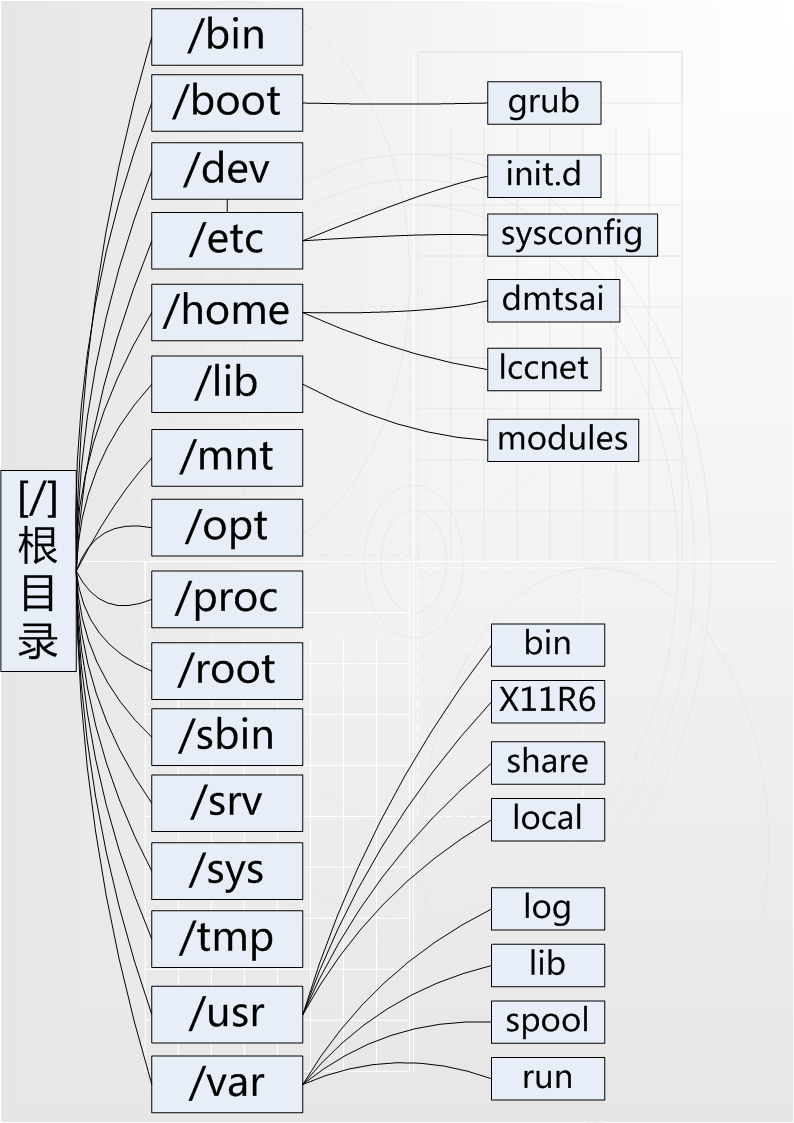
## linux目录结构



### 1、各级目录说明

**/ :** 根目录，一般根目录下只存放目录，不要存放文件，/etc、/bin、/lib、/sbin应该和跟目录放置在一个分区中

**/bin : /usr/bin :** 可执行二进制文件的目录，如常用的命令ls、tar、mv、cat等

**/boot :** 放置linux系统启动时用到的一些引导性文件。/boot/vmlinuz为linux的内核文件，以及/boot/grub。建议单独分区，分区大小100M即可

**/dev :** 存放linux系统下的设备文件，访问该目录下某个文件，相当于访问某个设备，常用的是挂载光驱mount /dev/cdrom /mnt

**/etc :** 系统配置文件存放的目录，不建议在此目录下存放可执行文件，重要的配置文件有/etc/inittab、/etc/fstab、/etc/init.d、/etc/X11、/etc/sysconfig、/etc/xinetd.d，修改配置文件之前记得备份，备注：/etc/X11存放与x windows有关的设置。

**/home :** 系统默认的用户家目录，新增用户账号时，用户的家目录都存放在此目录下，[~]表示当前用户的家目录，[~test]表示用户test的家目录。建议单独分区，并设置较大的磁盘空间，方便用户存放数据

**/lib : /usr/lib : /usr/local/lib :** 系统使用的函数库的目录，程序在执行过程中，需要调用一些额外的参数时需要函数库的协助，比较重要的目录为/lib/modules。

**/lost+fount :** 系统异常产生错误时，会将一些遗失的片段放置于此目录下，通常这个目录会自动出现在装置目录下。如加载硬盘于/disk中，此目录下就会自动产生目录/disk/lost+fount.

**/mnt : /media :** 光盘默认挂载点，通常光盘挂载于/mnt/cdrom下。

**/opt :** 给主机额外安装软件所放置的目录。

**/proc :** 此目录的数据都在内存中，如系统核心、外部设备、网络状态，由于数据都存放在内存中，所以不占用磁盘空间，比较重要的目录有/proc/cpuinfo、/proc/interrupts、/proc/dma、/proc/ioports、/proc/net/\*等

**/root :** 系统管理员root的家目录，系统第一启动的分区为/，所以最好将/root和/放置在一个分区下。

**/sbin : /usr/sbin : /usr/local/sbin :** 放置系统管理员使用的可执行命令，如fdisk、shutdown、mount等。与/bin不同的是，这几个目录是给系统管理员root使用的命令，一般用户只能查看而不能设置和使用

**/tmp :** 一般用户或正在执行的程序临时存放文件的目录，任何人都可以访问，重要数据不可以放置在此目录下

**/srv :** 服务启动之后需要访问的数据目录，如www服务需要访问的网页数据存放在/srv/www内

**/usr :** 应用程序存放目录，**/usr/bin** 存放应用程序，**/usr/share**存放共享数据，**/usr/lib** 存放不能直接运行却是许多程序运行所必需的一些函数库文件，**/usr/local** 存放软件升级包，**/usr/share/doc** 系统说明文件存放目录，**/usr/share/man** 程序说明文件存放的目录，使用[man ls]命令时会查询出/usr/share/man/man1/ls.1.gz的内容，建议单独分区，设置较大的磁盘空间

**/var :** 放置系统执行过程中经常变化的文件，如随时更改的日志文件/var/log，**/var/log/message**:存放所有的登录文件的目录，**/var/spool/mail**:存放邮件的目录，**/var/run**:程序或服务启动后，其PID存放在改目录下，建议单独分区，设置较大的磁盘空间

#### 1.1、/dev 目录

dev是设备(device)的英文缩写。/dev这个目录对所有的用户都十分重要。因为在这个目录中包含了所有Linux系统中使用的外部设备。但是这里并不是放的外部设备的驱动程序，这一点和windows,dos操作系统不一样。它实际上是一个访问这些外部设备的端口。我们可以非常方便地去访问这些外部设备，和访问一个文件，一个目录没有任何区别。  
Linux沿袭Unix的风格，将所有设备认成是一个文件。

设备文件分为两种：块设备文件(b)和字符设备文件(c)  
设备文件一般存放在/dev目录下，对常见设备文件作如下说明：

/dev/hd[a-t]：IDE设备  
/dev/sd[a-z]：SCSI设备

/dev/fd[0-7]：标准软驱  
/dev/md[0-31]：软raid设备  
/dev/loop[0-7]：本地回环设备  
/dev/ram[0-15]：内存  
/dev/null：无限数据接收设备,相当于黑洞  
/dev/zero：无限零资源  
/dev/tty[0-63]：虚拟终端  
/dev/ttyS[0-3]：串口  
/dev/lp[0-3]：并口  
/dev/console：控制台  
/dev/fb[0-31]：framebuffer  
/dev/cdrom => /dev/hdc  
/dev/modem => /dev/ttyS[0-9]  
/dev/pilot => /dev/ttyS[0-9]  
/dev/random：随机数设备  
/dev/urandom：随机数设备

(PS：随机数设备，后面我会再写篇博客总结一下)

/dev目录下的节点是怎么创建的?  
devf或者udev会自动帮你创建得。

kobject是sysfs文件系统的基础，udev通过监测、检测sysfs来获取新创建的设备的。

#### 1.2、/etc 目录

包含很多文件.许多网络配置文件也在/etc 中.   
**/etc/rc   or /etc/rc.d   or /etc/rc\*.d** 启动、或改变运行级时运行的scripts或scripts的目录.  
**/etc/passwd** 用户数据库，其中的域给出了用户名、真实姓名、家目录、加密的口令和用户的其他信息.   
**/etc/fstab** 启动时mount -a命令(在/etc/rc 或等效的启动文件中)自动mount的文件系统列表. Linux下，也包括用swapon -a启用的swap区的信息.  
**/etc/group** 类似/etc/passwd ，但说明的不是用户而是组.   
**/etc/inittab** init 的配置文件.   
**/etc/issue** getty 在登录提示符前的输出信息.通常包括系统的一段短说明或欢迎信息.内容由系统管理员确定.   
**/etc/motd** Message Of The Day，成功登录后自动输出.内容由系统管理员确定.经常用于通告信息，如计划关机时间的警告.   
**/etc/mtab** 当前安装的文件系统列表.由scripts初始化，并由mount 命令自动更新.需要一个当前安装的文件系统的列表时使用，例如df 命令.   
**/etc/shadow** 在安装了影子口令软件的系统上的影子口令文件.影子口令文件将/etc/passwd 文件中的加密口令移动到/etc/shadow 中，而后者只对root可读.这使破译口令更困难.   
**/etc/login.defs** login 命令的配置文件.   
**/etc/printcap** 类似/etc/termcap ，但针对打印机.语法不同.   
**/etc/profile , /etc/csh.login , /etc/csh.cshrc** 登录或启动时Bourne或C shells执行的文件.这允许系统管理员为所有用户建立全局缺省环境.   
**/etc/securetty** 确认安全终端，即哪个终端允许root登录.一般只列出虚拟控制台，这样就不可能(至少很困难)通过modem或网络闯入系统并得到超级用户特权.   
**/etc/shells** 列出可信任的shell.chsh 命令允许用户在本文件指定范围内改变登录shell.提供一台机器FTP服务的服务进程ftpd 检查用户shell是否列在 /etc/shells 文件中，如果不是将不允许该用户登录.   
**/etc/sysconfig** 网络配置相关目录

#### 1.3、/proc 目录

**/proc/cmdline** 加载 kernel 时所下达的相关参数！查阅此文件，可了解系统是如何启动的！

**/proc/cpuinfo** 本机的 CPU 的相关资讯，包含时脉、类型与运算功能等

**/proc/devices** 这个文件记录了系统各个主要装置的主要装置代号，与 mknod 有关呢！

**/proc/filesystems** 目前系统已经加载的文件系统罗！

**/proc/interrupts** 目前系统上面的 IRQ 分配状态。

**/proc/ioports** 目前系统上面各个装置所配置的 I/O 位址。

**/proc/kcore** 这个就是内存的大小啦！好大对吧！但是不要读他啦！

**/proc/loadavg** 还记得 top 以及 uptime 吧？没错！上头的三个平均数值就是记录在此！

**/proc/meminfo** 使用 free 列出的内存资讯，嘿嘿！在这里也能够查阅到！

**/proc/modules** 目前我们的 Linux 已经加载的模块列表，也可以想成是驱动程序啦！

**/proc/mounts** 系统已经挂载的数据，就是用 mount 这个命令呼叫出来的数据啦！

**/proc/swaps** 到底系统挂加载的内存在哪里？呵呵！使用掉的 partition 就记录在此啦！

**/proc/partitions** 使用 fdisk -l 会出现目前所有的 partition 吧？在这个文件当中也有纪录喔！

**/proc/pci** 在 PCI 汇流排上面，每个装置的详细情况！可用 lspci 来查阅！

**/proc/uptime** 就是用 uptime 的时候，会出现的资讯啦！

**/proc/version** 核心的版本，就是用 uname -a 显示的内容啦！

**/proc/bus/\*** 一些汇流排的装置，还有 U盘 的装置也记录在此喔！

#### 1.4、/usr 目录

/usr 文件系统经常很大，因为所有程序安装在这里. /usr 里的所有文件一般来自Linux distribution；本地安装的程序和其他东西在/usr/local 下.这样可能在升级新版系统或新distribution时无须重新安装全部程序.

**/usr/etc** 存放设置文件

**/usr/games** 存放游戏和教学文件

**/usr/include** 存放C开发工具的头文件

**/usr/share** 存放结构独立的数据

**/usr/bin**几乎所有用户命令.有些命令在/bin 或/usr/local/bin 中.

**/usr/sbin**根文件系统不必要的系统管理命令，例如多数服务程序.

**/usr/share/man , /usr/share/info , /usr/share/doc** 手册页、GNU信息文档和各种其他文档文件.

**/usr/include** C编程语言的头文件.为了一致性这实际上应该在/usr/lib 下，但传统上支持这个名字.

**/usr/lib** 程序或子系统的不变的数据文件，包括一些site-wide配置文件.名字lib来源于库(library); 编程的原始库存在/usr/lib 里.

**/usr/local** 本地安装的软件和其他文件放在这里.

**/usr/src** 存放程序的源代码

#### 1.5、/var 目录

包括系统一般运行时要改变的数据.每个系统是特定的，即不通过网络与其他计算机共享.

**/var/catman** 当要求格式化时的man页的cache.man页的源文件一般存在/usr/man/man\* 中；有些man页可能有预格式化的版本，存在/usr/man/cat\* 中.而其他的man页在第一次看时需要格式化，格式化完的版本存在/var/man 中，这样其他人再看相同的页时就无须等待格式化了. (/var/catman 经常被清除，就象清除临时目录一样.)

**/var/lib** 系统正常运行时要改变的文件. 

**/var/local** /usr/local 中安装的程序的可变数据(即系统管理员安装的程序).注意，如果必要，即使本地安装的程序也会使用其他/var 目录，例如/var/lock .

**/var/lock** 锁定文件.许多程序遵循在/var/lock 中产生一个锁定文件的约定，以支持他们正在使用某个特定的设备或文件.其他程序注意到这个锁定文件，将不试图使用这个设备或文件.

**/var/log** 各种程序的Log文件，特别是login  (/var/log/wtmp log所有到系统的登录和注销) 和syslog (/var/log/messages 里存储所有核心和系统程序信息. /var/log 里的文件经常不确定地增长，应该定期清除.

**/var/run** 保存到下次引导前有效的关于系统的信息文件.例如， /var/run/utmp 包含当前登录的用户的信息.

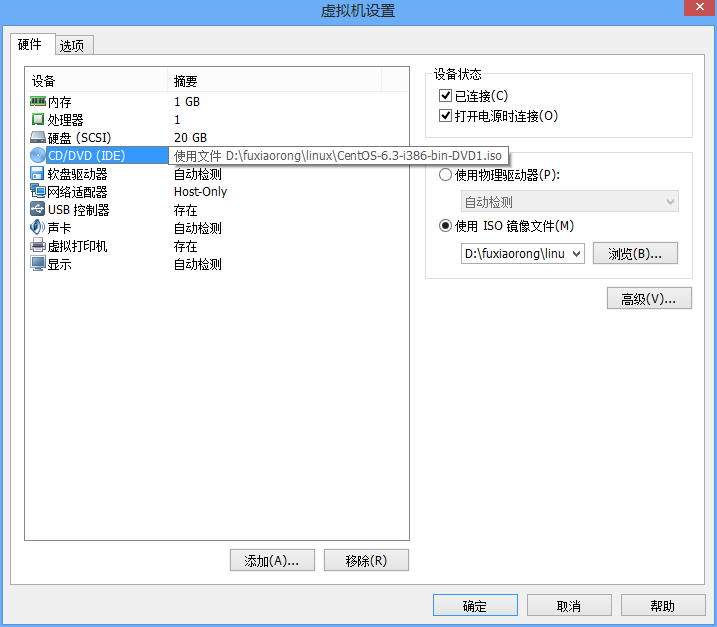
**/var/spool** mail, news, 打印队列和其他队列工作的目录.每个不同的spool在/var/spool 下有自己的子目录，例如，用户的邮箱在/var/spool/mail 中.

