

**1-19 RHEL6启动原理和故障排除**

**1.1-Linux启动过程详解**

**1.2-常见的启动故障排除**

**1.2-1 实战-防止黑客破解root密码**

**1.2-2 实战-服务器挂机,转移数据**

**1.2-3 实战-修复磁盘I节点耗尽**

**1.2-4 实战-磁盘坏道检测**

**一 .Linux启动过程详解**

**其实就是转交系统控制权的过程**

**Linux操作系统的引导过程中一般包括一下几个步骤**

**开机自检🡪MBR引导🡪GRUB菜单🡪加载Linux内核🡪init进程初始化🡪登录系统**

**BIOS ->** MB**R -> GRUB -> kernel -> init process -> login**

**1,服务器主机开始的时候,根据主板上的BIOS中对CPU 内存 显卡 键盘等设备进行初步检测,检测成功后根据CMOS预设的启动顺序移交系统控制权.**

**2,根据第一扇区的MBR主引导记录的设置,将系统的控制权交给包含操作系统的引导分区,或根据MBR记录中的引导信息调用启动菜单grub (MBR)**

**3,系统将控制权交给GRUB以后,将会显示启动菜单提供给用户选择,并根据所选项加载Linux内核文件,然后将系统的控制权交给内核.(GRUB)**

**4,内核获得系统的控制权以后,负责资源分配与调度,将完全掌握整个Linux操作系统的运行过程. (kernel)**

**5,由内核先将系统中的 /sbin/init 程序加载到内存中运行,完成初始化过程(init)**

**6.等待用户登录(login)**

**第一步--加载BIOS**

**因为BIOS中包含了CPU的相关 信息、设备启动顺序信息、硬盘信息、内存信息、时钟信息等等。**

**第二步--读取MBR**

**硬 盘上第0磁道第一个扇区被称为MBR，也就是Master Boot Record，即主引导记录，它的大小是512字节，可里面却存放了**

**预启动信息、分区表信息。**

**可分为两部分：**

**第一部分为引导（PRE-BOOT）区，占了 446个字节；**

**第二部分为分区表（PARTITION PABLE），共有66个字节，记录硬盘的分区信息。**

**引导区的作用之一是找到标记为活动（ACTIVE）的分区，并将活动分区的引导区读入内存。**

**系统找到BIOS所指定的硬盘的MBR后，就会将其复制到地址所在的物理内存中。**

**其实被复制到物理内存的内容就是Boot Loader，而具体到你的电脑，那就是grub了。