

# 尚硅谷大数据技术之 Linux 基础

(作者：大海哥)

官网：[www.atguigu.com](http://www.atguigu.com)

## 一、Linux 入门概述

### 1.1 概述

Linux 内核最初只是由芬兰人林纳斯 托瓦兹 (Linus Torvalds) 在赫尔辛基大学上学时出于个人爱好而编写的。

Linux 是一套免费使用和自由传播的类 Unix 操作系统，是一个基于 POSIX 和 UNIX 的多用户、多任务、支持多线程和多 CPU 的操作系统。Linux 能运行主要的 UNIX 工具软件、应用程序和网络协议。它支持 32 位和 64 位硬件。Linux 继承了 Unix 以网络为核心的设计思想，是一个性能稳定的多用户网络操作系统。

目前市面上较知名的发行版有：Ubuntu、RedHat、CentOS、Debian、Fedora、SuSE、OpenSUSE

### 1.2 下载地址

centos 下载地址：

网易镜像：<http://mirrors.163.com/centos/6/isos/>

搜狐镜像：<http://mirrors.sohu.com/centos/6/isos/>

### 1.3 Linux 特点

Linux 里面一切皆是文件

Linux 里面没有后缀名这一说

### 1.4 Linux 和 Windows 区别

目前国内 Linux 更多的是应用与服务上，而桌面操作系统更多使用的是 window。主要区别如下。

比较	Window	Linux
界面	界面统一，外壳程序固定所有 Windows 程序菜单几乎一致，快捷键也几乎相同	圆形界面风格依发布版本不同而不同，可能互不兼容。GNU/Linux 的终端机是从

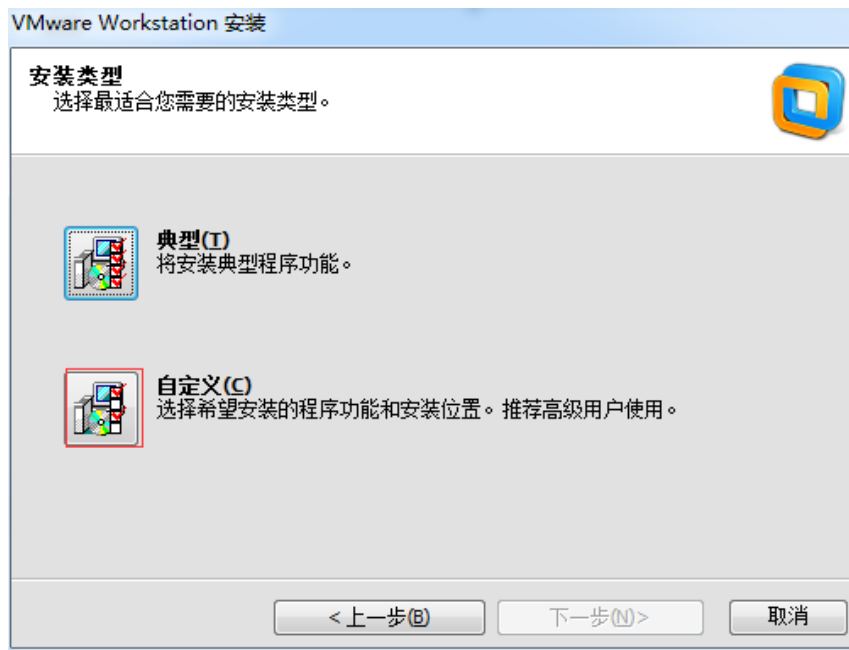
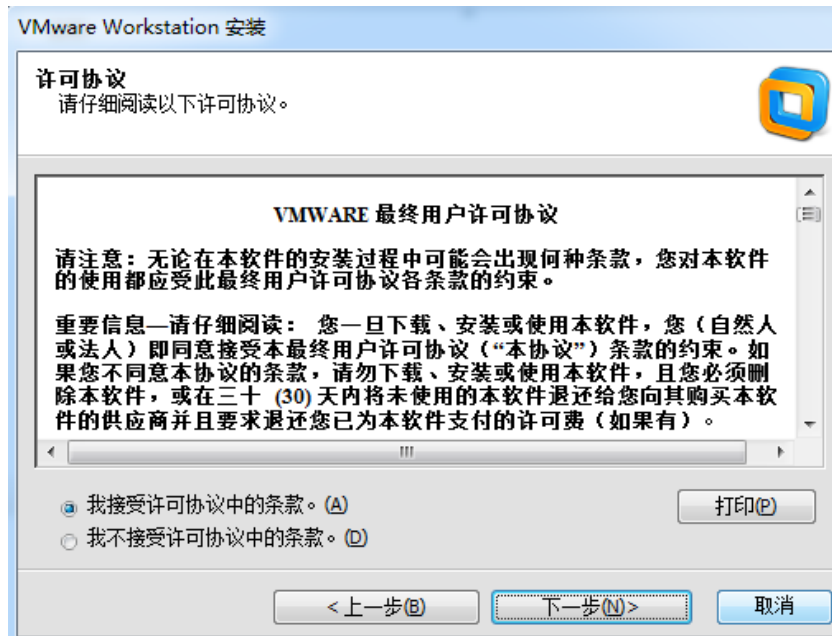
【更多 Java - Android 资料下载，可访问尚硅谷（中国）官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