**/var/tmp**比/tmp 允许的大或需要存在较长时间的临时文件. (虽然系统管理员可能不允许/var/tmp 有很旧的文件.)

## 虚拟机安装CentOS 6.3

### 步骤

#### 虚拟机挂载ISO文件



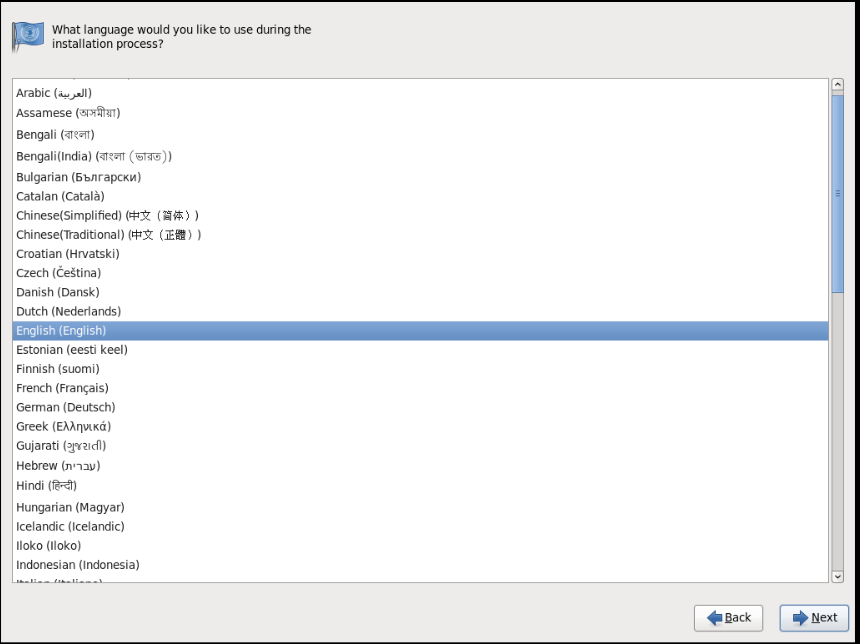
#### 启动虚拟机，进入欢迎页面，选择第一项，按【Enter】键进入



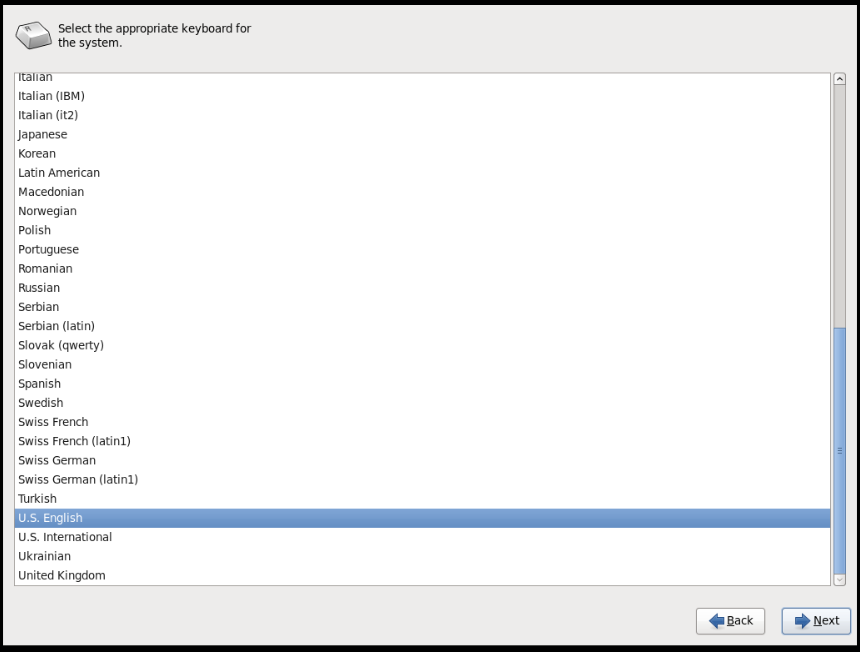
#### 1.3、接下来会提示时候效验安装介质，大多数情况下当前介质工作得不错，不需要做此效验操作，按【TAB】键选择[Skip]跳过，直接安装。



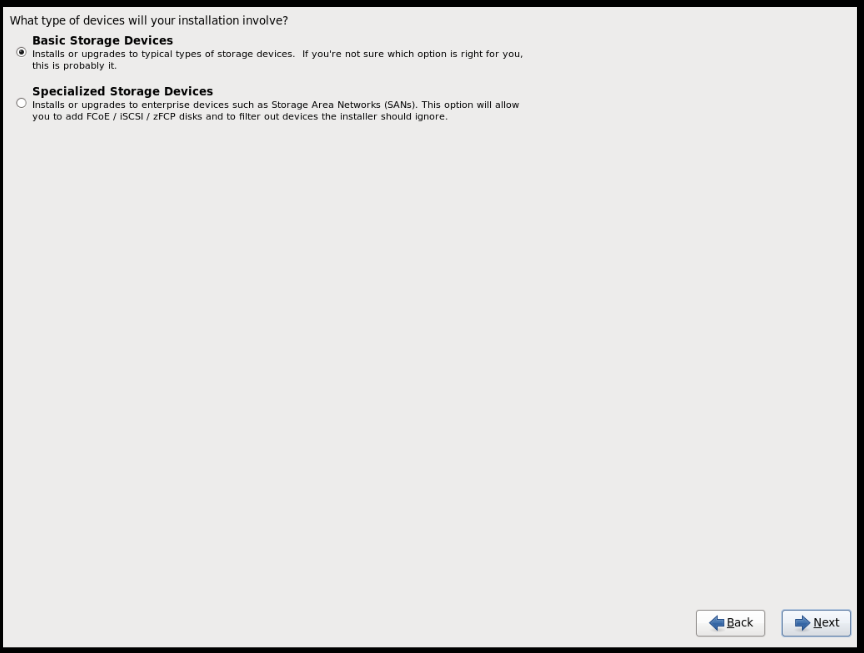
#### 1.4、之后是选择安装语言，这里选择[English]英语版，并单击Next继续。



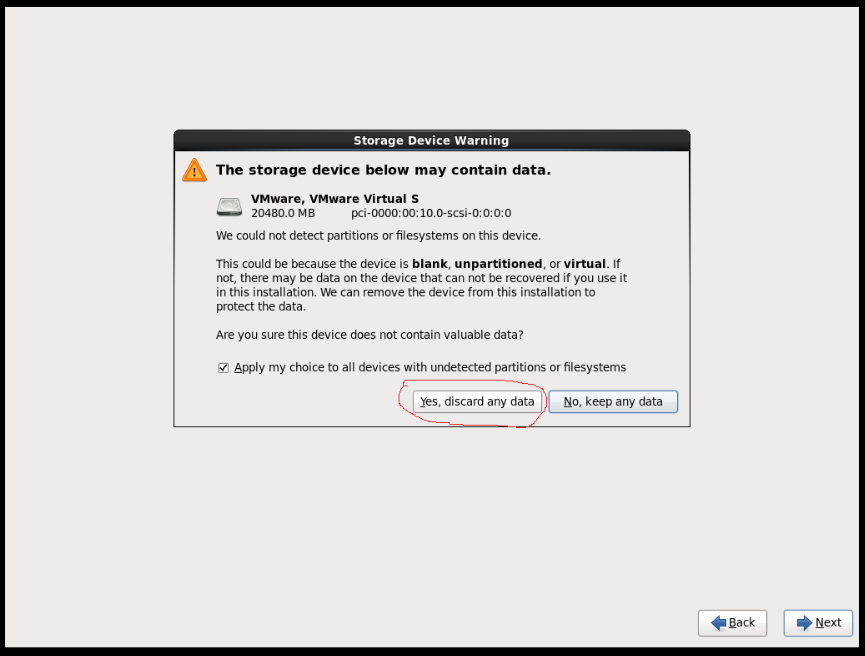
#### 1.5、选择键盘类型。在选择键盘布局的屏幕中列出了支持不同的键盘布局，最通用的选项是U.S.English键盘布局，选择U.S.English 并单击Next继续。



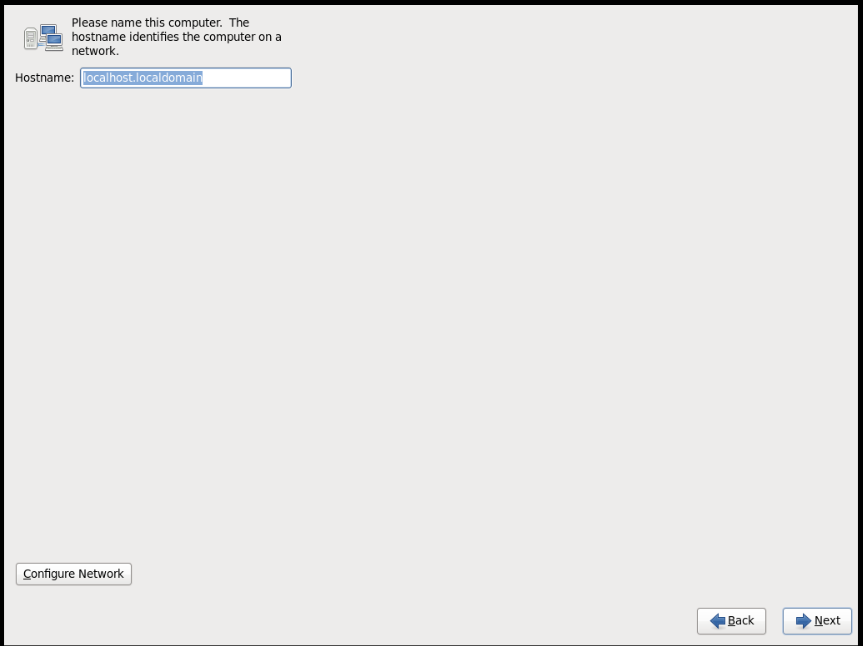
#### 1.6、选择安装设备驱动器如图所示。选择基本第一项基本设备存储设备驱动器即可，并点击Next继续。



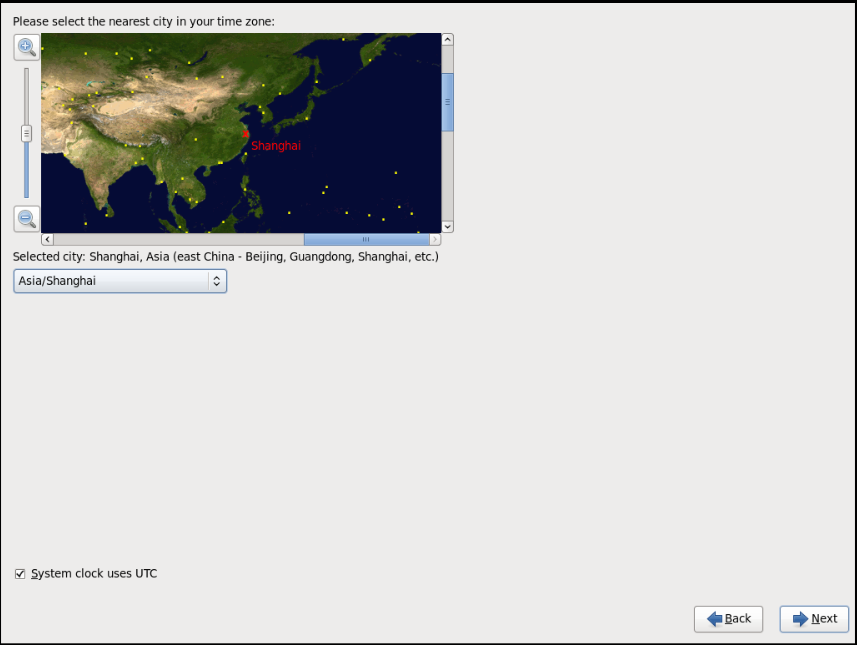
#### 检测到磁盘中可能有数据，是否放弃任何数据，这里选择YES。



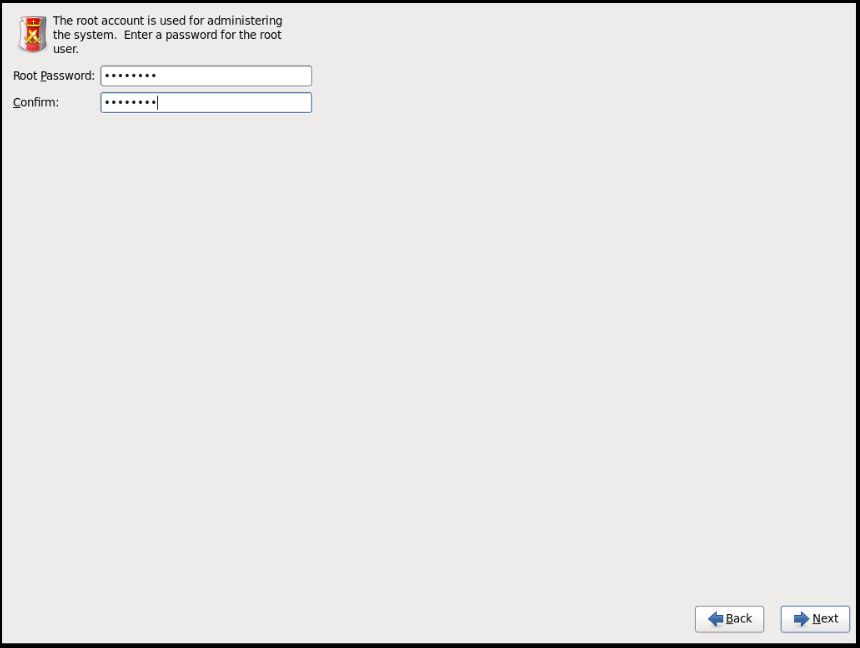
#### 1.8、设置服务器名称见图所示，默认显示“localhost.localdomain”,这个服务器名称可以修改成自己想要的名称，如“dns.server”可以表示为DNS服务器。

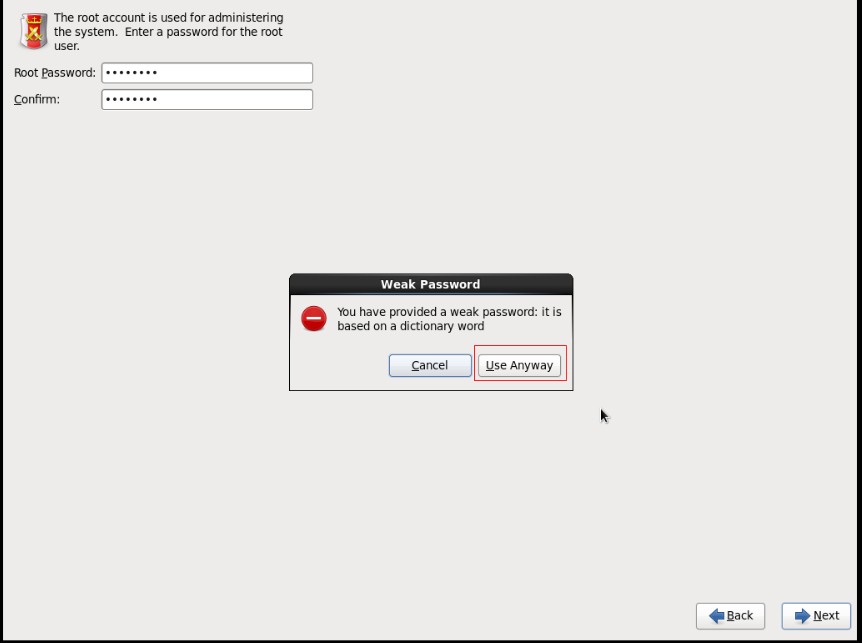


#### 1.9、选择时区见图所示，我这里选择是“Asia/Shanghai”，选择中文安装时默认选择“亚洲/上海”，英文安装时默认是美国时区，所以需要重新选择。

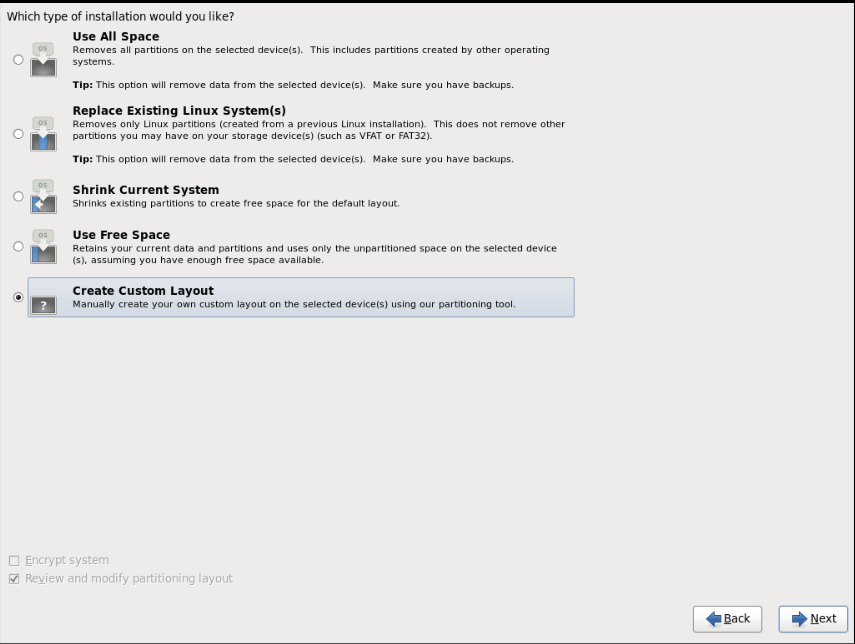


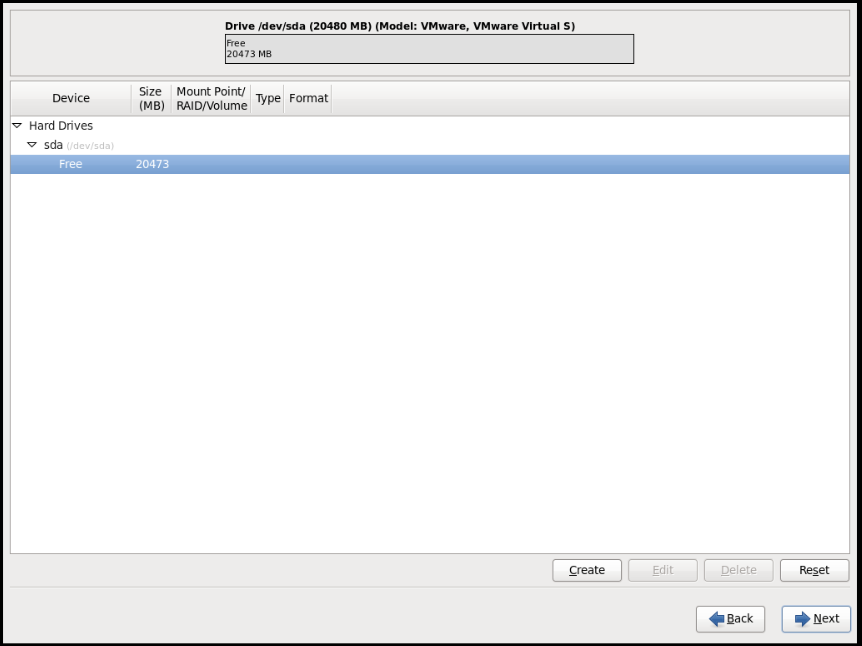
#### 1.10、设置root超级管理员密码见图。由于root用户的权限很大，设置的密码最好要包含数字、字母和特殊符号且密码长度不要太短，否则会有如下图的提示，当然也可以单击Use Anyway坚持你设置的密码。