**

**第三步－－Boot Loader**

**小程序 初始化硬件,建立内存空间**

**Boot Loader有若干种，其中Grub、Lilo和spfdisk是常见的Loader，这里以Grub为例来讲解吧。**

**系统读取内存中的grub配置信息（一般为menu.lst或grub.lst），并依照此配置信息来启动不同的操作系统。**

**第四步－－加载内核**

**系统将解压后的内核放置在内存之中，并调用start\_kernel()函数来启动一系列的初始化函数并初始化各种设备，完成Linux核心环境的建立。至此，Linux内核已经建立起来了，基于Linux的程序应该可以正常运行了。**

**第五步init依据inittab文件来设定运行等级**

**内核被加载后，第一个运行的程序便是/sbin/init，该文件会读取/etc/inittab文件，并依据此文件来进行初始化工作。**

**其实/etc/inittab文件最主要的作用就是设定Linux的运行等级，其设定形式是“：id:5:initdefault:”，这就表明Linux需要运行在等级5上。Linux的运行等级设定如下：**

**0：关机**

**1：单用户模式**

**2：无网络支持的多用户模式**

**3：有网络支持的多用户模式**

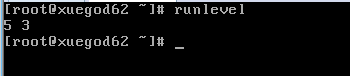
**4：保留，未使用**

**5：有网络支持有X-Window支持的多用户模式**

**6：重新引导系统，即重启**

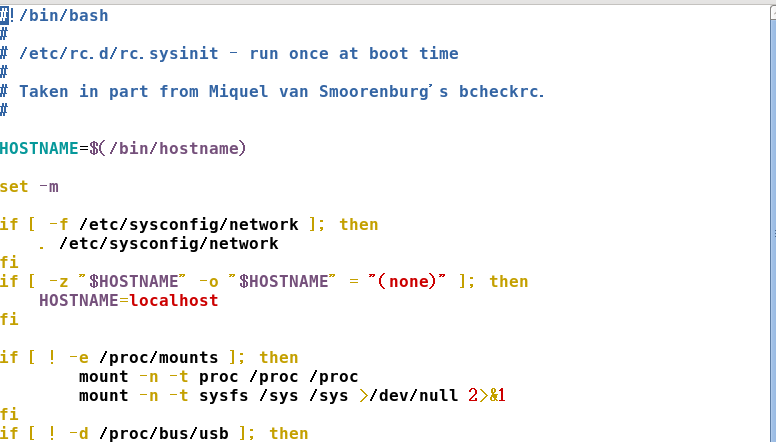
**切换运行级别: init 运行级别**

**查看运行级别: runlevel**



**第六步－－init进程执行rc.sysinit**

**在 设定了运行等级后，Linux系统执行的第一个文件就是/etc/rc.d/rc.sysinit脚本程序，它做的工作非常多，包括设定PATH、 设定网络配置（/etc/sysconfig/network）、启动swap分区、设定/proc等等。**



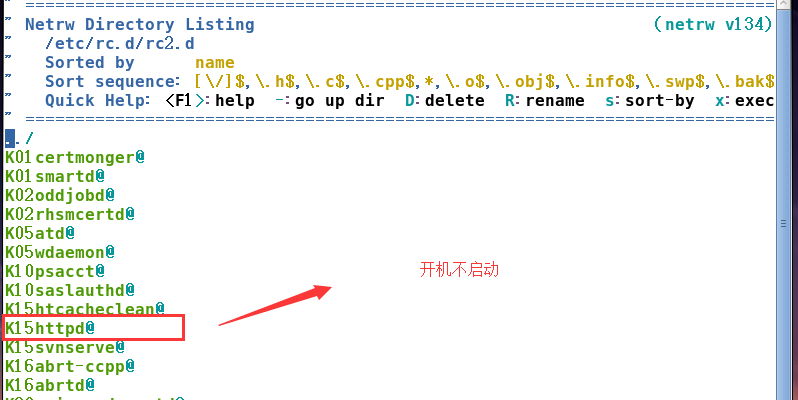


**第七步－－执行不同运行级别的脚本程序**

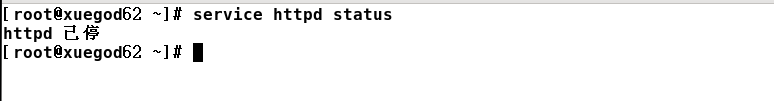
**根据运行级别的不同，系统会运行/etc/rc0.d到rc6.d中的相应的脚本程序，来完成相应的初始化工作和启动相应的服务。**