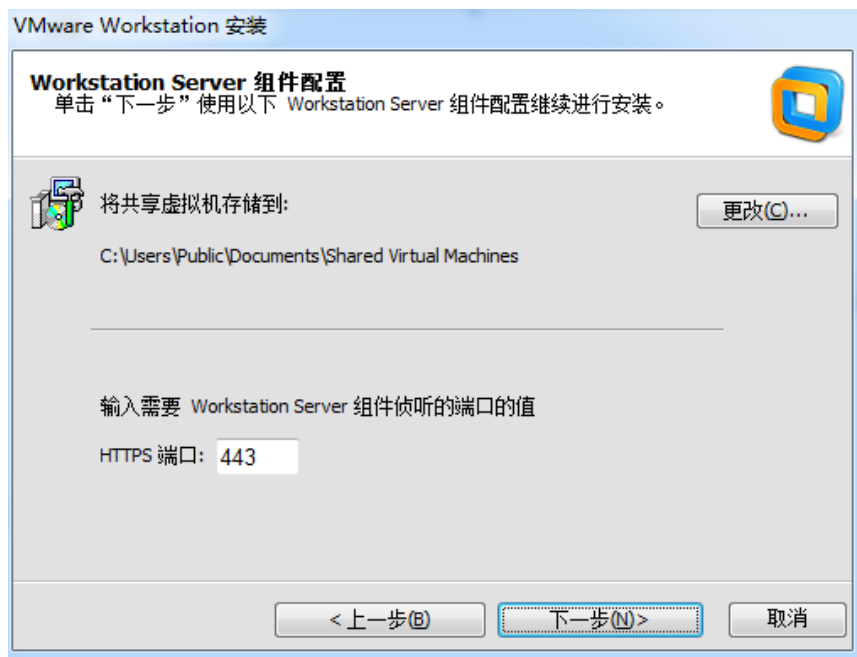
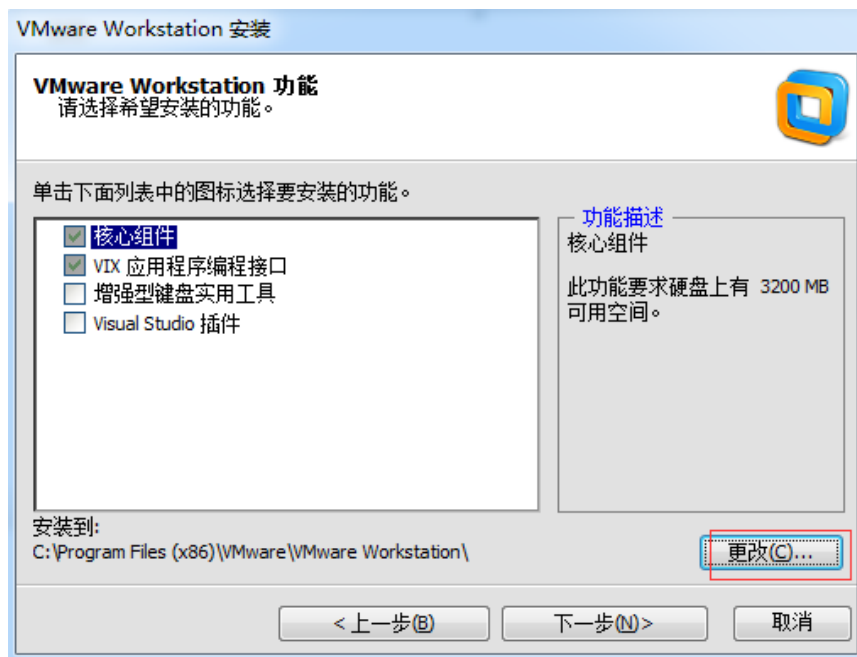
		UNIX 传承下来，基本命令和操作方法也几乎一致。
驱动程序	驱动程序丰富，版本更新频繁。默认安装程序里面一般包含有该版本发布时流行的硬件驱动程序，之后所出的新硬件驱动依赖于硬件厂商提供。对于一些老硬件，如果没有了原配的驱动有时候很难支持。另外，有时硬件厂商未提供所需版本的 Windows 下的驱动，也会比较头痛。	由志愿者开发，由 Linux 核心开发小组发布，很多硬件厂商基于版本考虑并未提供驱动程序，尽管多数无需手动安装，但是涉及安装则相对复杂，使得新用户面对驱动程序问题会一筹莫展。但是在开源开发模式下，许多老硬件尽管在 Windows 下很难支持的也容易找到驱动。HP、Intel、AMD 等硬件厂商逐步不同程度支持开源驱动，问题正在得到缓解。
使用	使用比较简单，容易入门。圆形化界面对没有计算机背景知识的用户使用十分有利。	圆形界面使用简单，容易入门。文字界面，需要学习才能掌握。
学习	系统构造复杂、变化频繁、且知识、技能淘汰快，深入学习困难	系统构造简单、稳定，且知识、技能传承性好，深入学习相对容易
软件	每一种特定功能可能都需要商业软件的支持，需要购买相应的授权	大部分软件都可以自由获取，同样功能的软件选择较少。

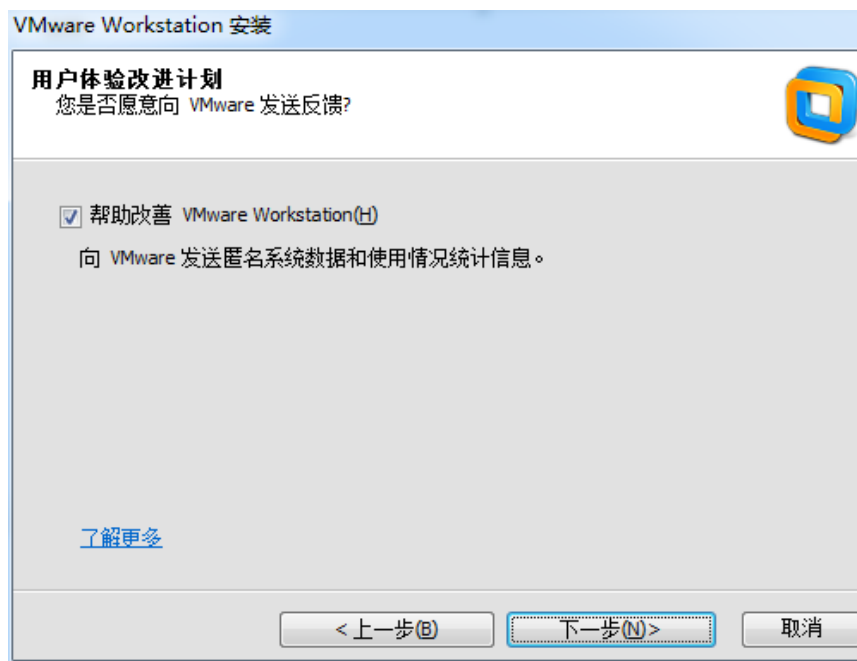
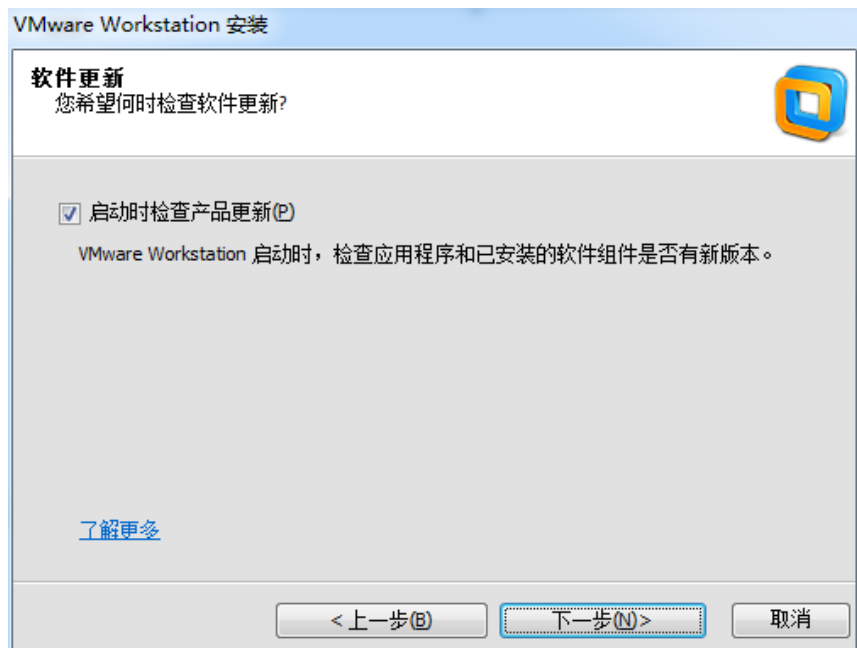
## 二、VM 安装相关