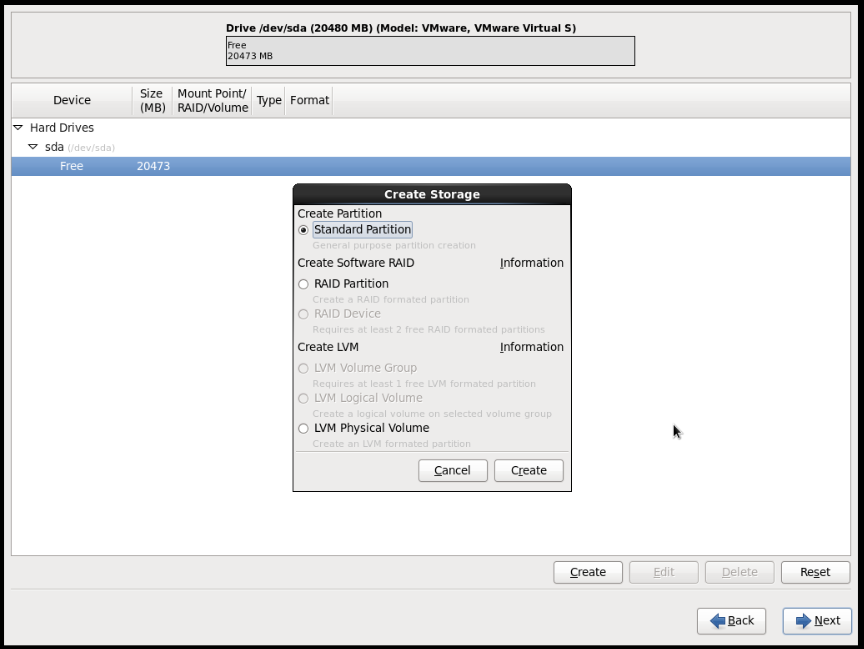


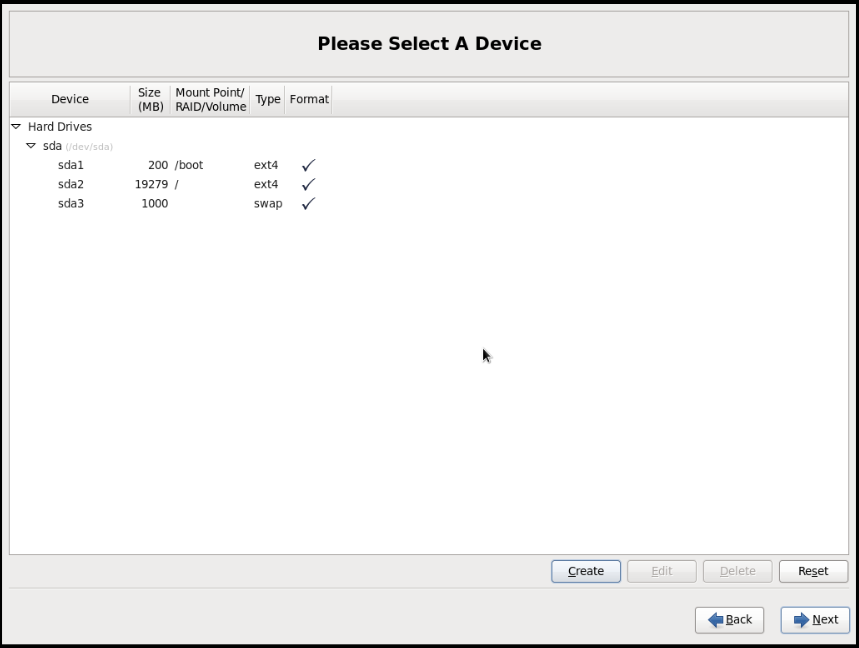


#### 1.11、磁盘分区方案，选择“Create Custom Layout”自定义分区见图。linux简单地可以分成三个分区：启动分区“boot”、交换分区“swap”和根分区“/”。启动分区设置100M空间即可，交换分区一般设置为服务器内存的2倍左右，其它剩余的磁盘空间全部分配给根分区，如图所示。

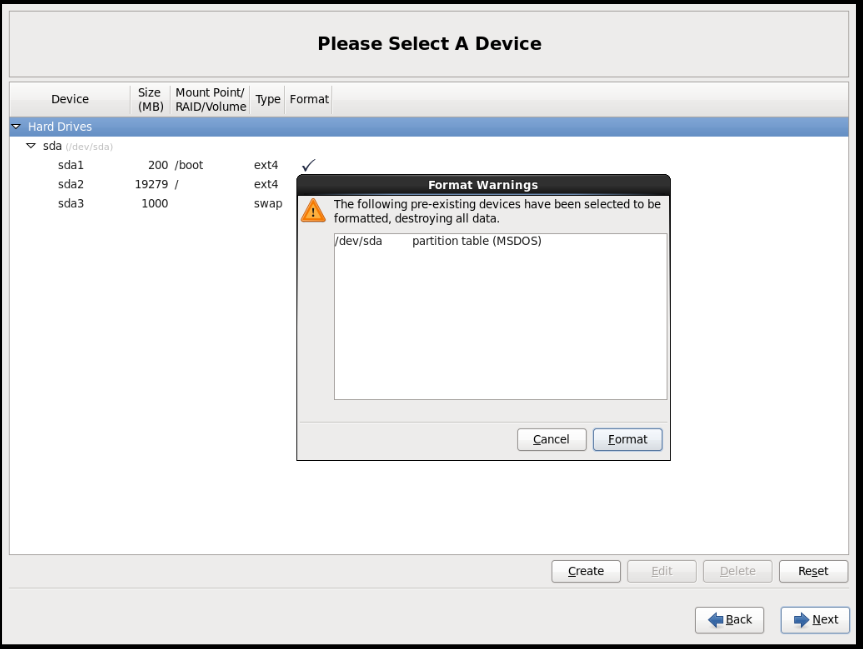




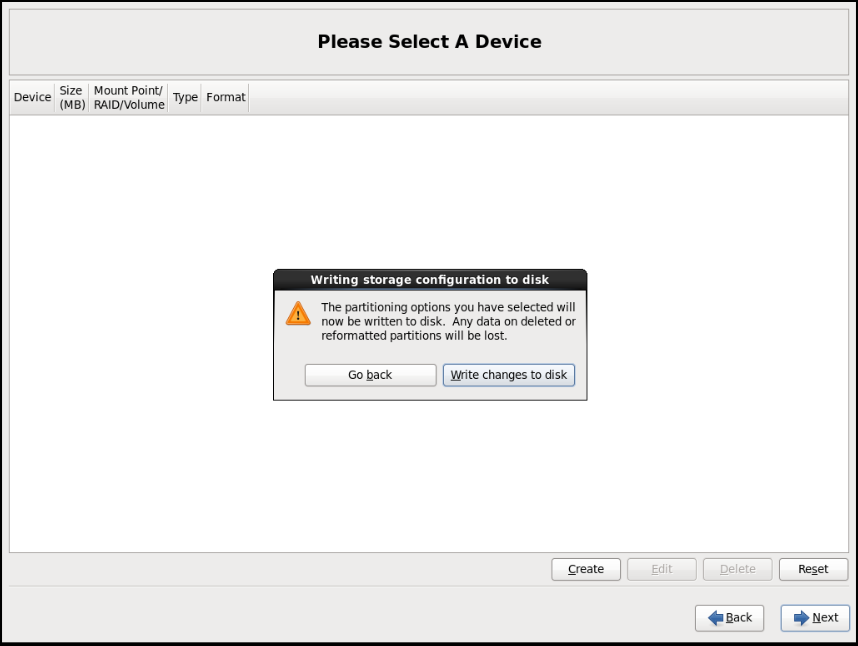


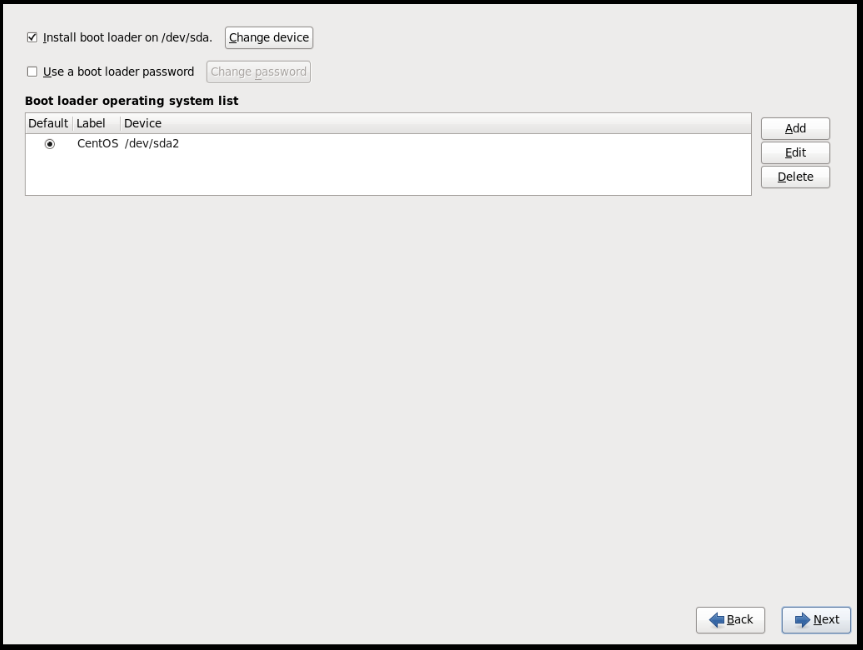


#### 接下来就要对磁盘中的数据进行格式化，如图：

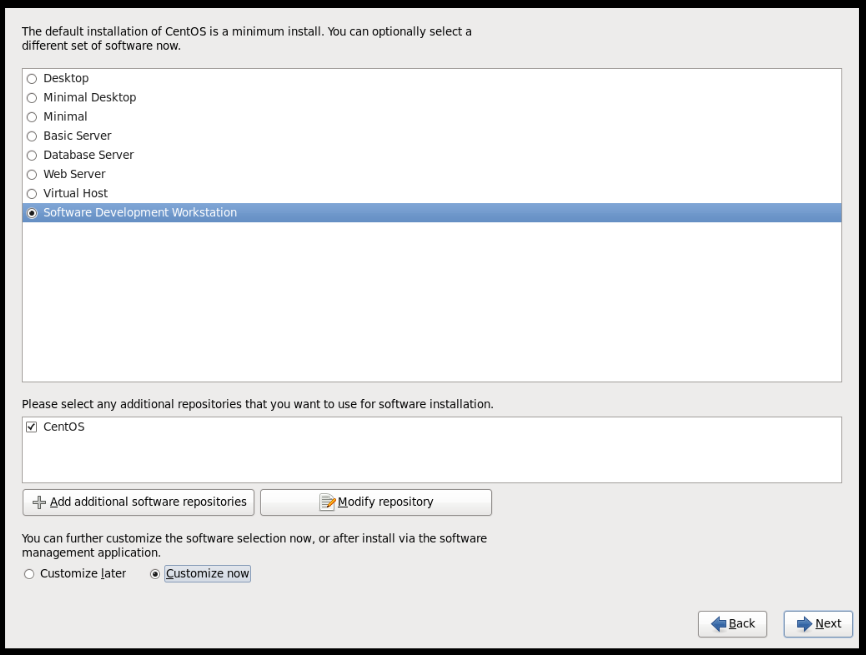


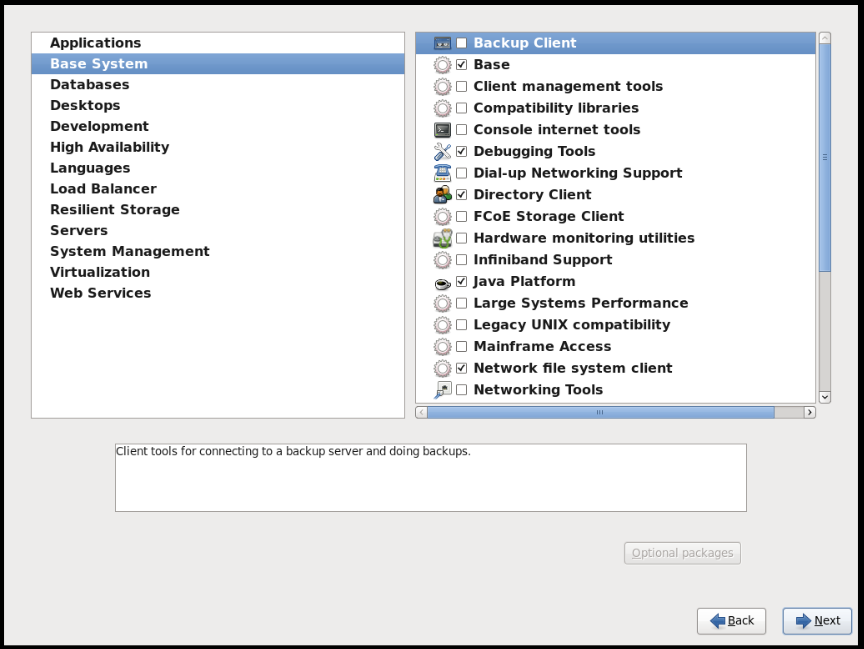
#### 1.13、将存储配置写入磁盘进行保存，选择“Write changes to disk”见下图。





#### 1.14、软件包的安装选项见下图。安装软件开发工作空间选择“Software Development Workstation”，选中“Custom now”自定义一些常用的软件包见图进行安装，我是把eclipse和jdk勾选掉了，这些由自己后面来安装，点击Next继续。

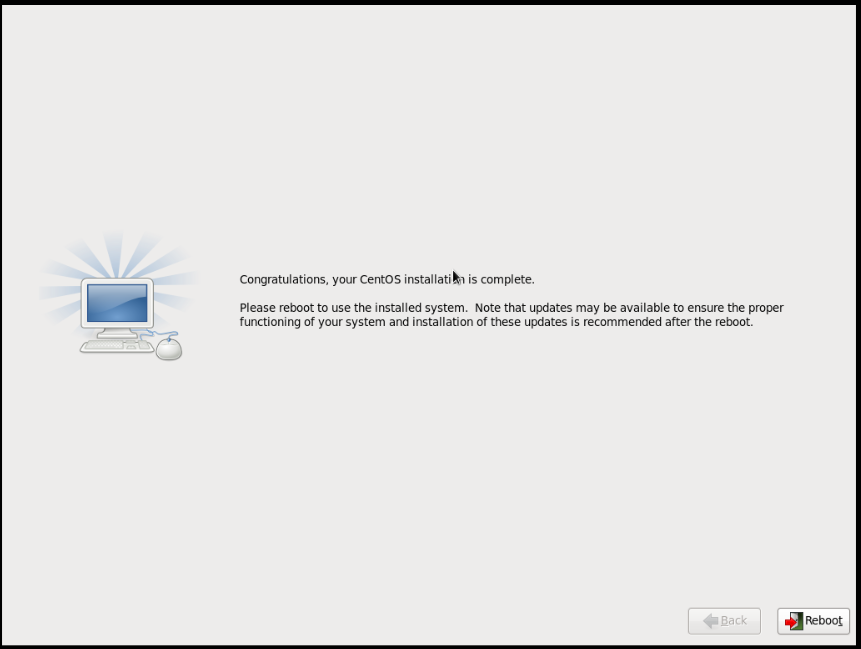




#### 进入系统软件安装界面了



#### 1.16、安装完成后，点击Reboot重启linux，这时可以退出系统盘了，见下图

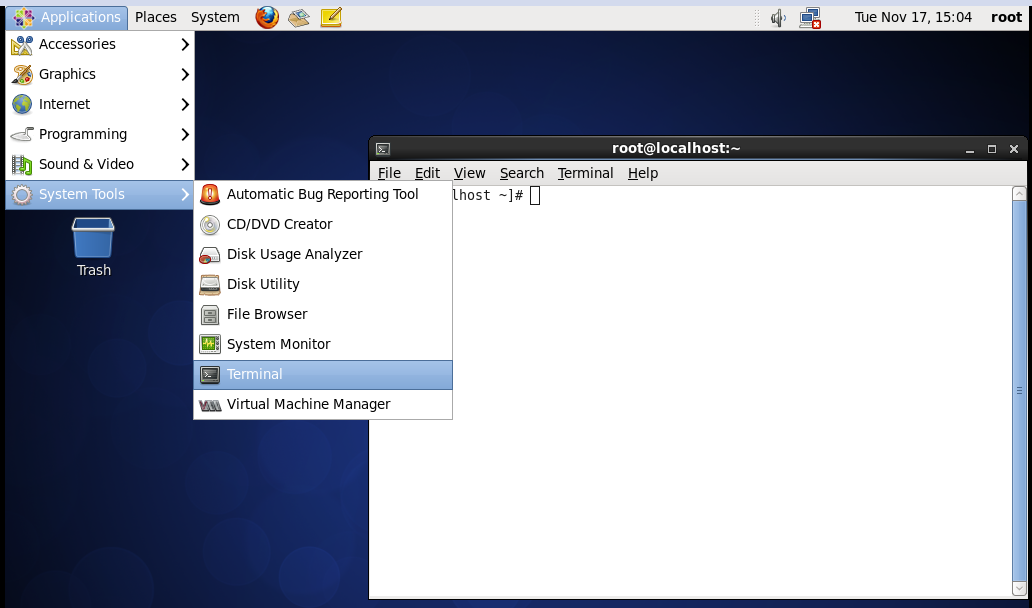


#### 1.17、服务器重启后，要进行非root用户的创建，日期时间的调整等设置，到此Linux 操作系统安装完成。

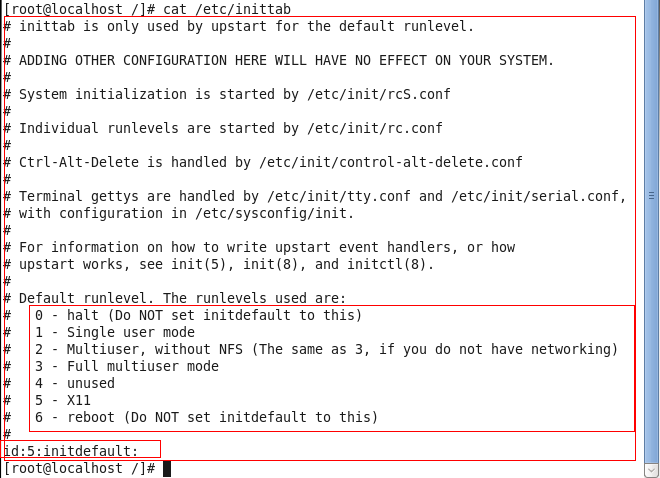
### 安装后必做事情之一修改开机模式

#### 用root用户登录

#### 打开命令行窗口



#### 2.3、linux开机模式是通过读取/etc/inittab文件中的设置来进行引导性开机，默认是5[图形化界面]，如图：



其中各个数字的含义是：

0：直接关机

1：单用户模式

2：多用户无网络模式，无NFS模式

3：多用户有网络模式

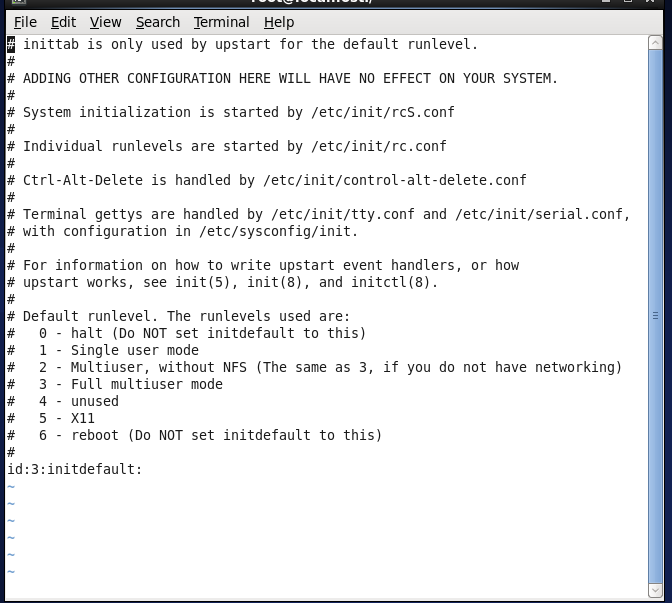
4：系统保留功能，暂无用

5：图形化界面模式

6：重启

如果需要修改成为开机启动命令行模式，则需要把id:5:initdefault:修改为id:3:initdefault:，使用vi对/etc/inittab文件进行修改，最后保存，重启linux即可。