**运行级别 5 /etc/rc5.d**

**查看文件的内容 发现都是 以 K 或者是 S开头的信息**



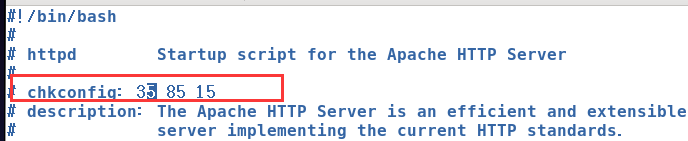
**S 开机要启动的服务 K 开机不启动的服务 中间的是服务的名称 优先级**



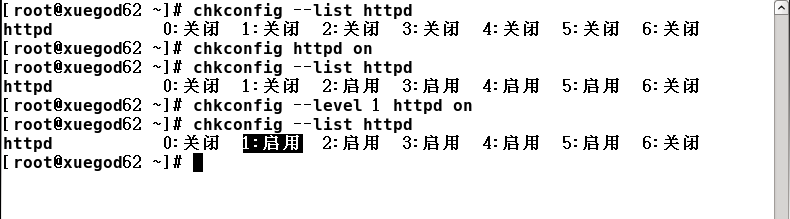
**案例2:**

**查看服务的脚本**

**/etc/init.d/httpd**



**chkconfig: 2345 服务运行级别 10 启动优先级 90 关闭优先级 -所有级别**



**第八步－－执行/etc/rc.d/rc.local**

**rc.local就是在一切初始化工作后，Linux留给用户进行个性化的地方。你可以把你想设置和启动的东西放到这里。**

**第九步－－执行/bin/login程序，进入登录状态**

**此时，系统已经进入到了等待用户输入username和password的时候了，你已经可以用自己的帐号登入系统了**

**常见的启动故障排除**

**每个人都可以,进入单用户模式，破解root密码。不安全**

**如果防止破解root密码？**

**grub加密**

**生成密文密码**



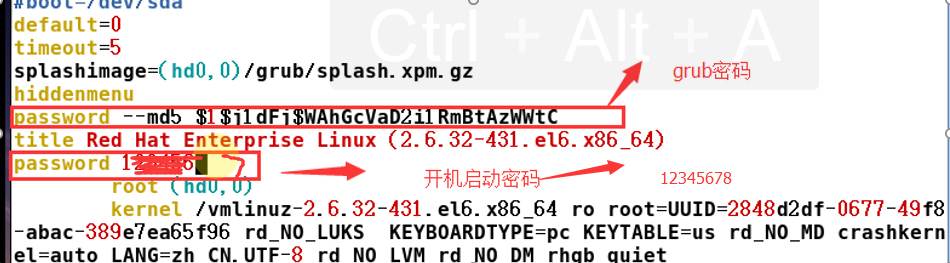
**grub配置文件 /boot/grub/grub.conf**

**注意加密后的密码的标点符号也得复制**

**根据位置不同**

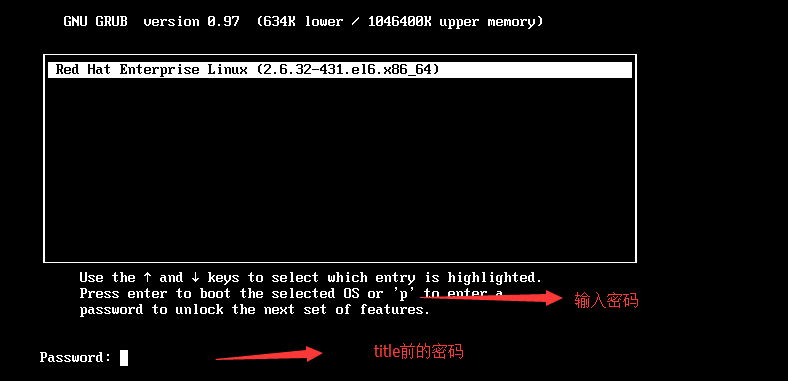
**title前:密码是对GRUB加密**

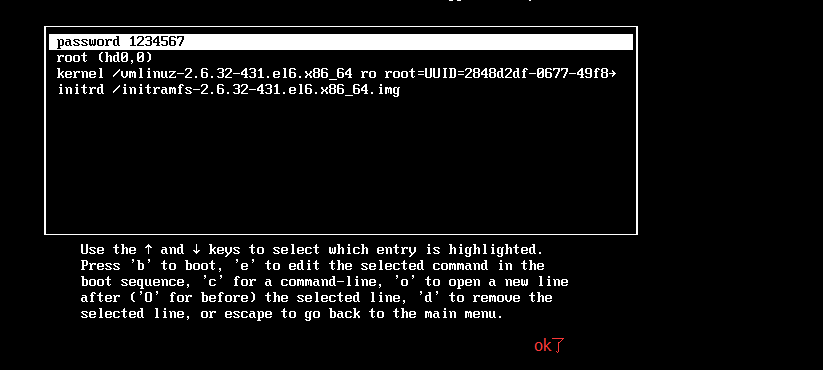
