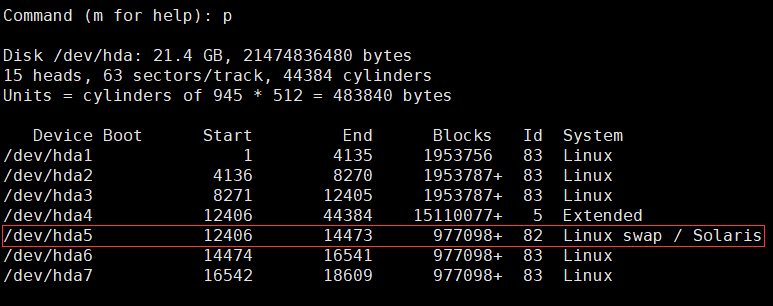


扩展分区上建立新分区

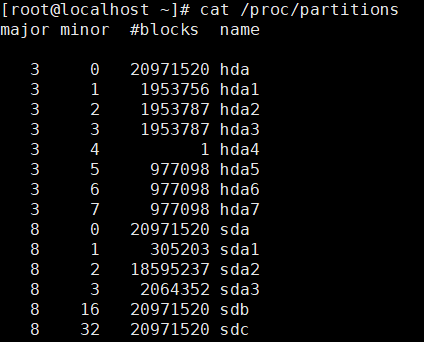


调整分区类型 Linux swap linux 的交换分区



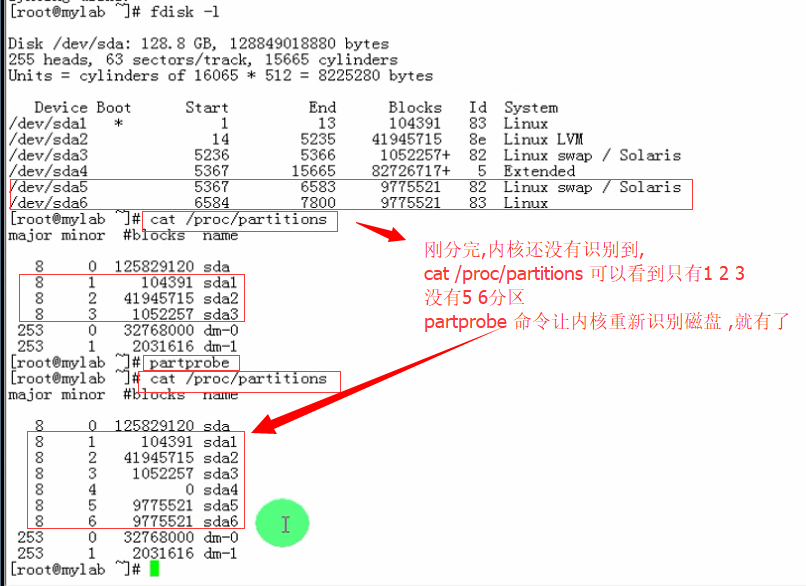


cat /proc/partitions



## partprobe

如果内核没有识别到,要通知内核识别,使用命令 partprobe



partprobe命令让内核重读分区表.就可以识别到新的分区,然后才可以格式化,挂载

partprobe redhat6 有其他的一个命令