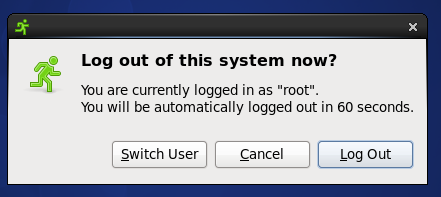


#### 2.4、命令行模式和图形化模式如何切换？

命令行模式→图形化模式：在命令行中输入startx即可

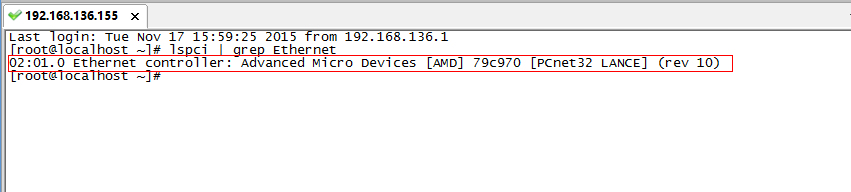


图形化模式→命令行模式：退出Log Out即可

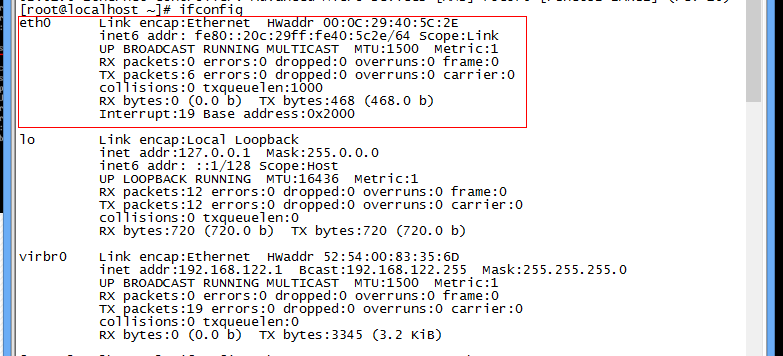


### 安装后必做事情之二配置网络信息

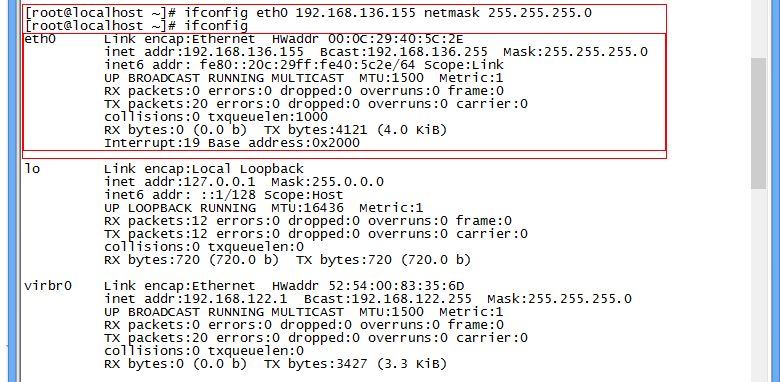
#### 3.1、用root用户登录，然后使用lspci | grep Ethernet命令检查计算机上的以太网 卡设备是否被内核检测到。这里小编要提醒大家，在默认的情况下，只有root也 就是根用户才有权限进行网络配置。红框内就是检测到的网卡信息。



#### 3.2、然后使用“ifconfig”命令查看系统已经识别并已激活的网络接口。如果没有找到可以用“ ifconfig -a ”命令找到所有已识别包括未激活的网络接口，然后使用”ifconfig 网卡名 up“激活。可以看出小编的CentOS6.3系统上有一块已激活的eth0网卡，下面就对这块网卡进行设置使它能够发挥作用。



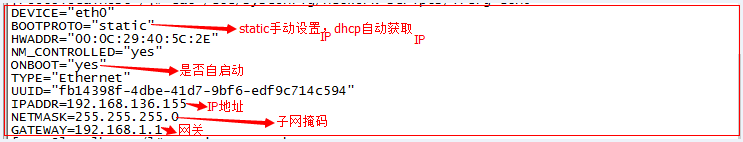
#### 3.3、网络配置之一如何临时指定IP地址。在命令行输入“ifconfig 网卡名 IP地址 netmask 子网掩码”就可以临时指定网卡的IP地址，但重启后失效。相对于上一张图，小编的IP临时设置已经生效。



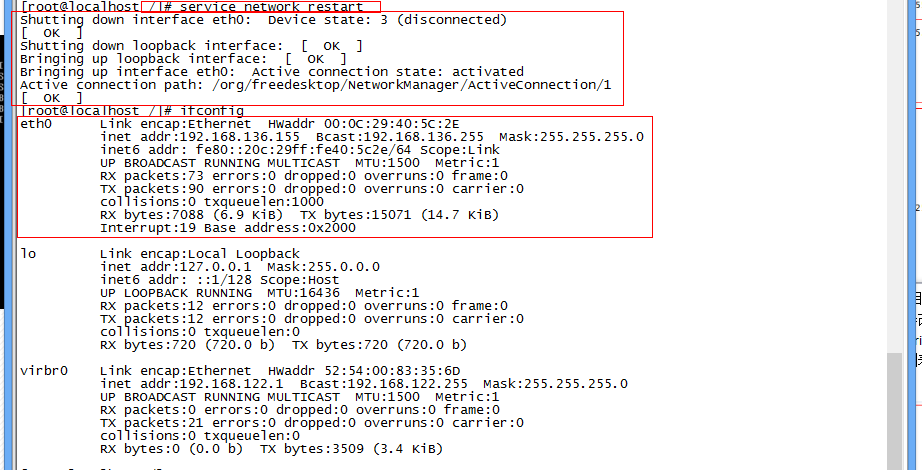
#### 3.4、网络配置之二如何永久改变IP地址。ip地址相关信息是在 /etc/sysconfig/network-scripts/ifcfg-eth0的网卡配置文件中，因此修改网卡配置文件 即可以永久改变ip地址，使用命令“ vi /etc/sysconfig/network-scripts/ifcfg-eth0 ” 编辑网卡eth0配置文件，同时提醒如果“ BOOTPROTO=dhcp ”则表示为自动获取 ip地址，那么IPADDR、NETMASK和GATWAY都可以不写。

BOOTPROTO=static 静态IP

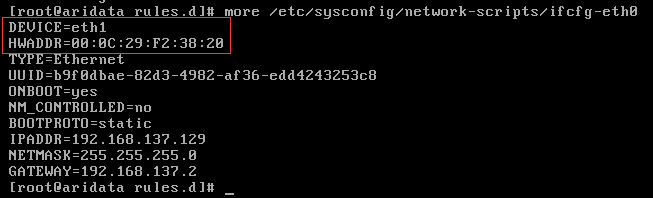
BOOTPROTO=dhcp 动态IP  
BOOTPROTO=none 无（不指定）



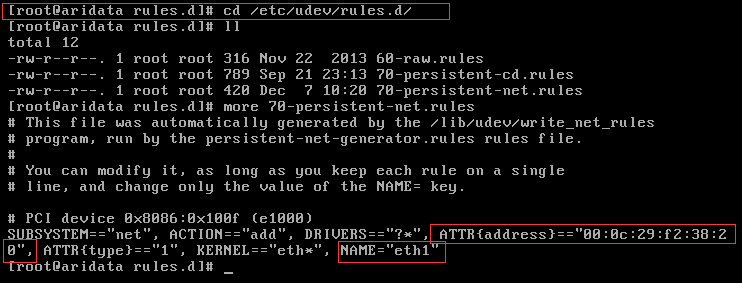
#### 3.5、设置完成后，使用“ sevice network restart ”命令重新启动network服务，是配置文件生效，四个状态都必须是OK，否则说明设置有问题，需重新修改设置。



#### 3.6、问题：报Device eth0 does not seem to be present, delaying initialization.错误的解决方案

删除/etc/udev/rules.d/70-persistent-net.rules文件，之后重启。重启之后，如果network服务能正常启动了，那就可以了。否则把/etc/sysconfig/network-scripts/ifcfg-eth0中的DEVICE和HWADDR 

修改成为新生成/etc/udev/rules.d/70-persistent-net.rules里面的内容



### 4、安装后必做事情之二配置防火墙

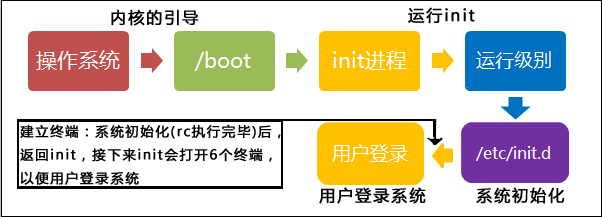
#### 4.1、基本命令

查询防火墙状态:  
[root@localhost ~]# service   iptables status  
停止防火墙:  
[root@localhost ~]# service   iptables stop  
启动防火墙:  
[root@localhost ~]# service   iptables start  
重启防火墙:  
[root@localhost ~]# service   iptables restart  
永久关闭防火墙:  
[root@localhost ~]# chkconfig   iptables off   
永久关闭后启用:  
[root@localhost ~]# chkconfig   iptables on

### 5、linux系统启动和关机

##### 5.1、启动linux系统五个阶段

* 内核的引导
* 运行init
* 系统初始化
* 建立终端
* 用户登录系统



##### 5.2、关机

1、正确的关机流程：**sync > shutdown > reboot > halt**

2、可以运行如下命令关机：

|  |
| --- |
| **sync** 将数据由内存同步到硬盘中，不管重启系统还是关闭系统，都应要运行**sync**命令  **shutdown –h now** 立马关机  **shutdown –h 20:35** 系统会在今天20:35关机  **shutdown –h +10** 十分钟后关机  **shutdown –h 10 ‘this server will shutdown after ten mins’** 这个命令告诉大家，计算机将在十分钟后关机，并且显示在登录用户的当前屏幕中  **shutdown –r now** 系统立马重启  **shutdown –r +10** 系统十分钟后重启  **reboot** 就是重启，等同于 **shutdown –r now**  **halt** 关闭系统，等同于**shutdown –h now** 和 **poweroff** |

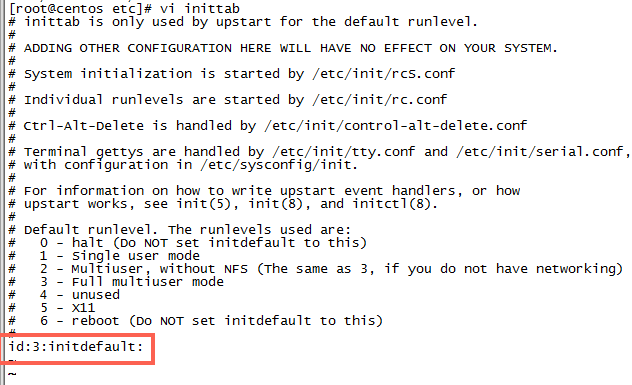
### 6、运行级别

##### 6.1、七个运行级别（runlevel）

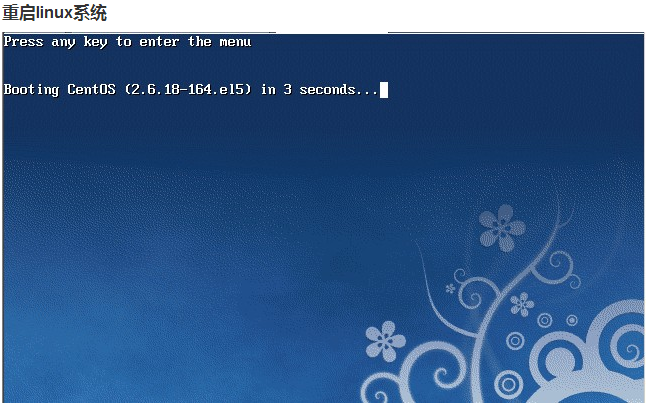
* 0：系统停机状态，系统默认运行级别不能设为0，否则不能正常启动
* 1：单用户工作状态，root权限，用户系统维护，静止远程登录
* 2：多用户状态（没有NFS）
* **3：完全的多用户状态（有NFS），登录后进入控制台命令行模式（常用）**
* 4：系统未使用，保留
* **5：X11控制台，登录后进入图形GUI模式（常用）**
* 6：系统正常关闭并重启，默认运行级别不能设为6，否则不能正常启动

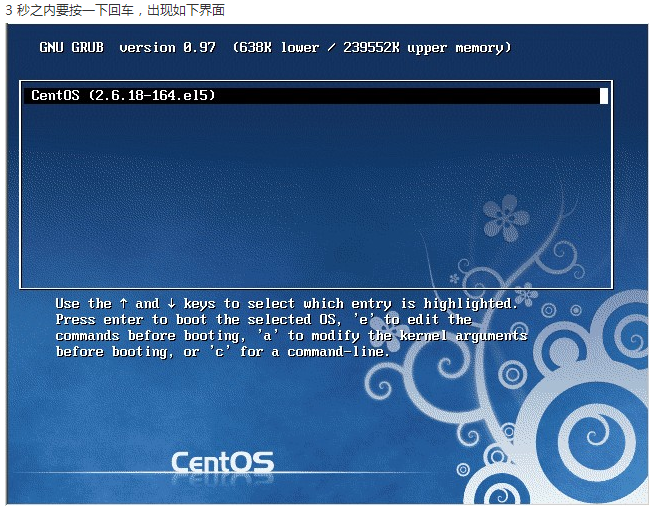
##### 6.2、修改系统运行级别

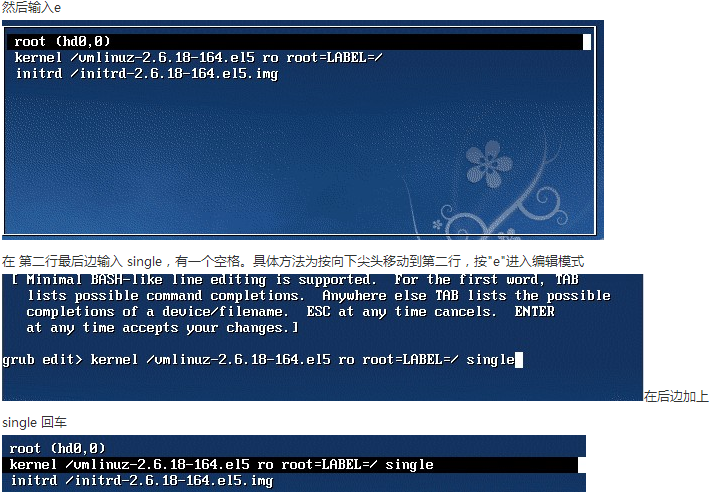
root用户登录，vi编辑 /etc/inittab 文件，修改最后一行的级别数，保存后重启系统生效。

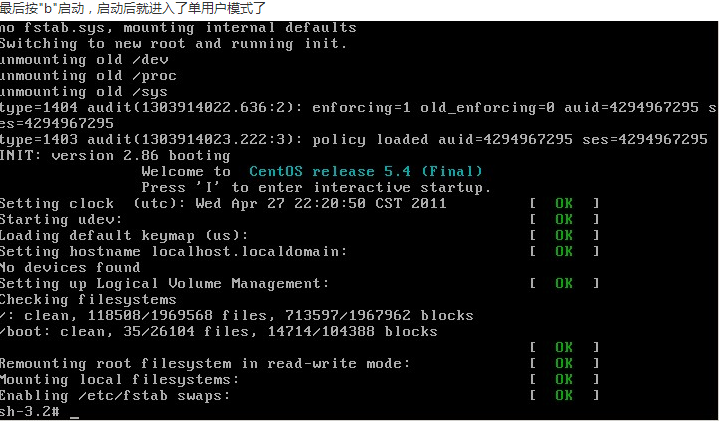


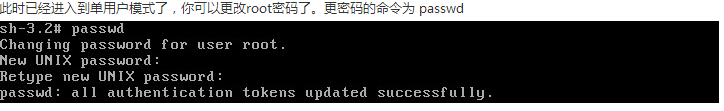
### 7、忘记root密码的解决方法











### 8、永久修改hostname

#### 8.1、vi /etc/sysconfig/network

|  |
| --- |
| NETWORKING=yes  HOSTNAME=mdw |

#### 8.2、vi /etc/hosts

|  |
| --- |
| 127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  192.168.137.129 locahost  192.168.137.129 mdw |

#### 8.3、重启系统（reboot/shutdown –r now）

## 常用命令

shutdown –h now : 立刻进行关机

shutdown –r now : 重启机器；reboot : 重启机器

logout : 退出登录

man 命令： 显示出命令使用格式、参数选项等信息

whereis 命令 : 显示出命令所在位置，同时可以检测linux系统中是否有此命令

echo $SHELL : 查看当前使用的shell

### 1、vi 命令

文本编辑器，新增或修改文件，格式为 ： vi 文件。vi可以分为三种状态：

1. 命令行模式（command mode）：控制屏幕光标的移动，字符、字或行的删除，移动复制某区段及进入Insert mode下，或者到last line mode。
2. 插入模式（Insert model）：只有在Insert mode下，才可以做文字输入，按ESC键可回到command mode。
3. 低行模式（last line mode）：将文件保存或退出vi，也可以设置编辑环境，如寻找字符串、列出行号等。
4. vi命令图 :