**title后:密码是对开机加密**



**重启测试：**

**加密后原先按e建编辑失效,必须按p 输入对应的密码**





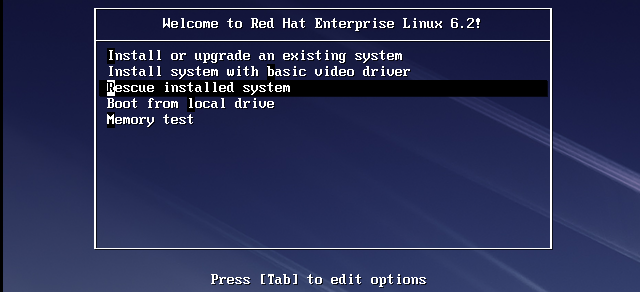
**这样就是防止随便有人破解ROOT密码**

**当我的服务器坏掉 进入一个基本的shell系统 将数据全部cp,便于恢复**

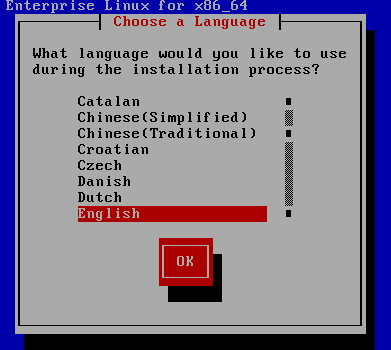
**进入救援模式：**

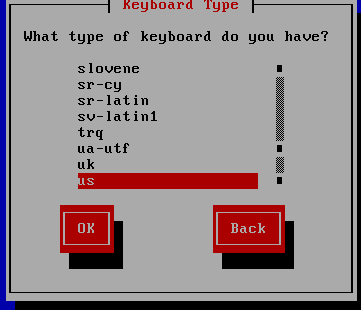
**开机BIOS cdrom 优先**

**救援模式**

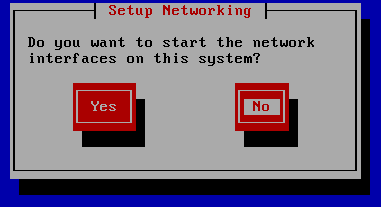


**语言**

**键盘类型**



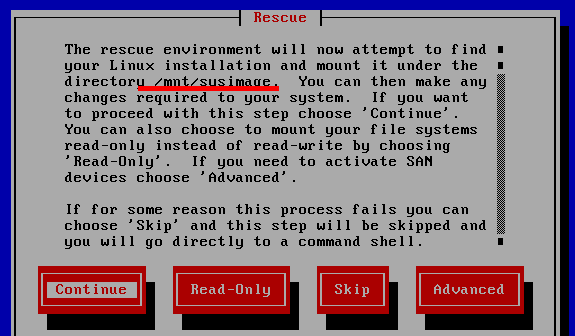
**是否启动网络 NO 不需要网络**



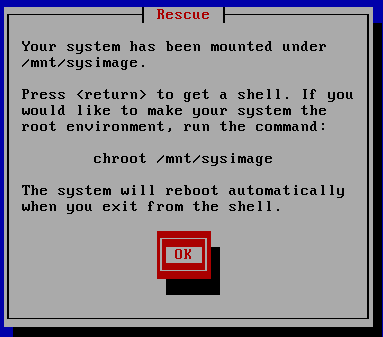
**挂载的目录**

**你的基本的系统是挂载到了 /mnt/sysimage 目录下,你可以去尽情的修改.**

**如果你要改变**



**挂载成功**

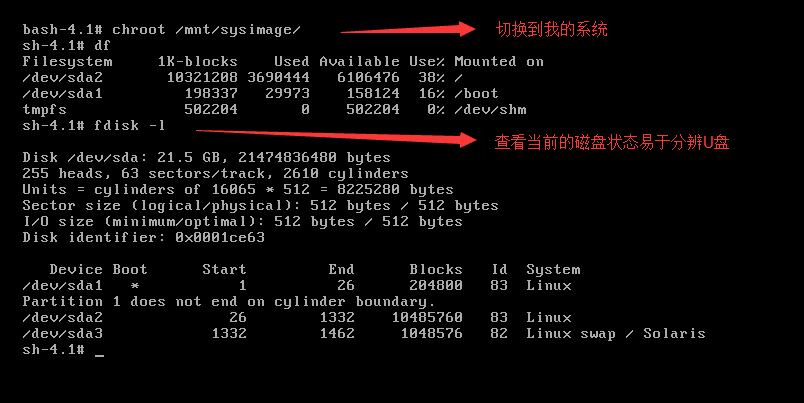


**启动一个shell环境**



换根

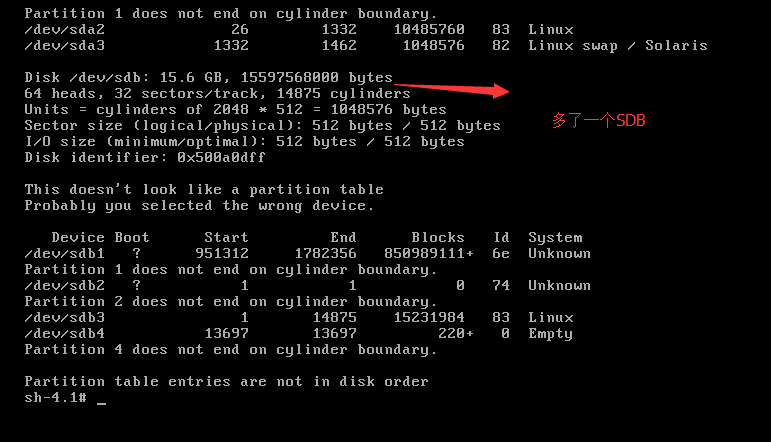
**chroot /mnt/sysimage # 切换文件系统根**



**插入U盘 最好是FAT格式**

**我的U盘是空的**



fdisk -l

加载模块 modprobe vfat 这是u盘的类型



**验证我的U盘**



**文件打不开的需要转换格式**

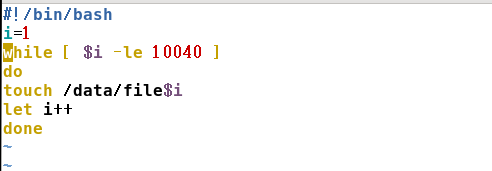