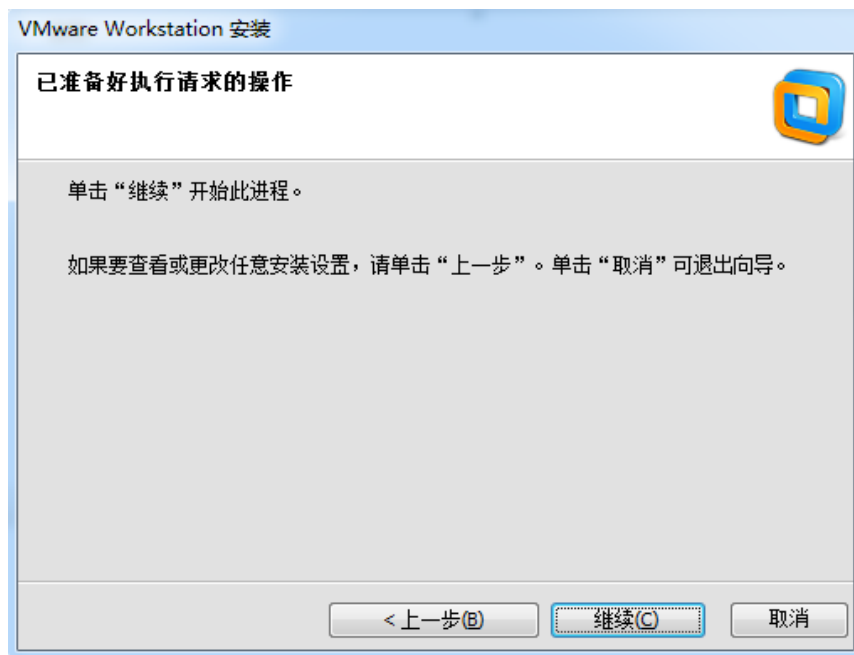
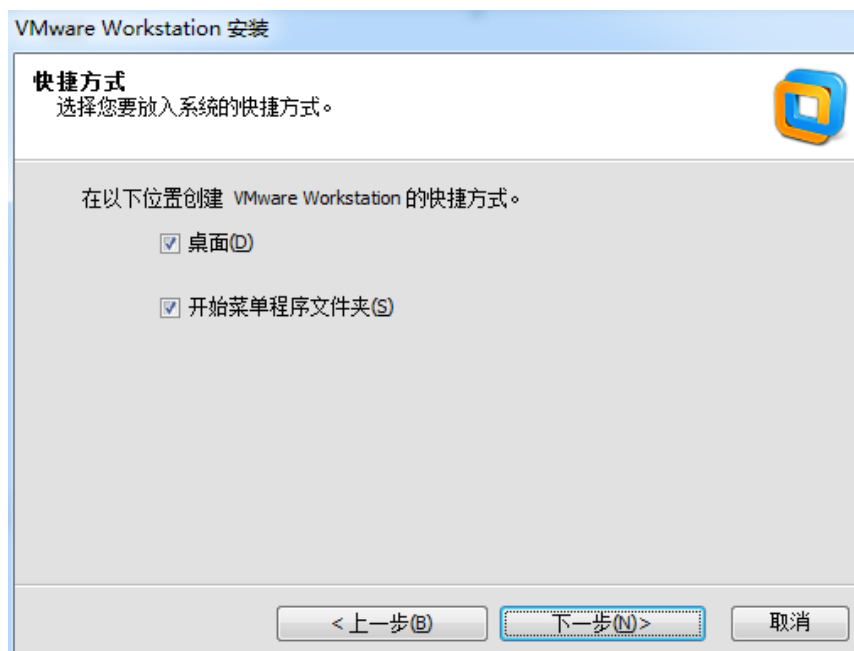
### 2.1 安装 VMWare 虚拟机

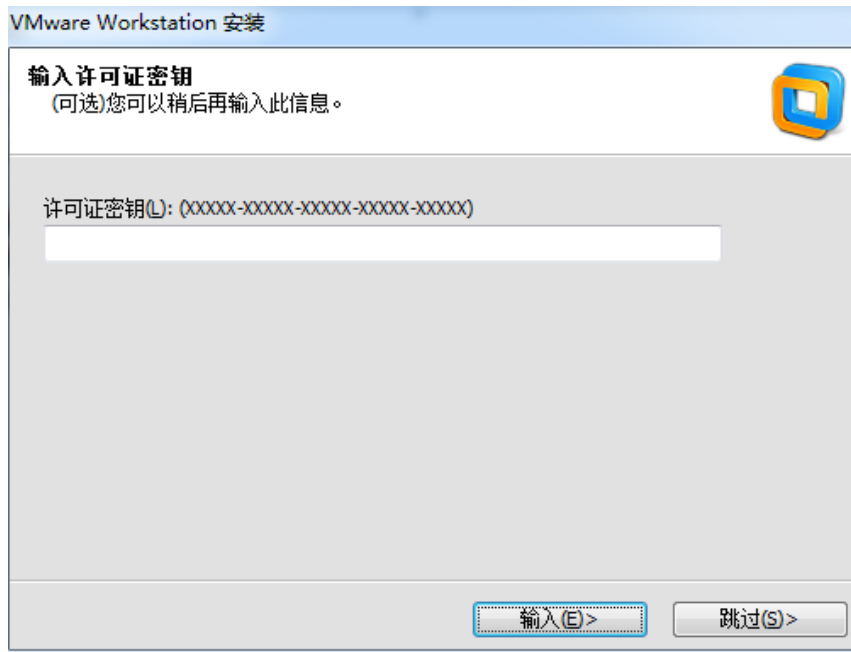
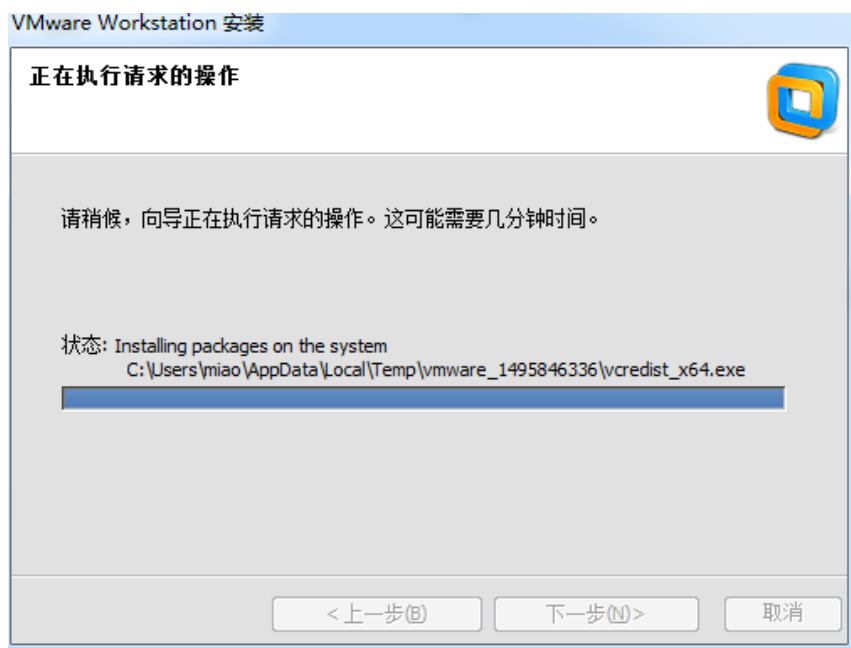