### 用户和用户组管理

Linux系统是一个多用户多任务的分时操作系统，任何一个要使用系统资源的用户，都必须首先向系统管理员申请一个账号，然后以这个账号的身份进入系统。

用户的账号一方面可以帮助系统管理员对使用系统的用户进行跟踪，并控制他们对系统资源的访问；另一方面也可以帮助用户组织文件，并为用户提供安全性保护。

每个用户账号都拥有一个惟一的用户名和各自的口令。

用户在登录时键入正确的用户名和口令后，就能够进入系统和自己的主目录

#### 用户账号管理

##### 查看系统中所有用户信息

cat /etc/passwd : 查看系统中所有用户信息

xiaohei:x:503:502:a:/home/xiaohei:/bin/bash

xiaohei : 用户名

x : 用户秘密加密信息

503 : 用户ID

502 : 用户所在组的ID

a : 用户信息的注释，为写时为空

/home/xiaohei : 用户的主目录

/bin/bash : 用户所使用的shell解析器

用户名:口令:用户标识号:组标识号:注释性描述:主目录:登录Shell

##### useradd 命令

添加新的用户账号

语法：useradd [选项] 用户名

选项：

**-c** ：comment 指定一段注释性描述

**-d** ：目录 指定用户主目录，如果此目录不存在，则同时使用-m选项，可以创建主目录

**-g** ：用户组 指定用户所属的用户组

**-G** ：用户组，用户组 指定用户所属的附加组

**-s** ：Shell文件 指定用户的登录Shell

**-u** ：用户号 指定用户的用户号，如果同时有-o选项，则可以重复使用其他用户的标识号

|  |
| --- |
| useradd -s /bin/bash -g customuser -d /home/sam -m sam |

##### usermod 命令

修改用户账号

语法：usermod [选项] 用户名

选项：

**-c** ：comment 指定一段注释性描述

**-d** ：目录 指定用户主目录，如果此目录不存在，则同时使用-m选项，可以创建主目录

**-g** ：用户组 指定用户所属的用户组

**-G** ：用户组，用户组 指定用户所属的附加组

**-s** ：Shell文件 指定用户的登录Shell

**-u** ：用户号 指定用户的用户号，如果同时有-o选项，则可以重复使用其他用户的标识号

**-l** ：新用户名

|  |
| --- |
| usermod -g developer -l samm -d /home/samm -m sam |

##### userdel 命令

删除账号

语法：userdel [选项] 用户名

选项：

**-r** ： 把用户的主目录一起删除

|  |
| --- |
| userdel –r samm |

##### passwd 命令

用户指定和修改用户口令。超级用户可以为自己和其他用户指定口令，普通用户只能用它修改自己的口令。

语法：passwd [选项] 用户名

选项：

**-l**：锁定口令，即禁用账号

**-u**：口令解锁

**-d**：使账号无口令

**-f**：强迫用户下次登录时修改口令

如果默认用户名，则修改当前用户的口令。

#### 用户组管理

##### 查看系统中所有组信息

cat /etc/group : 查看系统中有哪些组

manager:x:501:

manager : 组名

x : 权限加密信息

501 : 组号，唯一

组名:口令:组标识号:组内用户列表

##### groupadd 命令

添加一个新的用户组

语法：groupadd [选项] 用户组名

选项：

-g: GID 指定新用户组的朱标识号

-o: 一般与-g选项同时使用，表示新用户组的GID可以与系统已有用户组的GID相同

|  |
| --- |
| groupadd –g 101 developer |

##### groupmod 命令

修改用户组

语法：groupmod [选项] 用户组名

选项：

-g: GID 指定新用户组的朱标识号

-o: 一般与-g选项同时使用，表示新用户组的GID可以与系统已有用户组的GID相同

-n: 新用户组名

|  |
| --- |
| groupmod –n developer2 developer |

##### groupdel 命令

删除一个存在的用户组。如果有用户、文件隶属于该组，那么将无法删除。

语法：groupdel 用户组

|  |
| --- |
| groupdel developer |

##### newgrp 命令

如果一个用户同时属于多个用户组（主组或者附加组），那么用户在登录后，可以使用命令newgrp切换到其他用户组，以便具有其他用户组的权限。

语法：newgrp 目的用户组

|  |
| --- |
| newgrp root |

#### 批量添加用户

##### （1）先编辑一个文本用户文件

每一列按照/etc/passwd密码文件的格式书写，要注意每个用户的用户名、UID、宿主目录都不可以相同，其中密码栏可以留作空白或输入x号，一个范例文件user.txt内容如下：

user001::600:501:multiple add user test:/home/user001:/bin/bash

user002::601:501:multiple add user test:/home/user002:/bin/bash

user003::602:501:multiple add user test:/home/user003:/bin/bash

user004::603:501:multiple add user test:/home/user004:/bin/bash

user005::604:501:multiple add user test:/home/user005:/bin/bash

user006::605:501:multiple add user test:/home/user006:/bin/bash

##### （2）执行/usr/sbin/newusers 命令

以root身份执行命令/user/sbin/newusers，从刚创建的用户文件user.txt中导入数据，创建用户：

# newusers < user.txt

然后可以执行 cat /etc/passwd 检查/etc/passwd文件是否已经出现这些用户数据，以及执行ls –l /home 检查home目录下用户的宿主目录是否已经创建

##### （3）执行/usr/sbin/pwunconv 命令

将 /etc/shadow 产生的 shadow 密码解码，然后回写到 /etc/passwd 中，并将/etc/shadow的shadow密码栏删掉。这是为了方便下一步的密码转换工作，即先取消 shadow password 功能

# pwunconv

##### （4）编辑每个用户的密码对照文件

范例文件password.txt，内容如下：

user001:密码

user002:密码

user003:密码

user004:密码

user005:密码

user006:密码

##### （5）以root身份执行/usr/sbin/chpasswd 命令

# chpasswd < passwd.txt

##### （6）执行/usr/sbin/pwconv命令

执行命令 /usr/sbin/pwconv 将密码编码为 shadow password，并将结果写入 /etc/shadow。

# pwconv

### 处理目录的常用命令

#### ls/ll/dir 命令

dir/ls : 查看当前目录下面有哪些文件， -l ： 详细列表显示，-a : 显示隐藏文件，-al : 详细列表显示所有文件（显示和隐藏的文件），-F : 可视化显示文件类型（/:目录，@：符号链接文件，\*：可执行文件）

ll : 相当于 dir/ls –l

ls 命令常用参数说明：

-l : 列出文件的权限、所有者、组信息、文件大小、文件名等详细信息

-a : 列出目录下的所有文件，包括以 . 开头的隐含文件

-A :同 –a ，但不列出“.”(表示当前目录)和“..”(表示当前目录的父目录)

-c : 配合 –lt : 根据ctime顺序排序及显示ctime；配合 –l : 根据名称排序

-g : 类似 –l ,但不列出所有者

-o、-G : 类似 –l ,显示文件的除组信息外的详细信息

-h : (human readable)以容易理解的格式列出文件大小（例如 1K 20M 2G 3T）

-L : 当显示符号链接的文件信息时，显示符号链接所指向的对象而并非符号链接本身的信息

-m : 所有项目以逗号分隔，并填满整行行宽

-r : (reverse)以相反的次序排序

-R : (recursive)同时列出所有子目录层

-s : (size)以块大小为单位列出所有文件的大小

-S : 根据文件大小降序排序

-t : 以文件修改时间降序排序

-u : 配合 –lt 显示访问时间并依访问时间顺序排序；配合 –l 显示访问时间但根据名称排序；否则根据访问时间排序

-U : 不进行排序，依文件系统原有的次序列出项目

-x : 逐行列出项目而不是逐栏列出

-X : 根据扩展名排序

ls –lth [目录] | grep “^d”: 按文件修改时间降序列表当前目录或指定目录下的子目录列表详细信息

ls –lth [目录] | grep “^-“ : 按文件修改时间降序列出当前目录或指定目录下的文件列表详细信息

ls –l [目录] | grep “^-“ | wc –l : 计算当前目录或指定目录下的文件个数

ls –l [目录] | grep “^d” | wc –l : 计算当前目录或指定目录下的目录个数

#### cd命令

是Change Directory的缩写，用来变换工作目录的命令

#### pwd命令

是Print Working Directory的缩写，用来显示目前所在目录的命令

#### mkdir 命令

mkdir 命令用来创建指定名称的目录，要求创建目录的用户在当前目录中具有写权限，并且指定的目录名不能是当前目录中已有的目录（mkdir [选项] 目录名称）

mkdir 命令参数说明：

-m : 配置文件的权限，直接配置，不需要看默认权限（umask）的脸色

-p : parents 可以是一个路径名称。此时若路径中某些尚不存在，加上此选项后，系统将自动建立那些尚不存在的目录，即一次性可以建立多个目录

-v : verbose 每次创建新目录都显示信息

建立test目录：mkdir test

一次性创建一个项目的目录结构：

mkdir -vp scf/{lib/,bin/,doc/{info,product},

logs/{info,product},service/deploy/{info,product}}

给新创建的目录赋予711的权限：

mkdir –m 711 test2

#### rmdir 命令

rmdir 命令功能是删除空目录，一个目录被删除之前必须是空的，也必须具有对父目录的写权限（rmdir [选项] 目录）

rmdir 命令选项参数说明：

-p : 递归删除目录，当子目录删除后其父目录为空时，也一同被删除。

-v : 显示指令执行过程。

rmdir test : 删除test目录

rmdir test/test-1 : 删除test目录下的test-1目录

#### rm 命令

rm 命令功能是删除一个目录中的一个或多个文件或目录，它也可以将某个目录及其下的所有文件及子目录均删除。对于链接文件，只删除链接，原文件均保持不变。

rm 命令是一个危险的命令，使用的时候要特别当心，尤其对于新手，否则整个系统都毁在这个命令下（比如在 / （根目录）下执行 rm \* -rf，此命令的含义是删除根目录下包含子目录的全部目录及文件，并且不给提示），因此在执行rm命令时最好先确认一下在哪个目录，到底要删除什么文件或目录，操作时保持高度清醒。

rm [选项] 文件或目录

rm命令选项参数说明：

-f : force，忽略不存在的文件，从不给出提示。

-i : interactive，进行交互式删除，删除之前会询问确认

-r,-R : recursive，指示rm将参数中列出的全部目录和子目录均递归删除

-v : verbose ， 详细显示进行的步骤

示例1：自定义回收站功能（仅在删除文件时可用），myrm是回收站名称：

myrm(){D=/tmp/$(date+%Y%m%d%H%M%S); mkdir -p $D; mv "$@" $D && echo "moved to $D ok";}

alias rm=’myrm

示例2：删除所有后缀名相同的文件

**find . -name '\*.class' -type f -print -exec rm -rf {} \;**

(1) "."    表示从当前目录开始递归查找  
(2) “ -name '\*.exe' "根据名称来查找，要查找所有以.exe结尾的文件夹或者文件  
(3) " -type f "查找的类型为文件  
(4) "-print" 输出查找的文件目录名  
(5) 最主要的是是-exec了，-exec选项后边跟着一个所要执行的命令，表示将find出来的文件或目录执行该命令。  
     exec选项后面跟随着所要执行的命令或脚本，然后是一对儿{}，一个空格和一个\，最后是一个分号

#### mv 命令

是move的缩写，可以用来移动文件或者将文件改名，经常用来备份文件或者目录。

语法：mv [选项] 源文件或目录 目标文件或目录

选项：

-b : 若需覆盖文件，则覆盖前先行备份

-f：force强制的意思，如果目标文件已经存在，不会询问而直接覆盖

-i：若目标文件（destination）已经存在时，就会询问是否覆盖

-u：若目标文件已经存在，且source比较新，才会更新（update）

mv ttt.txt t2.txt：把ttt.txt改名问t2.txt

mv t2.txt dtext：把t2.txt文件移动到dtext目录中

#### cp 命令

是copy的缩写，用来拷贝文件、目录的命令

语法：

cp [-adfilprsu] 来源档（source） 目标档（destination）

cp [options] source1 source2 source3 …. Directory

选项与参数：

**-a：**相当於 -pdr 的意思，至於 pdr 请参考下列说明；(常用)

**-d：**若来源档为连结档的属性(link file)，则复制连结档属性而非文件本身

**-f：**为强制(force)的意思，若目标文件已经存在且无法开启，则移除后再尝试一次

**-i：**若目标档(destination)已经存在时，在覆盖时会先询问动作的进行(常用)

**-l：**进行硬式连结(hard link)的连结档创建，而非复制文件本身

**-p：**连同文件的属性一起复制过去，而非使用默认属性(备份常用)

**-r：**递归持续复制，用於目录的复制行为；(常用)

**-s：**复制成为符号连结档 (symbolic link)，亦即『捷径』文件

**-u：**若 destination 比 source 旧才升级 destination

用 root 身份，将 root 目录下的 .bashrc 复制到 /tmp 下，并命名为 bashrc

|  |
| --- |
| [root@www ~]# cp ~/.bashrc /tmp/bashrc  [root@www ~]# cp -i ~/.bashrc /tmp/bashrc  cp: overwrite `/tmp/bashrc'? n <==n不覆盖，y为覆盖 |

### 文件权限命令

#### chgrp 命令

用于更改文件所在组，语法如下：

|  |
| --- |
| **chgrp [-R] 文件名** |

参数选项：

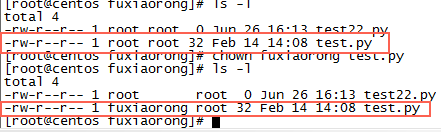
**-R ：**递归更改文件所在组，就是在更改某个目录所在组时，如果加上-R的参数，那么该目录下的所有文件所在组都会更改。

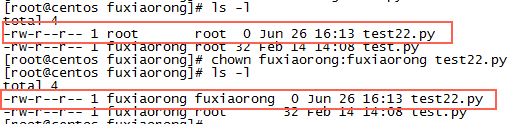
#### chown 命令

用于更改文件所有者，也可以同时更改文件所在组，语法如下：

|  |
| --- |
| **chown [-R] 所有者 文件名**  **chown [-R] 所有者:所在组 文件名** |

例子：





#### chmod命令

chmod命令用于改变文件或目录的访问权限。用它控制文件或目录的访问权限。该命令有两种用法：一种是包含字母和操作符表达式的文字设定法；另一种是包含数字的数字设定法。

Linux系统中的每个文件和目录都有访问许可权限，用它来确定谁可以通过何种方式对文件和目录进行访问和操作。

文件或目录的访问权限分为只读（r,4）、只写（w,2）和可执行（x,1）三种。文件被创建时，文件所有者自动拥有对该文件的读、写和可执行权限，以便于对文件的阅读和修改。用户也可以根据需要把访问权限设置为需要的任何组合。

有三种不用类型的用户可以对文件或目录进行访问：文件所有者、同组用户、其他用户。

文件所有者一般是文件的创建者。所有者可以允许同组用户有访问文件的权限，还可以将文件的访问权限赋予系统中的其他用户，在这种情况下，系统中每一位用户都能访问该用户拥有的文件或目录。

命令格式：chmod [选项] [--help] [--version] 文件或目录

选项参数：

-c : 当发生改变时，报告处理信息

-f : 错误信息不输出

-R : 处理指定目录以及子目录下的所有文件

-v : 运行时显示详情处理信息

文字设定法：chmod [who] [+|-|=] [mode] 文件或目录

<权限范围>+<权限设置> : 使权限范围内的目录或文件具有指定的权限

<权限范围>-<权限设置> : 删除权限范围的目录或文件的指定权限

<权限范围>=<权限设置> : 设置权限范围内的目录或文件的权限为指定的权限

权限范围：

u : 目录或文件的拥有者

g : 目录或文件的当前的群组

o : 除了目录或文件的当前用户或群组之外的用户或者群组

a : 所有的用户及群组

权限代号：

[r] :　读权限，用数字４表示

[ｗ] : 写权限，用数字2表示

[x] : 执行权限，用数字1表示

[-] : 删除权限，用数字0表示

数字设定法：chmod [mode] 文件或目录

我们必须首先了解用数字表示的属性的含义：0表示没有权限，1表示可执行权限，2表示可写权限，4表示可读权限，然后将其相加。

例如，如果想让某个文件的属主有“读/写”二种权限，需要把4（可读）+2（可写）＝6（读/写）。

数字与权限对应如下：（r=4，w=2，x=1）

若要rwx权限则是 ： 4+2+1=7；

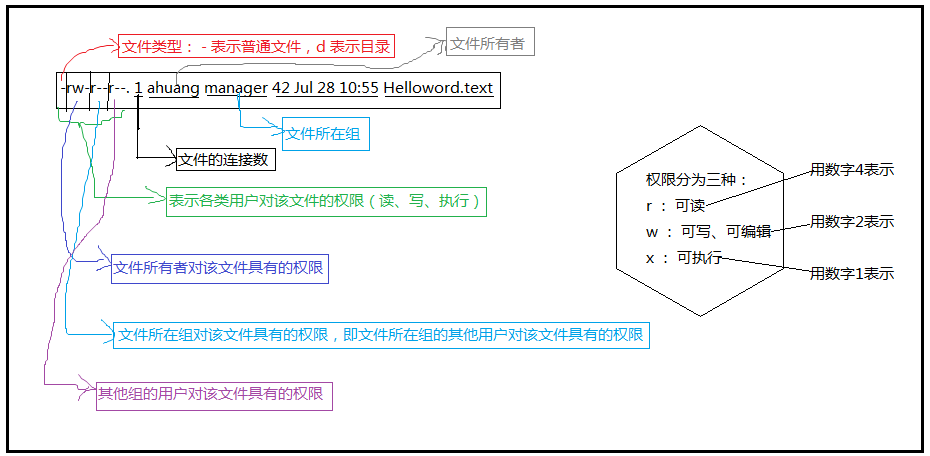
若要rw-权限则是 ： 4+2=6；

若要r-x权限则是 ： 4+1=5；

**符号类型改变文件权限**

还有一个改变权限的方法呦！从之前的介绍中我们可以发现，基本上就九个权限分别是(1)user (2)group (3)others三种身份啦！ 那么我们就可以藉由u, g, o来代表三种身份的权限！此外， a 则代表 all 亦即全部的身份！那么读写的权限就可以写成r, w, x！也就是可以使用底下的方式来看：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| chmod | u g o a | +(加入) -(除去) =(设定) | r w x | 文件或目录 |
|  |  |  |  |  |



### 文件内容查看命令

#### cat 命令

由第一行开始显示文件内容

语法：cat [-AbEnTv] 文件

选项与参数：

-A：相当於 -vET 的整合选项，可列出一些特殊字符而不是空白而已

-b：列出行号，仅针对非空白行做行号显示，空白行不标行号

-E：将结尾的断行字节 $ 显示出来

-n：列印出行号，连同空白行也会有行号，与 -b 的选项不同

-T：将 [tab] 按键以 ^I 显示出来

-v：列出一些看不出来的特殊字符

|  |
| --- |
| [root@www ~]# cat /etc/issue  CentOS release 6.4 (Final)  Kernel \r on an \m |

#### tac 命令

与cat 命令相反，文件内容从最后一行开始显示，可以看出tac是cat的倒着写。如：

|  |
| --- |
| [root@www ~]# tac /etc/issue  Kernel \r on an \m  CentOS release 6.4 (Final) |

#### nl 命令

显示行号

语法：nl [-bnw] 文件

选项与参数：

* -b ：指定行号指定的方式，主要有两种：  
  -b a ：表示不论是否为空行，也同样列出行号(类似 cat -n)；  
  -b t ：如果有空行，空的那一行不要列出行号(默认值)；
* -n ：列出行号表示的方法，主要有三种：  
  -n ln ：行号在萤幕的最左方显示；  
  -n rn ：行号在自己栏位的最右方显示，且不加 0 ；  
  -n rz ：行号在自己栏位的最右方显示，且加 0 ；
* -w ：行号栏位的占用的位数。

|  |
| --- |
| [root@www ~]# nl /etc/issue  1 CentOS release 6.4 (Final)  2 Kernel \r on an \m |

#### more 命令

功能类似cat命令，cat命令时整个文件的内容从上到下显示在屏幕上。More命令会以一页一页的显示，方面使用者逐页阅读，而最基本的指令就是按空格键(space)或F键就往下一页显示，按B键就会往上一页显示，而且还有搜寻字符串的功能。More命令从前向后读取文件，因此在启动时就加载整个文件。