**磁盘的I节点耗尽**

**当一个文件系统的磁盘I节点耗尽,将无法继续在该分区中创建新的文件数据,导致故障出现;**

**添加磁盘分区 32M 格式化**

**创建挂载点 /data 挂载**

**编写脚本模拟i节点耗尽**



**注释**

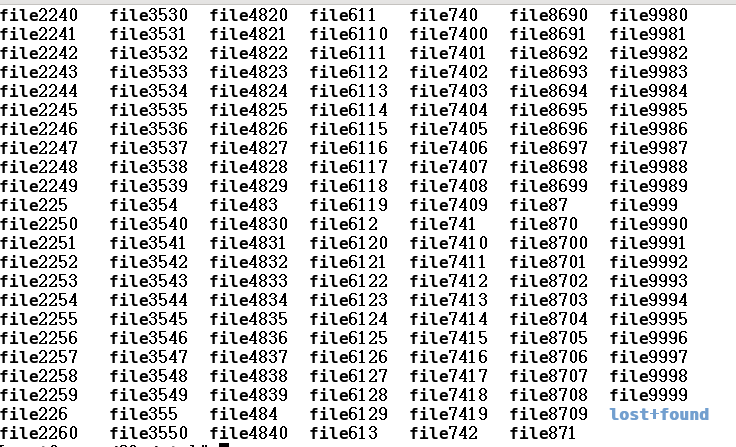
**i=1 赋值 i=1**

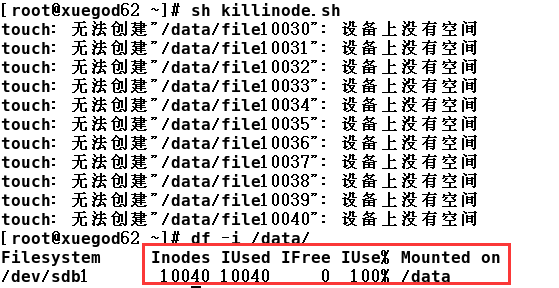
**while [条件 ] 条件循环语句 当i小于10040**

**do 执行的操作**

**创建file文件**

**let i++ 整数运算**

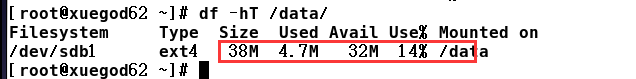




**手动创建文件**

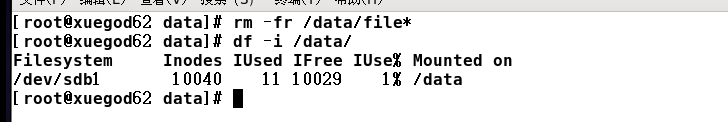


**有容量的**



**修复**

**找出该分区占用I节点大的细小文件 ,进行转移或删除 (建议为用户设置磁盘配额)**



**检测磁盘的坏道**

**分为两种**

**逻辑坏道:有软件操作不当造成,可以使用软件修复;**

**物理坏道:物理性损坏,更改磁盘分区的占用位置进行改善,排除掉坏的块**

**注意:如果硬盘出现坏道.若不及时修复或更换,坏道会越来越多,造成数据丢失.**

**情景:**

**读取磁盘中的数据时,磁盘设备发出异常的声响**

**访问某个文件时,反复读取却出错,提示文件损坏**

**新建立的分区无法执行格式化**

**使用磁盘的时候频繁死机**

**磁盘坏道检测**

**badblocks**

**-s 显示进度信息**

**-v 显示详情**