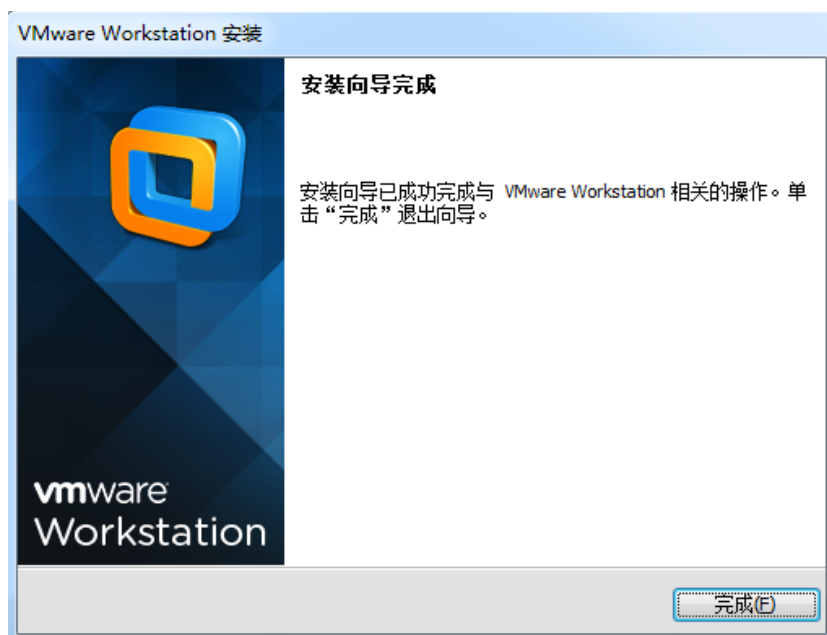
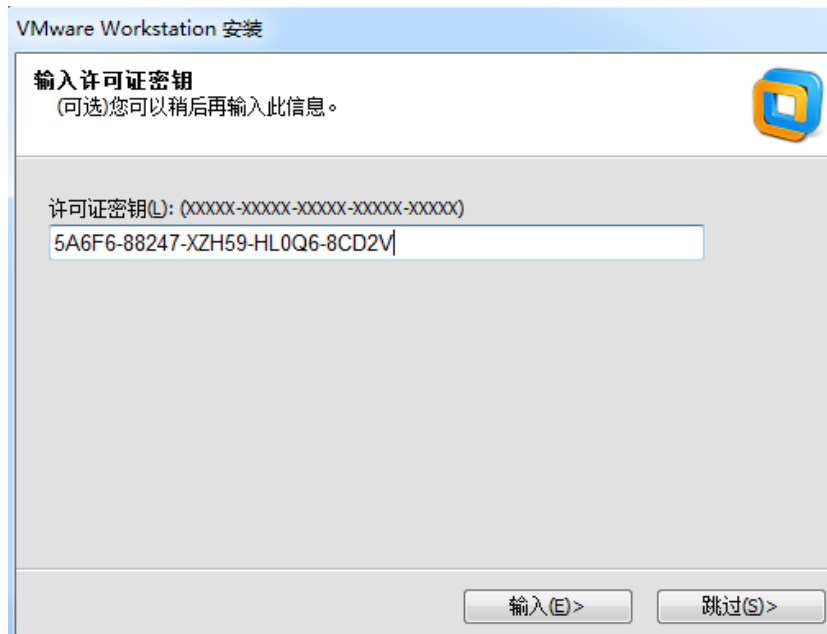












## 2.2 安装 CentOS



尚硅谷大数据技术  
之安装CentOS.doc

## 2.3 安装 VMTools 工具

### 1) 什么是 VMtools

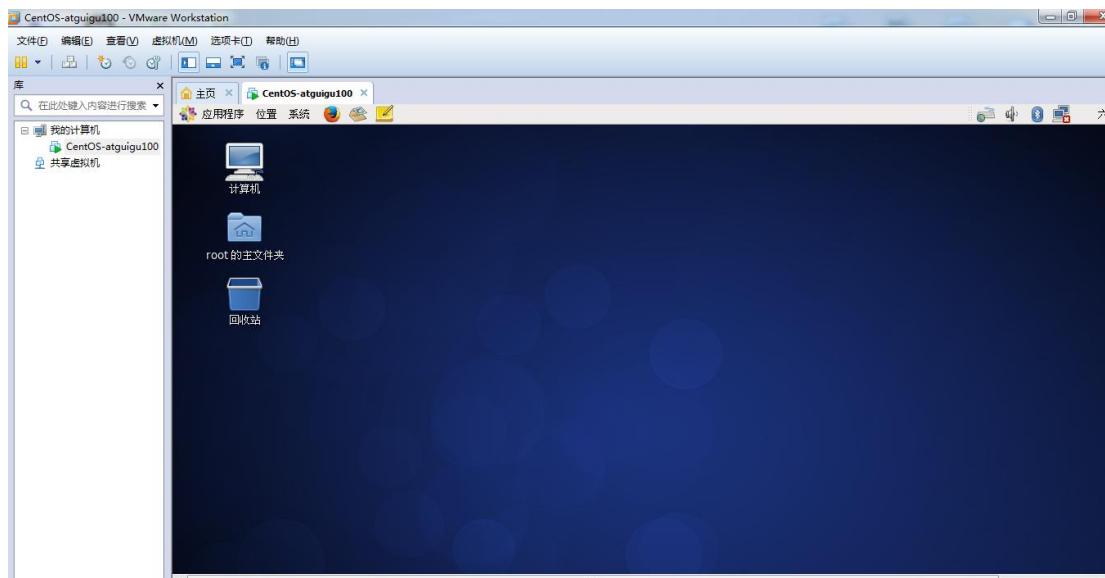
VM tools 顾名思义就是 VMware 的一组工具。主要用于虚拟主机显示优化与调整，另外

【更多 Java – Android 资料下载，可访问尚硅谷（中国）官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