命令格式：

more [-cdlpsu] [-num] [+/pattern] [+linenum] [file]

命令参数：

+linenum : 从第linenum行开始显示

-num : 定义屏幕大小为num行

+/pattern : 在每个文件显示前搜寻该字符串parttern，然后从该字符串前两行之后开始显示

-c : 从顶部清屏，然后显示

-d : 提示“Press space to continue,’q’to quit”,禁止响铃功能

-l : 忽略Ctrl+l（换页）字符

-p : 通过清除窗口而不是滚屏来对文件进行换页，与-c选项相似

-s : 把连续的多个空行显示为一行

-u : 把文件内容中的下划线去掉

在more这个程序的运行过程中，可以有以下些操作命令可以按：

Enter键 : 向下n行，需要定义。默认为1行

F键、space键：下一页

/字串：代表在这个显示的内容当中，向下搜寻[字串]这个关键字

B键：上一页

= 键：输出当前行的行号

[:f] : 输出文件名和当前行的行号

V 键：调用vi编辑器

！命令：调用Shell并执行命令

q键：退出more

|  |
| --- |
| [root@www ~]# more /etc/man.config  #  # Generated automatically from man.conf.in by the  # configure script.  #  # man.conf from man-1.6d  ....(中间省略)....  --More--(28%) <== 重点在这一行喔！你的光标也会在这里等待你的命令 |

#### less 命令

一页一页翻动，与more命令类似，在less运行时可以输入的命令有：

* 空白键 ：向下翻动一页
* [pagedown]：向下翻动一页
* [pageup]  ：向上翻动一页
* /字串 ：向下搜寻『字串』的功能
* ?字串 ：向上搜寻『字串』的功能
* n ：重复前一个搜寻 (与 / 或 ? 有关！)
* N ：反向的重复前一个搜寻 (与 / 或 ? 有关！)
* q ：离开 less 这个程序

|  |
| --- |
| [root@www ~]# less /etc/man.config  #  # Generated automatically from man.conf.in by the  # configure script.  #  # man.conf from man-1.6d  ....(中间省略)....  : <== 这里可以等待你输入命令！ |

#### head 命令

取出文件前面几行

语法：head [-n number] 文件

选项与参数：

-n ：后面接数字，代表显示几行的意思，默认数字为10

|  |
| --- |
| [root@centos ~]# head /etc/man.config  [root@centos ~]# head –n 20 /etc/man.config |

#### tail 命令

取出文件后面几行

语法：head [-n number] 文件

选项与参数：

-n ：后面接数字，代表显示几行的意思，默认数字为10

-f：表示持续侦测后面所接的档名，要等到按下[ctrl]-c才会结束tail的侦测

|  |
| --- |
| [root@centos ~]# tail /etc/man.config  [root@centos ~]# tail –n 20 /etc/man.config |

#### wc 命令

word count的简写，用来统计文件的行数、字节数、字数，并将统计结果显示输出。

语法：wc [选项] 文件

-l ：统计行数

-w ：统计字数

-c ：统计字节数

-m ：统计字符数，不能和-c同时使用

-L ：打印最长行的长度

|  |
| --- |
| [root@centos ~]# wc /home/tecmint.txt  12 16 112 tecmint.txt |
| 12：行数  16：字数  112：字符数  tecmint.txt：文件名 |

### 输入/输出重定向(追加或覆盖文件)

|  |  |
| --- | --- |
| 命令 | 说明 |
| command > file | 将输出重定向到file（会覆盖file文件的内容） |
| command < file | 将输入重定向到file（会覆盖file文件的内容） |
| command >> file | 将输出以追加的方式重定向到file |
| n > file | 将文件描述符为n的文件重定向到file |
| n >> file | 将文件描述符为n的文件以追加的方式重定向到file |
| n >& m | 将输出文件m和n合并 |
| n <& m | 将输入文件m和n合并 |
| << tag | 将开始标记tag和结束标记tag之间的内容作为输入 |

文件描述符n有以下几种：

0：标准输入（STDIN）

1：标准输出（STDOUT）

2：标准错误（STDERR）

### 10、touch 命令

touch [选项] 文件：新建一个不存在的文件或者修改文件时间戳

-a : 更改存取时间

-c : 不建立任何文档

-d : 使用指定的日期时间，而非现在的时间

-m : 更改变动时间

### 11、tar 命令

通过SSH访问服务器，难免会要用到压缩、解压缩、打包、解包等操作，tar命令就是必不可少的一个功能强大的工具。tar是麻雀虽小，五脏俱全，功能强大。

tar命令可以问linux的文件和目录创建档案，利用tar，可以为某一特定文件创建档案（备份文件），也可以在档案中改变文件，或者向档案中加入新的文件。Tar最初被用来在磁带上创建档案，现在，用户可以在任何设备上创建档案。利用tar，可以把一大堆的文件和目录全部打包成一个文件，便于网络传输。

首先要弄清两个概念：打包和压缩。打包是指将一大堆文件或目录变成一个总文件；压缩是将一个大的文件通过一些压缩算法变成一个小文件。

区分这两个概念，主要是源于linux中很多压缩程序只能针对一个文件进行压缩，这样当你想要压缩一大堆文件时，得先将这一大堆文件先打成一个包（tar命令），然后再用压缩程序进行压缩（gzip bzip2命令）

Linux下最常用的打包程序就是tar了，使用tar程序打出来的包称之为tar包，tar包文件的命令通常都是以.tar结尾，生成tar包后，就可以用其他的程序来进行压缩。

命令格式：tar [必要参数] [可选参数] 文件

命令功能：用来压缩和解压文件。tar 本身不具有压缩功能，是调用压缩功能实现的

必要参数说明：

-A : 新增压缩文件到已存在的压缩

-B : 设置区块大小

-c : 建立新的压缩文件

-d : 记录文件的差别

-r : 添加文件到已经压缩的文件

-u : 添加改变了和现有的文件到已经存在的压缩文件

-x : 从压缩的文件中提取文件

-t : 显示压缩文件的内容

-z : 支持gzip解压文件

-j : 支持bzip2解压文件

-Z : 支持compress解压文件

-v : 显示操作过程

-l : 文件系统边界设置

-k : 保留原有文件不覆盖

-m :保留文件不被覆盖

-W : 确保压缩文件的正确性

可选参数说明：

-b : 设置区块数目

-C : 切换到指定目录

-f : 指定压缩文件

### 12、chkconfig 命令

chkconfig命令主要用来更新（启动或停止）和查询系统服务的运行级信息。谨记chkconfig不是立即自动禁止或激活一个服务，它只是简单的改变了符号连接。

使用语法：  
chkconfig [--add][--del][--list][系统服务] 或 chkconfig [--level <等级代号>][系统服务][on/off/reset]

chkconfig在没有参数运行时，显示用法。如果加上服务名，那么就检查这个服务是否在当前运行级启动。如果是，返回true，否则返回false。如果在服务名后面指定了on，off或者reset，那么chkconfi 会改变指定服务的启动信息。on和off分别指服务被启动和停止，reset指重置服务的启动信息，无论有问题的初始化脚本指定了什么。on和off开关，系统默认只对运行级3，4，5有效，但是reset可以对所有运行级有效。

参数用法：  
   --add 　增加所指定的系统服务，让chkconfig指令得以管理它，并同时在系统启动的叙述文件内增加相关数据。  
   --del 　删除所指定的系统服务，不再由chkconfig指令管理，并同时在系统启动的叙述文件内删除相关数据。  
   --level<等级代号> 　指定读系统服务要在哪一个执行等级中开启或关毕。  
      等级0表示：表示关机  
      等级1表示：单用户模式  
      等级2表示：无网络连接的多用户命令行模式  
      等级3表示：有网络连接的多用户命令行模式  
      等级4表示：不可用  
      等级5表示：带图形界面的多用户模式  
      等级6表示：重新启动  
      需要说明的是，level选项可以指定要查看的运行级而不一定是当前运行级。对于每个运行级，只能有一个启动脚本或者停止脚本。当切换运行级时，init不会重新启动已经启动的服务，也不会再次去停止已经停止的服务。

    chkconfig --list [name]：显示所有运行级系统服务的运行状态信息（on或off）。如果指定了name，那么只显示指定的服务在不同运行级的状态。  
    chkconfig --add name：增加一项新的服务。chkconfig确保每个运行级有一项启动(S)或者杀死(K)入口。如有缺少，则会从缺省的init脚本自动建立。  
    chkconfig --del name：删除服务，并把相关符号连接从/etc/rc[0-6].d删除。  
    chkconfig [--level levels] name：设置某一服务在指定的运行级是被启动，停止还是重置。

运行级文件：  
每个被chkconfig管理的服务需要在对应的init.d下的脚本加上两行或者更多行的注释。第一行告诉chkconfig缺省启动的运行级以及启动和停止的优先级。如果某服务缺省不在任何运行级启动，那么使用 - 代替运行级。第二行对服务进行描述，可以用\ 跨行注释。  
例如，random.init包含三行：  
# chkconfig: 2345 20 80  
# description: Saves and restores system entropy pool for \  
# higher quality random number generation.

使用范例：  
chkconfig --list #列出所有的系统服务  
chkconfig --add httpd #增加httpd服务  
chkconfig --del httpd #删除httpd服务  
chkconfig --level httpd 2345 on #设置httpd在运行级别为2、3、4、5的情况下都是on（开启）的状态  
chkconfig --list  #列出系统所有的服务启动情况  
chkconfig --list mysqld #列出mysqld服务设置情况  
chkconfig --level 35 mysqld on #设定mysqld在等级3和5为开机运行服务，--level 35表示操作只在等级3和5执行，on表示启动，off表示关闭  
chkconfig mysqld on #设定mysqld在各等级为on，“各等级”包括2、3、4、5等级

如何增加一个服务：  
1.服务脚本必须存放在/etc/ini.d/目录下；  
2.chkconfig --add servicename  
    在chkconfig工具服务列表中增加此服务，此时服务会被在/etc/rc.d/rcN.d中赋予K/S入口了；  
3.chkconfig --level 35 mysqld on  
    修改服务的默认启动等级。

### 13、su命令

作用：变更为其他普通用户

语法：su username

备注：如果要从普通用户变更为root用户，则使用 root – 命令，提示输入root用户密码即可。

### 14、find命令

find命令用于：在一个目录（及子目录）中搜索文件，可以指定一些匹配条件，如按文件名、文件类型、用户甚至时间戳查找文件。

命令格式：

find [-H] [-L] [-P] [-D debugopts] [-Olevel] [path...] [expression]

其中[-H] [-L] [-P] [-D debugopts] [-Olevel] 这几个选项并不常用，所以命令可以简化为：

find [path...] [expression]

* path：find命令所查找的目录路径。例如用.来表示当前目录，用/来表示系统根目录
* expression：expression可以分为——“-options [-print -exec -ok ...]”
* -options，指定find命令的常用选项，下节详细介绍
* -print，find命令将匹配的文件输出到标准输出
* -exec，find命令对匹配的文件执行该参数所给出的shell命令。相应命令的形式为'command' {  } \;，注意{   }和\；之间的空格   
  find ./ -size 0 -exec rm {} \; 删除文件大小为零的文件 （还可以以这样做：rm -i `find ./ -size 0`  或 find ./ -size 0 | xargs rm -f &）   
  为了用ls -l命令列出所匹配到的文件，可以把ls -l命令放在find命令的-exec选项中：find . -type f -exec ls -l {  } \;   
  在/logs目录中查找更改时间在5日以前的文件并删除它们：find /logs -type f -mtime +5 -exec rm {  } \;
* -ok，和-exec的作用相同，只不过以一种更为安全的模式来执行该参数所给出的shell命令，在执行每一个命令之前，都会给出提示，让用户来确定是否执行。   
  find . -name "\*.conf"  -mtime +5 -ok rm {  } \; 在当前目录中查找所有文件名以.LOG结尾、更改时间在5日以上的文件，并删除它们，只不过在删除之前先给出提示

实例：

* -name   
  按照文件名查找文件。   
  find /dir -name filename  在/dir目录及其子目录下面查找名字为filename的文件   
  find . -name "\*.c" 在当前目录及其子目录（用“.”表示）中查找任何扩展名为“c”的文件
* -perm   
  按照文件权限来查找文件。   
  find . -perm 755 –print 在当前目录下查找文件权限位为755的文件，即文件属主可以读、写、执行，其他用户可以读、执行的文件
* -prune   
  使用这一选项可以使find命令不在当前指定的目录中查找，如果同时使用-depth选项，那么-prune将被find命令忽略。   
  find /apps -path "/apps/bin" -prune -o –print 在/apps目录下查找文件，但不希望在/apps/bin目录下查找   
  find /usr/sam -path "/usr/sam/dir1" -prune -o –print 在/usr/sam目录下查找不在dir1子目录之内的所有文件
* -user   
  按照文件属主来查找文件。   
  find ~ -user sam –print 在$HOME目录中查找文件属主为sam的文件
* -group   
  按照文件所属的组来查找文件。   
  find /apps -group gem –print 在/apps目录下查找属于gem用户组的文件
* -mtime -n +n   
  按照文件的更改时间来查找文件， - n表示文件更改时间距现在n天以内，+ n表示文件更改时间距现在n天以前。   
  find / -mtime -5 –print 在系统根目录下查找更改时间在5日以内的文件   
  find /var/adm -mtime +3 –print 在/var/adm目录下查找更改时间在3日以前的文件
* -nogroup   
  查找无有效所属组的文件，即该文件所属的组在/etc/groups中不存在。   
  find / –nogroup -print
* -nouser   
  查找无有效属主的文件，即该文件的属主在/etc/passwd中不存在。   
  find /home -nouser –print
* -newer file1 ! file2   
  查找更改时间比文件file1新但比文件file2旧的文件。
* -type   
  查找某一类型的文件，诸如：   
  b - 块设备文件。   
  d - 目录。   
  c - 字符设备文件。   
  p - 管道文件。   
  l - 符号链接文件。   
  f - 普通文件。   
  find /etc -type d –print 在/etc目录下查找所有的目录   
  find . ! -type d –print 在当前目录下查找除目录以外的所有类型的文件   
  find /etc -type l –print 在/etc目录下查找所有的符号链接文件
* -size n：[c] 查找文件长度为n块的文件，带有c时表示文件长度以字节计。   
  find . -size +1000000c –print 在当前目录下查找文件长度大于1 M字节的文件   
  find /home/apache -size 100c –print 在/home/apache目录下查找文件长度恰好为100字节的文件   
  find . -size +10 –print 在当前目录下查找长度超过10块的文件（一块等于512字节）
* -depth：在查找文件时，首先查找当前目录中的文件，然后再在其子目录中查找。   
  find / -name "CON.FILE" -depth –print 它将首先匹配所有的文件然后再进入子目录中查找
* -mount：在查找文件时不跨越文件系统mount点。    
  find . -name "\*.XC" -mount –print 从当前目录开始查找位于本文件系统中文件名以XC结尾的文件（不进入其他文件系统）
* -follow：如果find命令遇到符号链接文件，就跟踪至链接所指向的文件。

find与xargs：

在使用find命令的-exec选项处理匹配到的文件时， find命令将所有匹配到的文件一起传递给exec执行。但有些系统对能够传递给exec的命令长度有限制，这样在find命令运行几分钟之后，就会出现溢出错误。错误信息通常是“参数列太长”或“参数列溢出”。这就是xargs命令的用处所在，特别是与find命令一起使用。

find命令把匹配到的文件传递给xargs命令，而xargs命令每次只获取一部分文件而不是全部，不像-exec选项那样。这样它可以先处理最先获取的一部分文件，然后是下一批，并如此继续下去。

在有些系统中，使用-exec选项会为处理每一个匹配到的文件而发起一个相应的进程，并非将匹配到的文件全部作为参数一次执行；这样在有些情况下就会出现进程过多，系统性能下降的问题，因而效率不高；

而使用xargs命令则只有一个进程。另外，在使用xargs命令时，究竟是一次获取所有的参数，还是分批取得参数，以及每一次获取参数的数目都会根据该命令的选项及系统内核中相应的可调参数来确定。

来看看xargs命令是如何同find命令一起使用的，并给出一些例子。

find . -type f -print | xargs file 查找系统中的每一个普通文件，然后使用xargs命令来[测试](http://lib.csdn.net/base/softwaretest)它们分别属于哪类文件

find / -name "core" -print | xargs echo "" >/tmp/core.log 在整个系统中查找内存信息转储文件(core dump) ，然后把结果保存到/tmp/core.log 文件中：

find . -type f -print | xargs grep "hostname" 用grep命令在所有的普通文件中搜索hostname这个词

find ./ -mtime +3 -print|xargs rm -f –r 删除3天以前的所有东西 （find . -ctime +3 -exec rm -rf {} \;）

find ./ -size 0 | xargs rm -f & 删除文件大小为零的文件

find命令配合使用exec和xargs可以使用户对所匹配到的文件执行几乎所有的命令。

### 15、grep命令

grep (global search regular expression\_r(RE) and print out the line,全面搜索正则表达式并把行打印出来)是一种强大的文本搜索工具，它能使用正则表达式搜索文本，并把匹配的行打印出来。

命令格式：

grep [OPTIONS] PATTERN [FILE...]   
grep [OPTIONS] [-e PATTERN | -f FILE] [FILE...]

grep命令用于搜索由Pattern参数指定的模式，并将每个匹配的行写入标准输出中。这些模式是具有限定的正则表达式，它们使用ed或egrep命令样式。如果在File参数中指定了多个名称，grep命令将显示包含匹配行的文件的名称。对 shell 有特殊含义的字符 ($, \*, [, |, ^, (, ), \ ) 出现在 Pattern参数中时必须带双引号。如果 Pattern参数不是简单字符串，通常必须用单引号将整个模式括起来。在诸如 [a-z], 之类的表达式中，-（减号）cml 可根据当前正在整理的序列来指定一个范围。整理序列可以定义等价的类以供在字符范围中使用。如果未指定任何文件，grep会假定为标准输入。

grep正则表达式元字符集(基本集)：

^  锚定行的开始 如：'^grep'匹配所有以grep开头的行。

$  锚定行的结束 如：'grep$'匹配所有以grep结尾的行。

.   匹配一个非换行符的字符 如：'gr.p'匹配gr后接一个任意字符，然后是p。

\*  匹配零个或多个先前字符 如：'\*grep'匹配所有一个或多个空格后紧跟grep的行。 .\*一起用代表任意字符。

[] 匹配一个指定范围内的字符，如'[Gg]rep'匹配Grep和grep。

[^]  匹配一个不在指定范围内的字符，如：'[^A-FH-Z]rep'匹配不包含A-R和T-Z的一个字母开头，紧跟rep的行。

..  标记匹配字符，如：'love'，love被标记为1。

\<  锚定单词的开始，如：'\<grep'匹配包含以grep开头的单词的行。

\>  锚定单词的结束，如'grep\>'匹配包含以grep结尾的单词的行。

x\{m\} 连续重复字符x，m次，如：'o\{5\}'匹配包含连续5个o的行。

x\{m,\} 连续重复字符x,至少m次，如：'o\{5,\}'匹配至少连续有5个o的行。

x\{m,n\} 连续重复字符x，至少m次，不多于n次，如：'o\{5,10\}'匹配连续5--10个o的行。

\w  匹配一个文字和数字字符，也就是[A-Za-z0-9]，如：'G\w\*p'匹配以G后跟零个或多个文字或数字字符，然后是p。

\W  w的反置形式，匹配一个非单词字符，如点号句号等。\W\*则可匹配多个。

\b  单词锁定符，如: '\bgrep\b'只匹配grep，即只能是grep这个单词，两边均为空格。

常用选项及实例：

-?