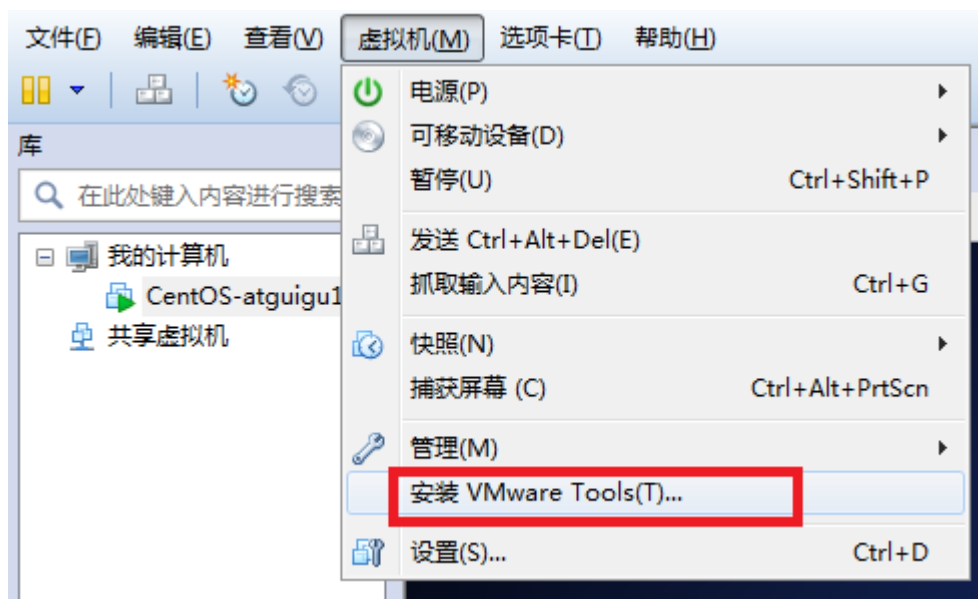


还可以方便虚拟主机与本机的交互，如允许共享文件夹，甚至可以直接从本机向虚拟主机拖放文件、鼠标无缝切换、显示分辨率调整等，十分实用。

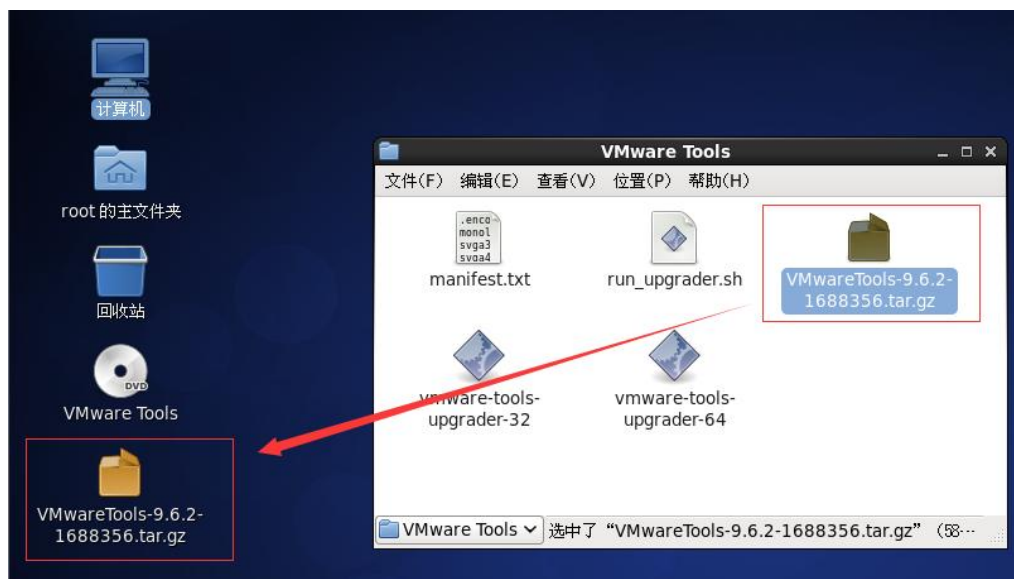
2) 先启动 CentOS 并成功登录如下图，发现底部提示且窗口中等大小，准备安装



3) 选择虚拟机菜单栏--安装 VMware tools



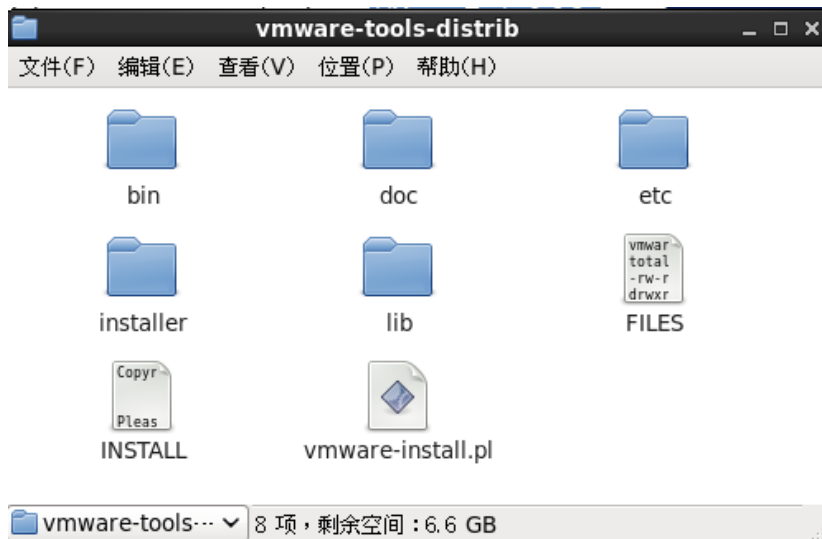
4) 光驱自动挂载 VMTools



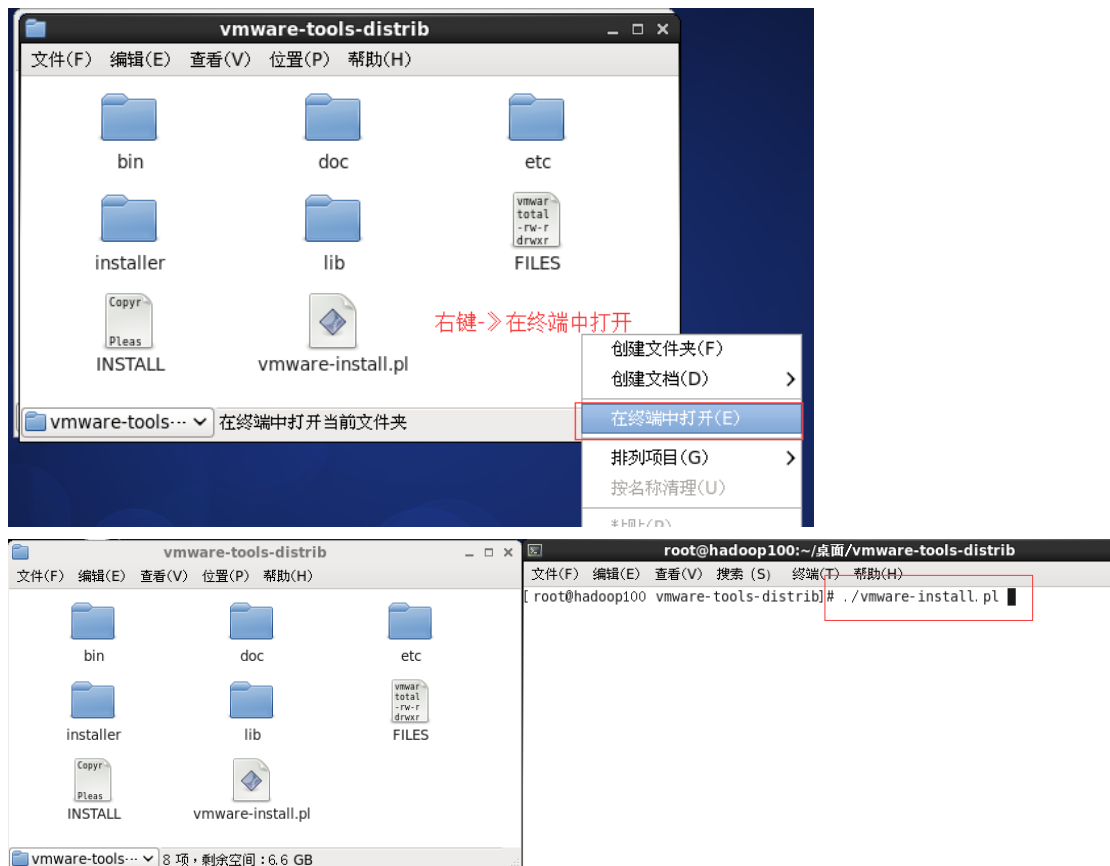
5) 右键解压 VMwaretools-9.6.2-1688356.tar.gz

进入文件夹并确认看到 vmware-install.pl 文件





6) 运行 vmware-install.pl 文件



最后用“./vmware-install.pl”命令来运行该安装程序，然后根据屏幕提示一路回车。到此整个安装过程算是完成了。

7) 直接按到/dev/hdc...停止为止，安装完成

```
root@hadoop100:~/桌面/vmware-tools-distrib
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
command line.

To enable advanced X features (e.g., guest resolution fit, drag and drop, and
file and text copy/paste), you will need to do one (or more) of the following:
1. Manually start /usr/bin/vmware-user
2. Log out and log back into your desktop session; and,
3. Restart your X session.

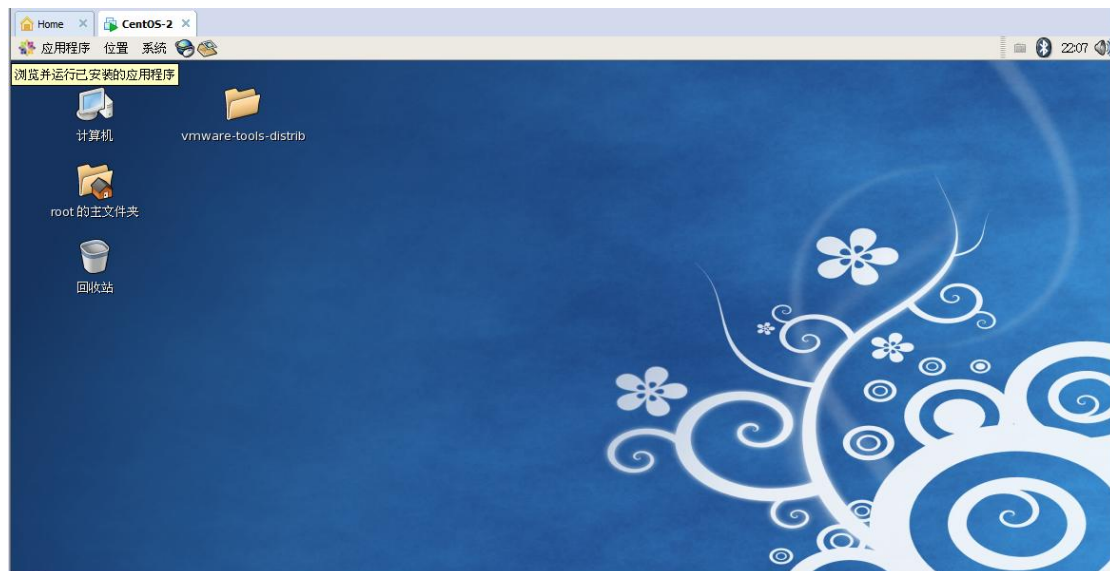
to use the vmxnet driver, either reboot or
1. stop networking or stop any interface using the vmxnet or pcnet32 driver
2. remove the pcnet32 module with 'rmmod pcnet32'
3. remove the vmxnet module with 'rmmod vmxnet'
4. load the vmxnet module with 'modprobe -v vmxnet'
5. and restart networking or restart the stopped network interfaces

Enjoy,

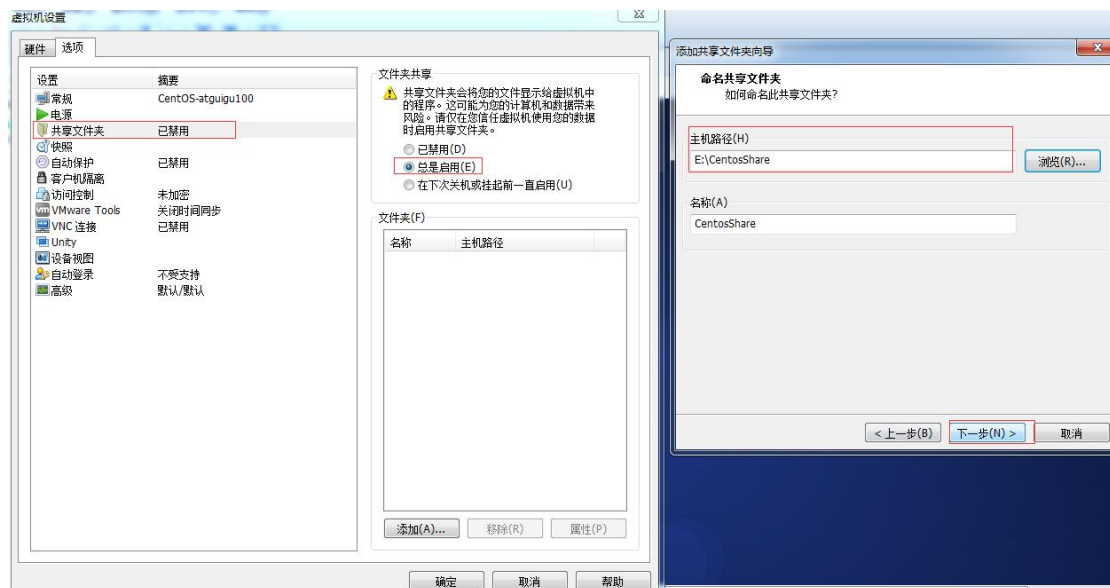
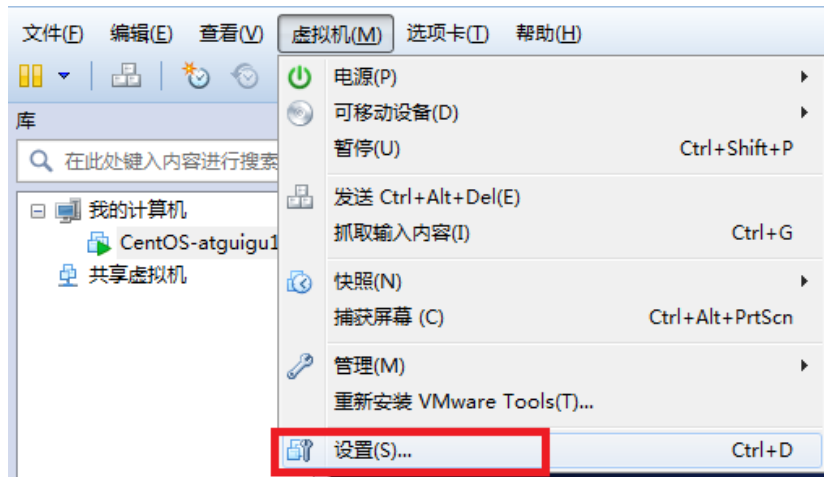
--the VMware team

Found VMware Tools CDRROM mounted at /media/VMware Tools. Ejecting device
/dev/sr0 ...
/sbin/restorecon: Warning no default label for /tmp/vmware-block-restore0/tmp_f
ile
[ root@hadoop100 vmware-tools-distrib]#
```