同时显示匹配行上下的？行，如：grep -2 pattern filename同时显示匹配行的上下2行。

-b，--byte-offset

打印匹配行前面打印该行所在的块号码。

-c,--count

只打印匹配的行数，不显示匹配的内容。

-f File，--file=File

从文件中提取模板。空文件中包含0个模板，所以什么都不匹配。

-h，--no-filename

当搜索多个文件时，不显示匹配文件名前缀。

-i，--ignore-case

忽略大小写差别。

-q，--quiet

取消显示，只返回退出状态。0则表示找到了匹配的行。

-l，--files-with-matches

打印匹配模板的文件清单。

-L，--files-without-match

打印不匹配模板的文件清单。

-n，--line-number

在匹配的行前面打印行号。

-s，--silent

不显示关于不存在或者无法读取文件的错误信息。

-v，--revert-match

反检索，只显示不匹配的行。

-w，--word-regexp

如果被\<和\>引用，就把表达式做为一个单词搜索。

-V，--version

显示软件版本信息。

=====

ls -l | grep '^a' 通过管道过滤ls -l输出的内容，只显示以a开头的行。

ls -l | grep 'test' d\* 显示所有以d开头的文件中包含test的行。

ls -l | grep 'test' aa bb cc 显示在aa，bb，cc文件中匹配test的行。

ls -l | grep '[a-z]' aa 显示所有包含每个字符串至少有5个连续小写字符的字符串的行。

ls -l | grep 'w(es)t.\*' aa 如果west被匹配，则es就被存储到内存中，并标记为1，然后搜索任意个字符(.\*)，这些字符后面紧跟着另外一个es()，找到就显示该行。如果用egrep或grep -E，就不用""号进行转义，直接写成'w(es)t.\*'就可以了。

### 16、crontab

crontab命令的功能是在一定时间间隔调度一些命令的执行

|  |
| --- |
|  |

**1.Cron的启动与关闭**

由于Cron是Linux的内置服务，可以用以下的方法启动.关闭这个服务:

service crond start           //启动服务  
service crond stop            //关闭服务  
service crond restart        //重启服务  
service crond reload         //重新载入配置

**2.Cron配置文件**

**2.1全局配置文件**

crontab在/etc目录下面存在

cron.hourly,

cron.daily,

cron.weekly,

cron.monthly,

cron.d

五个目录和crontab,cron.deny二个文件.

job,cron.hourly是每个小时执行一次的

cron.daily是每天执行一次的

job,cron.weekly是每个星期执行一次的

job.cron.monthly是每月执行一次的

job.cron.d是系统自动定期需要做的任务,但是又不是按小时,按天,按星期,按月来执行的,那么就放在这个目录下面.

/etc/crontab文件一般如下:

01 \* \* \* \* root run-parts /etc/cron.hourly  
02 4 \* \* \* root run-parts /etc/cron.daily  
22 4 \* \* 0 root run-parts /etc/cron.weekly  
42 4 1 \* \* root run-parts /etc/cron.monthly

我们可在此文件中添加自己需要的cron job.

/etc/cron.deny文件就是用于控制不让哪些用户使用Crontab的功能.

**2.2用户配置文件**

每个用户都有自己的cron配置文件,通过crontab -e 就可以编辑,一般情况下我们编辑好用户的cron配置文件保存退出后,系统会自动就存放于/var/spool/cron/目录中,文件以用户名命名.

linux的cron服务是每隔一分钟去读取一次/var/spool/cron,/etc/crontab,/etc/cron.d下面所有的内容.

**3.Cron命令格式**

crontab [ －u user ] 文件   
crontab [ －u user ] { －l | －r | －e }

-u:指定某一用户

－e:执行文字编辑器来设定用户(当前用户或指定用户)时程表,内定的文字编辑器是vi.  
－r:删除用户时程表.  
－l:列出用户时程表.

**4.Cron文件格式**

\*  \*  \*  \*  \*  command  
分 时 日 月 周　  命令

第1列表示分钟1～59, 每分钟用\*或者 \*/1表示  
第2列表示小时1～23(0表示0点)  
第3列表示日期1～31  
第4列表示月份1～12  
第5列标识号星期0～6(0表示星期天)  
第6列要运行的命令

**5.Cron文件使用说明**

**5.1 一般情况**

当f1 为 \* 时表示每分钟都要执行 program,f2 为 \* 时表示每小时都要执行程序,其余类推.

\* \* \* \* \* /bin/usershell  每天每分钟执行一次/bin/usershell  
当f1 为 a-b 时表示从第 a 分钟到第 b 分钟这段时间内要执行,f2 为 a-b 时表示从第 a 到第 b 小时都要执行,其余类推.

0-12 \* \* \* \* /bin/usershell  每天每小时从0到12分钟每分钟执行一次/bin/usershell  
当f1 为 \*/n 时表示每 n 分钟个时间间隔执行一次,f2 为 \*/n 表示每 n 小时个时间间隔执行一次,其余类推.

\* \*/2 \* \* \* /bin/usershell  每天每2小时执行一次/bin/usershell  
当f1 为 a, b, c,... 时表示第 a, b, c,... 分钟要执行,f2 为 a, b, c,... 时表示第 a, b, c...个小时要执行,其余类推.

\* 1,3,5,7 \* \* \* /bin/usershell  每天每逢1,3,4,7点执行一次/bin/usershell

**5.2. 冲突逻辑**

日期可以用月限定,也可以用“星期”指定,如果两个段有冲突,那么,第六段的命令将在匹配任何一个的情况下都运行,比如

"30 4 1,15 \* 5",将在每月的1号和15号加每个周五,上午4：30运行.

**5.3. 符号"%"**

"%"在Cron文件中,有"结束命令行","换行","重定向"的作用,假如不需要"%"的特殊作用,需要使用转义符转义.

**5.4. @reboot**

这个不需要理解,为了达到在开机后运行,且只运行一次的目的.除了这个,也无法通过前五段的设置来实现.

@annually也是这个功能.

至于@yearly,@monthly等等其实都可以用上面的五段来设置.

**小例子**做了下测试：

使用**crontab -e**进行编辑，内容为：

\*/2 \* \* \* \* echo "i am crontab" >> /home/zhang/hello

这样隔两分钟就会往hello这个文件里写i am crontab

写一个脚本task.sh，直接运行task.sh时可以运行，可是加到crontab里就有运行了，先来看一下task.sh脚本：

#!/bin/sh

java -jar offergateway.biz.offer-1.0-SNAPSHOT.jar

crontab定时任务的写法为：

\* \* \* \* \*  /task.sh

以上定时任务和task.sh都有问题，因为在运行crontab，必须使用绝对路径。

在crontab的执行环境变量中，没有相应用户设置的环境变量，你需要手动设置一下环境变量，才能让它生效。把上面的脚本改成：

#!/bin/sh

. /etc/profile

java -jar /home/zhang/offergateway.biz.offer-1.0-SNAPSHOT.jar

crontab也改成绝对路径：

\* \* \* \* \*  sh /home/zhang/task.sh

运行OK~

### 17、yum命令

command not found的解决方法：yum –y install xxx

## 软件安装详解

通常Linux应用软件的安装包有以下几种：

1. run包，如myeclipse-10.0-offline-installer-linux.run
2. bin 包，如j2sdk-1\_4\_2\_07-linux-i586.bin
3. rpm包，如jdk-7u79-linux-i586.rpm，它是Redhat Linux提供的一种包封装格式，但是其原始设计理念是开放的，在其他版本的linux中也可以使用，算是公认的行业标准。
4. tar包，如apache-tomcat-6.0.44.tar.gz，它是使用UNIX系统的打包工具tar打包的
5. dpkg包，如sftware-1.2.3-1.deb，它是Debain Linux提供的一种包封装格式

### run包、bin包安装与卸载

1、安装： 将安装包拷贝到指定安装目录中 （cp ????.bin /usr/local/ 或 cp ????.run /usr/local/ ），再切换到安装目录 （/usr/local/），执行命令 （./?????bin 或 ./?????.run）进行安装即可，最后删除安装文件节省系统空间。

2、卸载 : 直接删除安装目录

### 2、tar 包安装与卸载

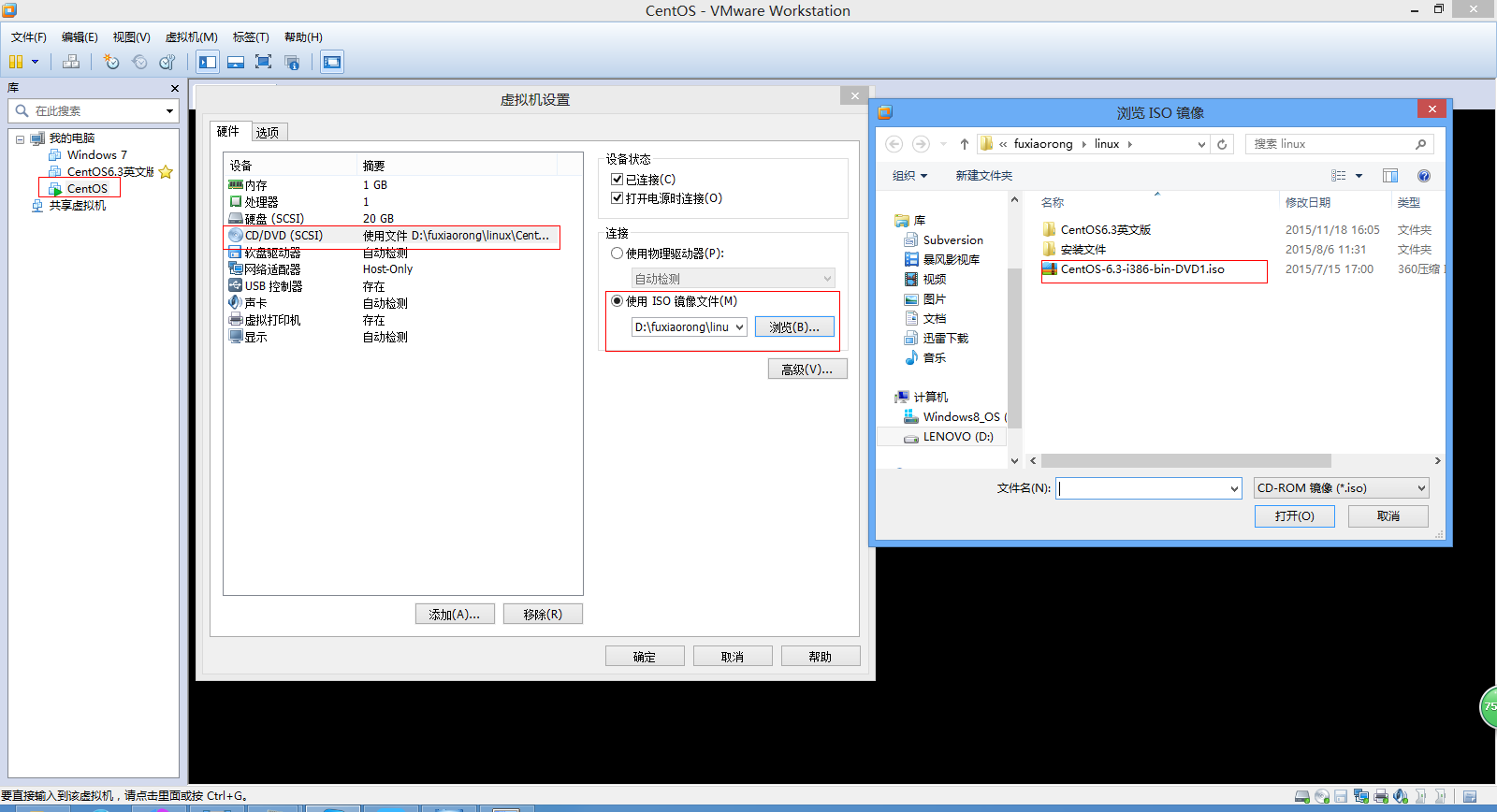
1. 安装：以后缀名.tar.gz的安装包，其实就是压缩包，只需要对其进行解压即可。把安装包拷贝到安装目录中（cp ?????.tar.gz /usr/local/ ），再切换到安装目录 （/usr/local/），执行命令（tar –zxvf ????.tar.gz）进行解压即可，最后删除安装文件节省系统空间
2. 卸载：由于tar包是压缩包，卸载的方式就是直接删除安装目录及子目录和文件（rm –rf apache-tomcat-6.0.44）

## shell编程

## 其他

### 本地yum配置

1. 虚拟机挂载CentOS-6.3-i386-bin-DVD1.iso ，linux系统iso文件

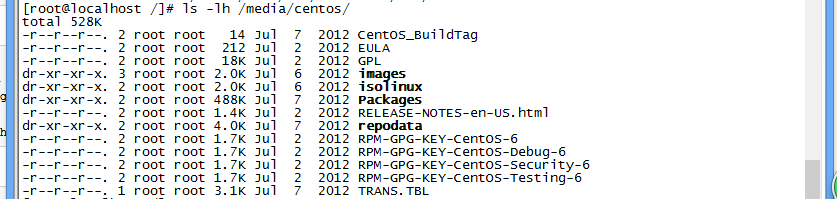


1. 启动CentOS，ISO默认挂载在/media目录下，如果在/media或者/mnt下无法找到挂载的文件，则按下面步骤：

* 光驱对应的设备文件是/dev/hdc或者/dev/cdrom或/dev/cdrom1，其中/dev/cdro和/dev/cdrom1文件是链式文件目录
* 挂载命令：mount 需要挂载的设备文件（光驱、u盘等） 将要挂载的文件目录

mount /dev/cdrom1 /media/centos/ (其中centos目录是手动创建的空目录)

* 使用上面的命令将光驱/dev/cdrom1文件挂载到/media/centos/目录下，之后就可以进入/media/centos/目录下对光驱里的文件操作



1. 在/opt目录下新建localserver目录，并且把系统安装包下package目录下所有的文件（repo文件）复制到/opt/localserver/目录中

|  |
| --- |
| [root@localhost opt]# mkdir localserver  [root@localhost opt]# cp -rv /media/centos/Packages/\* /opt/localserver/ |

1. 安装createrepo命令

|  |
| --- |
| [root@localhost opt]# cd localserver/  [root@localhost localserver]# rpm -ivh createrepo-0.9.9-17.el6.noarch.rpm  warning: createrepo-0.9.9-17.el6.noarch.rpm: Header V3 RSA/SHA1 Signature, key ID c105b9de: NOKEY  error: Failed dependencies:  python-deltarpm is needed by createrepo-0.9.9-17.el6.noarch |

以上提示安装失败，需要安装依赖包：python-deltarpm，如果提示多个依赖包，需要先把依赖包都安装完成

|  |
| --- |
| [root@localhost localserver]# rpm -ivh createrepo-0.9.9-17.el6.noarch.rpm python-deltarpm-3.5-0.5.20090913git.el6.x86\_64.rpm deltarpm-3.5-0.5.20090913git.el6.x86\_64.rpm  warning: createrepo-0.9.9-17.el6.noarch.rpm: Header V3 RSA/SHA1 Signature, key ID c105b9de: NOKEY  Preparing... ########################################### [100%]  1:deltarpm ########################################### [ 33%]  2:python-deltarpm ########################################### [ 67%]  3:createrepo ########################################### [100%] |

createrepo安装完成

1. 利用createrepo创建yum仓库，创建索引信息（需要等待一段时间）

|  |
| --- |
| [root@localhost localserver]# createrepo -v ./ |

1. 配置文件，将/etc/yum.repos.d/下现存文件都重命名为.bak，然后新建local.repo文件，并添加相应的内容

|  |
| --- |
| [root@sl yum.repos.d]# vi local.repo　　#repo后缀  #  # This repo can be used with mounted DVD media, verify the mount point for  # CentOS-6. You can use this repo and yum to install items directly off the  # DVD ISO that we release.  #  # To use this repo, put in your DVD and use it with the other repos too:  # yum --enablerepo=c6-media [command]  #  # or for ONLY the media repo, do this:  #  # yum --disablerepo=\\* --enablerepo=c6-media [command]  [local]　　　　　　　　　　　　　　 #库名称  name=CentOS-local　　 #名称描述  baseurl=file:///opt/localserver/ #yum源目录，源地址  gpgcheck=1　　　　　　　　　　 #检查GPG-KEY，0为不检查，1为检查  enabled=1　　　　　　　　　　　 #是否用该yum源，0为禁用，1为使用  gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6　　#gpgcheck=0时无需配置 |

1. 配置成功后，对yum进行测试：

|  |
| --- |
|  |

yum clean all : 清除缓存

yum makecache : 重新建立缓存

### RPM包管理

1. **概述**

一种用于互联网下载包的打包及安装工具，它包含在某些linux分发版中。它生成具 有.RPM扩展名的文件。RPM是RedHat Package Manager(RedHat软件包管理工具)的缩写。 这一文件格式名虽然打上了RedHat的标志，但是其原始设计理念是开放式的，其他Linux 的分发版本都有采用，可算是公认的行业标准。

1. **RPM包管理的用途**
2. 可以安装、删除、升级和管理软件，也支持在线安装和升级软件。
3. 通过RPM包管理能知道软件包包含哪些文件，也能知道系统中的某个文件属于哪 个软件包。
4. 可以查询系统中的软件包是否安装以及其版本。
5. 作为开发者可以把自己的程序打包为RPM包发布。
6. 软件包签名GPG和MD5的导入、验证和签名发布。
7. 依赖性的检查，查看是否有软件包由于不兼容而扰乱了系统。
8. **RPM的使用权限**