8) 重启 CentOS, 这时候屏幕变成全屏了



9) 设置共享文件夹, 实现 Windows  $\longleftrightarrow$  CentOS 文件共享



```
root@atguigu:/mnt/hgfs/centos2-share-window
文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)
[root@atguigu centos2-share-window]# pwd
/mnt/hgfs/centos2-share-window
[root@atguigu centos2-share-window]# ls -ll
总计 1
-rwxrwxrwx 1 root root 17 10-19 22:16 abc.txt
[root@atguigu centos2-share-window]# more abc.txt
aaaaaaaaaaaaaaaaaaaa
[root@atguigu centos2-share-window]#
```

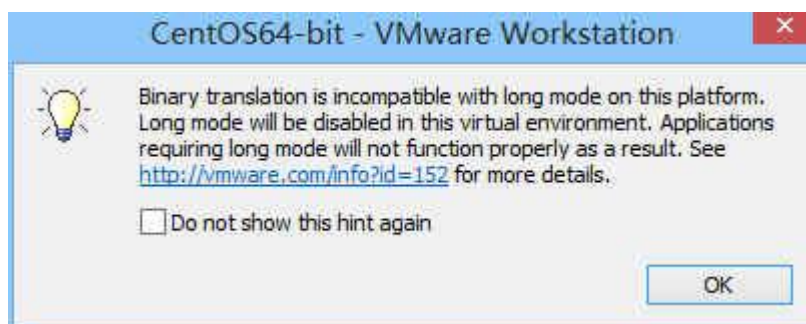
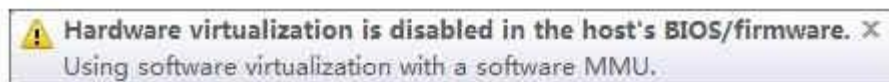
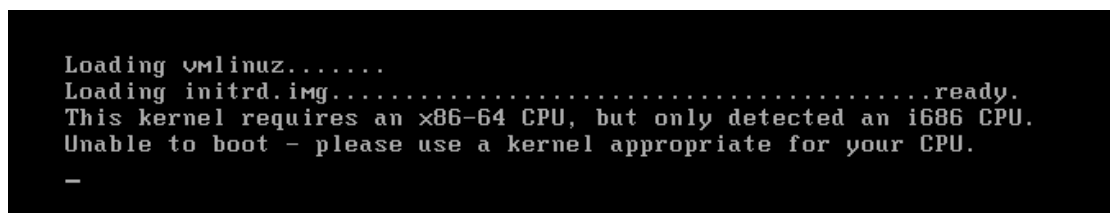
## 2.4 虚拟机屏幕保护设置