RPM软件的安装、删除、更新只有root权限才能使用，对于查询功能任何用户都可以 操作。如果普通用户拥有安装目录的权限，同样可以进行安装。

1. **命令格式**

|  |
| --- |
| rpm {-q|--query} [select-options] [query-options] |

常用参数：

|  |  |
| --- | --- |
| -i, --install | install package(s)（安装） |
| -v, --verbose | provide more detailed output（安装详情） |
| -h, --hash | 安装进度条 |
| -e, --erase | erase (uninstall) package（卸载） |
| -U, --upgrade=<packagefile>+ | upgrade package(s)（更新） |
| －-replacepkge | 无论软件包是否已被安装，都强行安装软件包 |
| --test | 安装测试，并不实际安装 |
| --nodeps | 忽略软件包的依赖关系强行安装或卸载 |
| --force | 忽略软件包及文件的冲突 |
| Query options (with -q or --query): | |
| -a, --all | query/verify all packages |
| -p, --package | query/verify a package file |
| -l, --list | list files in package |
| -d, --docfiles | list all documentation files |
| -f, --file | query/verify package(s) owning file |

常用命令组合：

|  |
| --- |
| －ivh：安装显示安装进度--install--verbose--hash －Uvh：升级软件包--Update； －qpl：列出RPM软件包内的文件信息[Query Package list]； －qpi：列出RPM软件包的描述信息[Query Package install package(s)]； －qf：查找指定文件属于哪个RPM软件包[Query File]； －Va：校验所有的RPM软件包，查找丢失的文件[View Lost]； －e：删除包 |

示例：

|  |
| --- |
| rpm -q samba //查询程序是否安装  rpm -ivh file.rpm 　// [安装新的rpm]--install--verbose--hash  rpm -ivh --relocate /=/opt/gaim gaim-1.3.0-1.fc4.i386.rpm //软件包指定安装目录  rpm -ivh  /media/cdrom/RedHat/RPMS/samba-3.0.10-1.4E.i386.rpm //按路径安装并显示进度  rpm -ivh --relocate /=/opt/gaim gaim-1.3.0-1.fc4.i386.rpm    //指定安装目录  rpm -ivh --test gaim-1.3.0-1.fc4.i386.rpm //用来检查依赖关系；并不是真正的安装  rpm -Uvh --oldpackage gaim-1.3.0-1.fc4.i386.rpm //新版本降级为旧版本  rpm -qa | grep httpd　　　　　 //[搜索指定rpm包是否安装]--all搜索\*httpd\*  rpm -ql httpd　　　　　　　　　//[搜索rpm包]--list所有文件安装目录  rpm -qpi Linux-1.4-6.i368.rpm //[查看rpm包]--query--package--install package信息  rpm -qpf Linux-1.4-6.i368.rpm　//[查看rpm包]--file  rpm -qpR file.rpm　　　　　　　//[查看包]依赖关系  rpm2cpio file.rpm |cpio -div    //[抽出文件]  rpm -Uvh file.rpm    //[升级一个rpm]--upgrade  rpm -e file.rpm      //[删除一个rpm包]--erase |

1. **软件的查询、查看**
2. 查询已安装的软件

语法：rpm -q 软件名

举例：

|  |
| --- |
| [root@localhost /]# rpm -q gzip  gzip-1.3.12-18.e16.i686 |

说明：-q 就是--query，“查询”的意思，此命令表示的是，是不是系统安装了gzip； 如果已安装则输出软件全称，否则输出未安装的信息。

1. 查看系统所有已安装的包

语法：rpm -qa

说明：-a 就是“所有”的意思，如果要分页查看则再加一个管道 | 和more 命令：

rpm -qa | more

1. 在所有已经安装的软件中查找某个软件

语法：rpm -qa | grep gzip

说明：grep命令是搜索命令

1. 查询一个已经安装的文件属于那个软件包

语法：rpm -qf 文件名

举例：

|  |
| --- |
| [root@localhost /]# rpm -qf /usr/lib/libacl.a  libacl-devel-2.2.49-6.el6.i686 |

说明：文件名所在的绝对路径要指出

1. 查询已安装软件包都安装到何处

语法：rpm -ql 软件名 或 rpmquery -ql 软件名

举例：

|  |
| --- |
| [root@localhost /]# rpm -ql libacl  /lib/libacl.so.1  /lib/libacl.so.1.1.0  [root@localhost /]# rpmquery -ql libacl  /lib/libacl.so.1  /lib/libacl.so.1.1.0 |

1. 查询一个已安装软件包的信息

语法：rpm -qi 软件名

举例：

|  |
| --- |
| [root@localhost /]# rpm -qi libacl  Name : libacl  Relocations: (not relocatable)  Version : 2.2.49  Vendor: CentOS  Release: 6.el6  Build Date: Thu 08 Dec 2011 12:51:12 AM CST  Install Date: Tue 17 Nov 2015 10:16:49 PM CST  Build Host: c6b6.bsys.dev.centos.org  Group: System Environment/Libraries  Source RPM: acl-2.2.49-6.el6.src.rpm  Size: 30364  ...... |

说明：-i 就是info 信息的意思

1. 查询已安装软件的配置文件

语法：rpm -qc 软件名

说明：-c 就是 configfiles 配置的意思

1. 查询已安装软件的文档安装位置

语法：rpm -qd 软件名

说明：-d 就是 docfiles 文档的意思

1. 查询已安装软件所依赖的软件包和文件

语法：rpm -qR 软件名

说明：-R 就是 requires 必须的意思

1. **rpm包的安装和升级**

语法：

|  |  |
| --- | --- |
| rpm -vih xxxxxx.rpm | 安装一个新的rpm包 |
| rpm -Uih xxxxxx.rpm | 升级一个rpm包 |

如果有依赖关系的，请解决依赖关系，或者强制安装。

--nodeps 忽略软件包的依赖关系强行安装

--force 忽略软件包及文件的冲突

语法结构：

|  |
| --- |
| rpm -vih xxxxxx.rpm --nodeps --force |
| rpm -Uih xxxxxx.rpm --nodeps --force |

举例应用：

|  |
| --- |
| [root@localhost localserver]# rpm -ivh lynx-xxxx.rpm Preparing... ############################ [100%]  1:lynx ############################ [100%] [root@localhost localserver]#rpm -ivh --replacepkgs xx.rpm Preparing... ############################ [100%]  1:xx ############################ [100%] |

注：--replacepkgs参数是已安装的软件再安装一次。

--------------------------------------------------------------------------------------------------------------------------

--test 参数是测试安装参数，用来检查依赖关系，并不是真正的安装

|  |
| --- |
| [root@localhost localserver]#rpm -ivh --test lynx-xx.rpm Preparing... ############################ [100%] |

-------------------------------------------------------------------------------------------------------------------------

--oldpackage 参数指的是由新版本降级为旧版本，执行更新命令

|  |
| --- |
| [root@localhost localserver]#rpm -Uvh --oldpackage lynx.rpm Preparing... ############################ [100%]  1:lynx ############################ [100%] |

-------------------------------------------------------------------------------------------------------------------------

--relocate 参数指的是为软件包指定安装目录，举例是把lynx.rpm指定安装在/opt/lynx 目录中

|  |
| --- |
| [root@localhost localserver]# rpm -ivh --relocate /=/opt/lynx lynx-xxxx.rpm Preparing... ############################ [100%]  1:lynx ############################ [100%] |

1. **删除rpm包**

语法：rpm -e 软件包名

说明：如果有依赖关系，可以用--nodeps参数来忽略依赖的检查进行删除。不过最好用 软件包管理器systerm-config-packages来删除或添加软件。

|  |
| --- |
| [root@localhost localserver]#rpm -e lynx --nodeps |

1. **updatedb、locate**

通过updatedb命令，可以保持已安装软件库最新，所以每隔一段时间也要执行一次。 然后我们可以用 [ locate 软件名或文件名 ] 来查询一些软件安装到哪里了。Updatedb 是slocate软件包的，如果没有这个命令，就得先安装slocate包。

### 文件共享NFS搭建（linux与linux之间）

1. **概述**

NFS 是Network File System的缩写，即网络文件系统。一种使用于分散式文件系统的协定，由Sun公司开发，于1984年向外公布。功能是通过网络让不同的机器、不同的操作系统能够彼此分享个别的数据，让应用程序在客户端通过网络访问位于服务器磁盘中的数据，是在类Unix系统间实现磁盘文件共享的一种方法。

NFS的基本原则是“容许不同的客户端及服务器通过一组RPC分享相同的文件系统”，它是独立于操作系统，容许不同硬件及操作系统的系统共同进行文件的分享。

NFS在文件传送或者消息传送过程中依赖于RPC协议。RPC，远程过程调用（Remote Procedure Call）是能使客户端执行其他系统程序的一种机制。NFS本身是没有提供信息传送的协议和功能的，但NFS却能让我们通过网络进行资料的分享，这是因为NFS使用了一些其他的传输协议。而这些传输协议用到这个RPC功能的。可以说NFS本身就是使用RPC的一个程序。或者说NFS也是一个RPC SERVER。所以只要用到NFS的地方都要启动RPC服务，不论是NFS SERVER或者NFS CLIENT。这样SERVER和CLIENT才能通过RPC来实现PROGRAM PORT的对应。可以这么理解RPC和NFS的关系：NFS是一个文件系统，而RPC是负责信息的传输。

1. **系统环境**

系统平台：CentOS release 6.3（Final）

NFS Server IP ：192.168.100.155

Client IP ：192.168.100.138

防火墙已关闭：service iptables status ---查看防火墙的状态

service iptables stop ---关闭防火墙

Service iptables start ---开启防火墙

1. **安装NFS服务**

NFS的安装是非常简单的，只需要两个软件包即可，通常情况下，是作为系统的默认包 安装好了。

|  |
| --- |
| * nfs-utils-\*：包括基本的NFS命令与监控程序 * rpcbind-\*：支持安全NFS RPC服务的连接 |

1. 查看系统是否已安装NFS（有些版本进行端口映射的不是rpcbind，而是portmap）

|  |
| --- |
|  |

系统默认已经安装了nfs-utils，rpcbind两个软件包，如果系统中没有安装NFS所需的软 件包，则需要手动进行安装，nfs-utils和rpcbind在系统关盘中都有，在本地yum查找 如下：

|  |
| --- |
|  |
| [root@localhost localserver]# rpm -ivh rpcbind-0.2.0-9.el6.i686.rpm  [root@localhost localserver]# rpm -ivh nfs-utils-1.2.3-26.el6.i686.rpm |

1. **NFS系统守护进程**

* **nfsd：**它是基本的NFS守护进程，主要功能是管理客户端是否能够登录服务器。
* **mountd：**它是RPC安装守护进程，主要功能是管理NFS的文件系统。当客户端顺利通过nfsd登录NFS服务器后，在使用NFS服务所提供的文件前，还必须通过文件使用权限的验证。它会读取NFS的配置文件/etc/exports来对比客户端权限。
* **rpcbind/portmap**：主要功能是进行端口映射工作。当客户端尝试连接并使用RPC服务器提供的服务（如NFS服务）时，rpcbind/portmap会将所管理的与服务对应的端口提供给客户端，从而使客户可以通过该端口向服务器请求服务。

1. **NFS服务器的配置**

NFS服务器的配置相对比较简单，只需要在相应的配置文件中进行设置，然后启动NFS 服务器即可。

NFS的常用目录

|  |  |
| --- | --- |
| /etc/exports | NFS服务的主要配置文件 |
| /usr/sbin/exportfs | NFS服务的管理命令 |
| /usr/sbin/shownmount | 客户端的查看命令 |
| /var/lib/nfs/etab | 记录NFS分享出来的目录的完整权限设定值 |
| /var/lib/nfs/xtab | 记录曾经登录过的客户端信息 |

NFS服务的配置文件为/etc/exports，这个文件是NFS的主要配置文件，不过系统并没有 默认值，所以这个文件不一定会存在，可能要使用vi手动建立，然后在文件里面写 入配置内容。

/etc/exports文件内容格式：

|  |
| --- |
| <输出目录> [客户端1 选项 (访问权限,用户映射,其他)] [客户端2 选项 (访问权限,用户映射,其他)] |

1. 输出目录：是指NFS系统中需要共享给客户机使用的目录。
2. 客户端：是指网络中可以访问这个NFS输出目录的计算机。

客户端常用的指定方式:

* 指定IP地址的主机：192.168.100.138
* 指定子网中的所有主机：192.168.100.0/24 192.168.100.0/255.255.255.0
* 指定域名的主机：david.bsmart.cn
* 指定域中的所有主机：\*.bsmart.cn
* 所有主机：\*

1. 选项：用来设置输出目录的访问权限、用户映射等。

主要有三类选项：

* 访问权限选项：

|  |  |
| --- | --- |
| ro | 设置输出目录【只读】权限 |
| rw | 设置输出目录【读写】权限 |

* 用户映射选项：

|  |  |
| --- | --- |
| all\_squash | 将远程访问的所有普通用户及所属组都映射为匿名用户或用户组（nfsnobody） |
| no\_all\_squash | 与all\_squash 取反（默认设置） |
| root\_squash | 将root用户及所属组都映射为匿名用户或用户组（默认设置） |
| no\_root\_squash | 与rootsquash取反 |
| anonuid=xxx | 将远程访问的所有用户都映射为匿名用户，并指定该用户为本地用户（UID=xxx） |
| anongid=xxx | 将远程访问的所有用户组都映射为匿名用户组账户，并指定该匿名用户组账户为本地用户组账户（GID=xxx） |

* 其他选项：

|  |  |
| --- | --- |
| secure | 限制客户端只能从小于1024的tcp/ip端口连接nfs服务器（默认设置） |
| insecure | 允许客户端从大于1024的tcp/ip端口连接服务器 |
| sync | 将数据同步写入内存缓冲区和磁盘中，效率低，单可以保证数据的一致性 |
| async | 将数据先保存在内存缓冲区中，必要时才写入磁盘 |
| wdely | 奖惩是否有相关的写操作，如果有则将这些写操作一起执行，这样可以提高效率（默认设置） |
| no\_wdely | 若有写操作则立即执行，应与sync配合使用 |
| subtree | 若输出目录是一个子目录，则nfs服务器将检查其父目录的权限（默认设置） |
| no\_subtree | 即使输出目录是一个子目录，nfs服务器也不检查其父目录的权限，这样可以提高效率 |

1. /etc/exports 文件内容示例：

/home/work 192.168.\*.\*(rw,sync,no\_root\_squash) \*(ro)

挂载/home/work，为192.168段ip地址的客户端均有读写权限，其他地址的客户 端只有读权限

1. **NFS服务器的启动、停止与自动启动**

在对exports文件进行了正确的配置后，就可以启动NFS服务器了。

1. 启动NFS服务器

为了使NFS服务器能正常工作，需要启动rpcbind/portmap和nfs两个服务，并且 rpcbind/portmap一定要先于nfs启动。

|  |
| --- |
| # service rpcbind start  # service nfs start |
|  |

1. 查看NFS服务器状态

|  |
| --- |
| # service rpcbind status  # service nfs status |
|  |

1. 停止NFS服务器

要停止NFS运行时，需要先停止nfs服务再停止rpcbind/portmap服务，对于系统 中有其他服务（如NIS）需要使用rpcbind/portmap时，不需要停止rpcbind/portmap 服务

|  |
| --- |
| # service rpcbind stop  # service nfs stop |

1. 设置NFS服务器的自动启动状态

对于实际的应用系统，每次启动linux系统后都要手动启动nfs服务器是不现实的， 需要设置系统在指定的运行级别自动启动rpcbind和nfs服务

|  |
| --- |
| # chkconfig --list rpcbind  # chkconfig --list nfs |
|  |

在CentOS6.3中，默认设置了rpcbind服务在运行级别2、3、4、5都自动启动，因 此还需要设置nfs服务在系统运行级别3、5自动启动

|  |
| --- |
| # chkconfig --list rpcbind  # chkconfig --list nfs |
|  |

重启linux系统后，查看rpcbind和nfs的状态

|  |
| --- |
|  |

1. **客户端使用NFS（挂载、卸载、自动挂载）**
2. 客户端使用showmount命令查询服务端NFS的共享状态

|  |
| --- |
| # showmount -e NFS服务器IP  ------------------------------------------------------------------------------------------------------------------  # showmount -e 192.168.100.155 |

1. 客户端挂载NFS服务器中的共享目录

|  |
| --- |
| # mount NFS服务器IP:服务器共享目录 本地挂载点目录  ------------------------------------------------------------------------------------------------------------------  # mount 192.168.100.155:/home/work/ /temp/david/  # mount | grep nfs |

1. 客户端卸载已挂载的NFS共享目录

|  |
| --- |
| # umount 本地挂载点目录  ------------------------------------------------------------------------------------------------------------------  # umount /temp/david/  # mount | grep nfs |

1. 自动挂载nfs文件系统

注：自动挂载nfs文件系统，服务器和客服端都需要关闭防火墙，意思就是说防火 墙不能随系统自动启动（关闭防火墙自动启动命令是：chkconfig iptables off）。

格式

|  |
| --- |
| <server>:</remote/export> </local/directory> nfs <options> 0 0  -------------------------------------------------------------------------------------  server : NFS服务器IP  /remote/export : 服务器共享目录  /local/directory : 本地挂载点目录  options : 参数设定，默认为defaults |

编辑/etc/fstab文件

|  |
| --- |
|  |

保存退出，重启系统，重启后查看/home/nfsdvid有没有自动挂载。

|  |
| --- |
|  |

自动挂载成功IMG_256。

1. **相关命令**
2. exportfs命令

如果在启动了NFS之后又修改了/etc/exports，是不是还要重启nfs呢？这个时候可 以使用exportfs命令来使改动立刻生效，格式如下：

# exportfs [-aruv]

-a : 全部挂载或卸载/etc/exports中的内容

-r : 重新读取/etc/exports中的信息，并同步更新/etc/exports、/var/lib/nfs/xtab

-u : 卸载单一目录（和-a 一起使用为卸载所有/etc/exports文件中的目录）

-v : 在exports的时候，将详细的信息输出到屏幕上。

具体例子：

# exportfs -au 卸载所有共享目录

# exportfs -rv 重新共享所有目录并数据详细信息

1. nfsstat 命令（列出NFS客户端和服务端的工作状态）

查看NFS的运行状态，对调整NFS的运行有很大的帮助

1. rpcinfo命令

查看rpc执行信息，可以用于检测rpc运行情况的工具，利用rpcinfo -p 可以查看 出RPC开启的端口所提供的程序有哪些。

1. showmount命令

-a 显示已经于客户端连接上的目录信息

-e [ip或者hostname] 显示此IP地址分享出来的目录

## 问题解决

### [禁止Centos系统You have new mail in /var/spool/mail/root提示](http://blog.csdn.net/oyym_mv/article/details/54018619)

该功能为Linux操作系统对系统资源状态并汇总，默认发送到root用户的/var/spool/mail/root目录，并在标准输出中提示该信息。该功能使用以下命令可以禁止：

|  |
| --- |
| [root@aridata sbin]# echo "unset MAILCHECK" >> /etc/profile  [root@aridata sbin]# source /etc/profile |