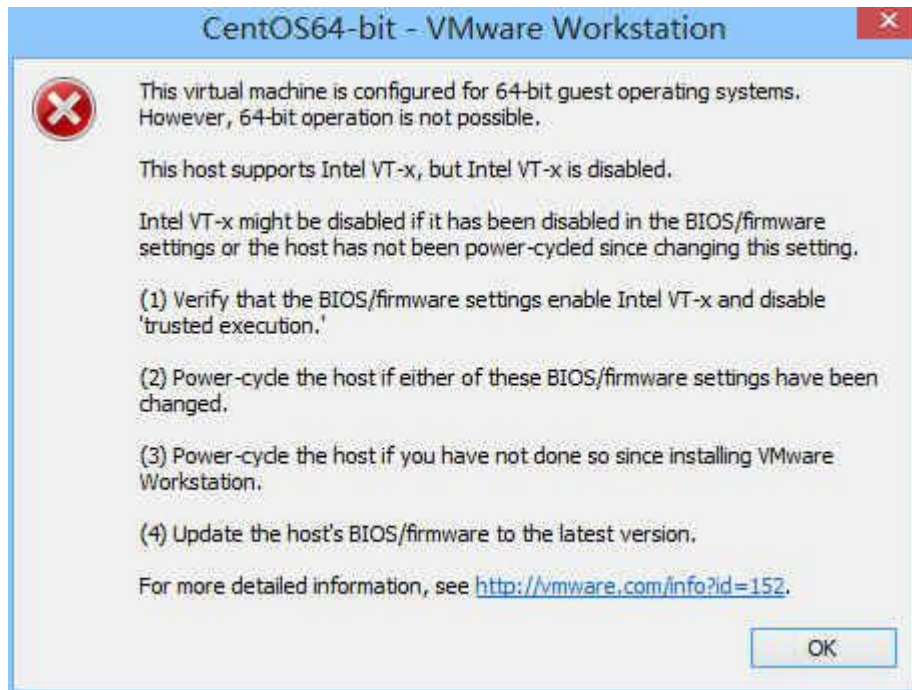


## 2.5 IVT 虚拟化支持

### 1) 异常情况

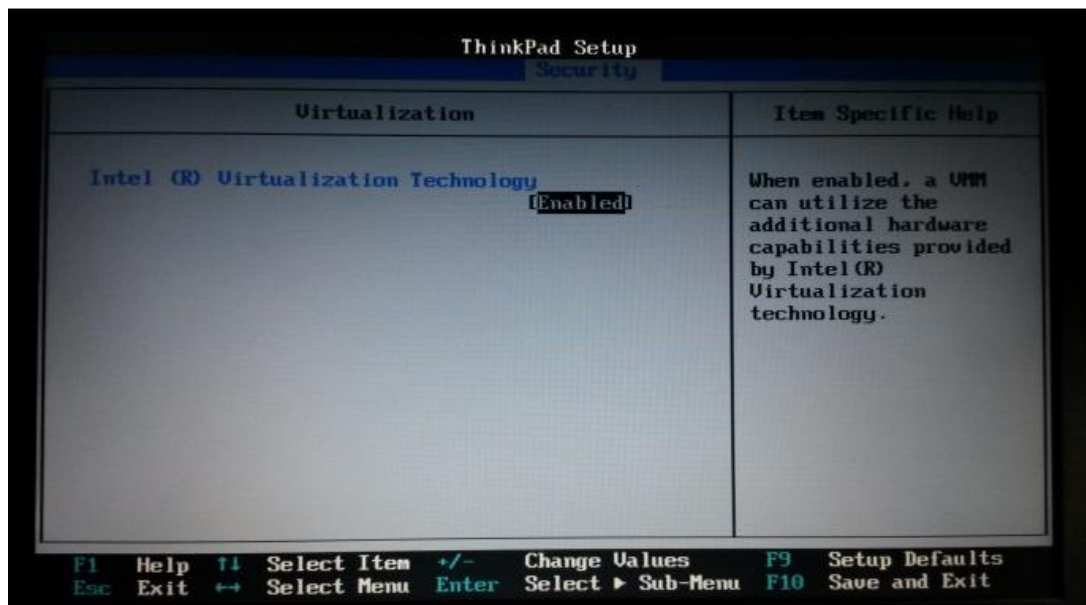






2) 宿主机 BIOS 设置中的硬件虚拟化被禁用了

需要打开笔记本 BIOS 中的 IVT 对虚拟化的支持



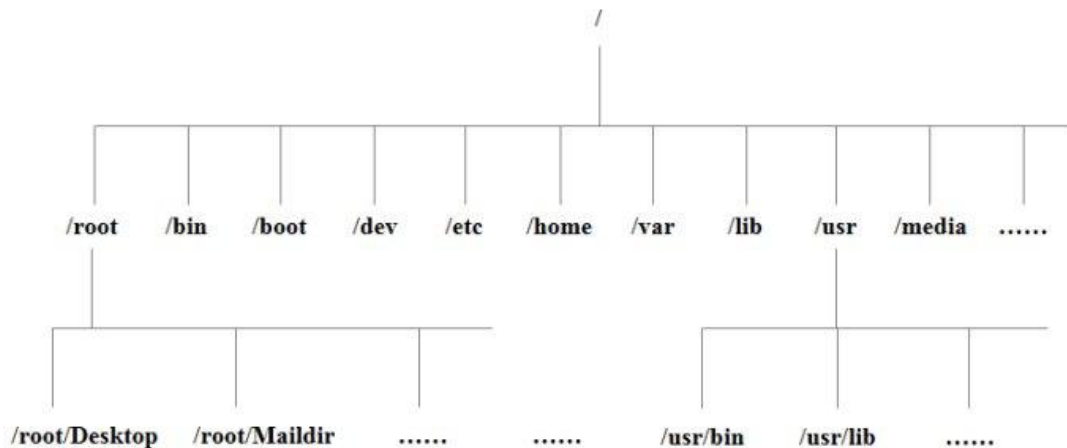


## 三、Linux 目录结构

### 3.1 概览

```
[root@cloud ~]# pwd
/
[root@cloud ~]# ls -l
总计 146
drwxr-xr-x  2 root root  4096 10-08 12:24 bin
drwxr-xr-x  4 root root  1024 10-08 01:33 boot
drwxr-xr-x 11 root root  4380 10-23 14:41 dev
drwxr-xr-x 94 root root 12288 10-23 14:42 etc
drwxr-xr-x  3 root root  4096 10-22 12:26 home
drwxr-xr-x 13 root root  4096 10-08 12:24 lib
drwxr-xr-x  2 root root 16384 10-08 01:26 lost+found
drwxr-xr-x  3 root root  4096 10-23 14:42 media
drwxr-xr-x  2 root root    0 10-23 14:41 misc
drwxr-xr-x  3 root root  4096 10-07 23:03 mnt
drwxr-xr-x  2 root root    0 10-23 14:41 net
drwxr-xr-x  2 root root  4096 2010-01-27 opt
dr-xr-xr-x 142 root root    0 10-23 14:40 proc
drwxr-xr-x 22 root root  4096 10-23 15:24 root
drwxr-xr-x  2 root root 12288 10-08 12:24 sbin
drwxr-xr-x  2 root root  4096 10-08 01:27 selinux
drwxr-xr-x  2 root root  4096 2010-01-27 srv
drwxr-xr-x 11 root root    0 10-23 14:40 sys
drwxrwxrwt 15 root root  4096 10-23 15:25 tmp
drwxr-xr-x 16 root root  4096 10-15 09:10 usr
drwxr-xr-x 22 root root  4096 10-08 01:35 var
```

### 3.2 树状目录结构



/bin: 是 Binary 的缩写，这个目录存放着系统必备执行命令

/boot: 这里存放的是启动 Linux 时使用的一些核心文件，包括一些连接文件以及镜像文件，自己的安装别放这里

/dev: Device(设备)的缩写，该目录下存放的是 Linux 的外部设备，在 Linux 中访问设备的方式和访问文件的方式是相同的。

【更多 Java - Android 资料下载，可访问尚硅谷（中国）官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

**/etc:** 所有的系统管理所需要的配置文件和子目录。

**/home:** 存放普通用户的主目录，在Linux中每个用户都有一个自己的目录，一般该目录名是以用户的账号命名的。

**/lib:** 系统开机所需要最基本的动态连接共享库，其作用类似于Windows里的DLL文件。几乎所有的应用程序都需要用到这些共享库。

**/lost+found:** 这个目录一般情况下是空的，当系统非法关机后，这里就存放了一些文件。

**/media:** linux系统会自动识别一些设备，例如U盘、光驱等等，当识别后，linux会把识别的设备挂载到这个目录下。

**/misc:** 该目录可以用来存放杂项文件或目录，即那些用途或含义不明确的文件或目录可以存放在该目录下。

**/mnt:** 系统提供该目录是为了让用户临时挂载别的文件系统的，我们可以将光驱挂载在/mnt/上，然后进入该目录就可以查看光驱里的内容了。

**/net** 存放着和网络相关的一些文件。

**/opt:** 这是给主机额外安装软件所摆放的目录。比如你安装一个ORACLE数据库则就可以放到这个目录下。默认是空的。

**/proc:** 这个目录是一个虚拟的目录，它是系统内存的映射，我们可以通过直接访问这个目录来获取系统信息。

**/root:** 该目录为系统管理员，也称作超级权限者的用户主目录。

**/sbin:** s就是Super User的意思，这里存放的是系统管理员使用的系统管理程序。

**/selinux:** 这个目录是Redhat/CentOS所特有的目录，Selinux是一个安全机制，类似于windows的防火墙

**/srv:** service缩写，该目录存放一些服务启动之后需要提取的数据。

**/sys:** 这是linux2.6内核的一个很大的变化。该目录下安装了2.6内核中新出现的一个文件系统 sysfs 。

**/tmp:** 这个目录是用来存放一些临时文件的。

**/usr:** 这是一个非常重要的目录，用户的很多应用程序和文件都放在这个目录下，类似与windows下的program files目录。

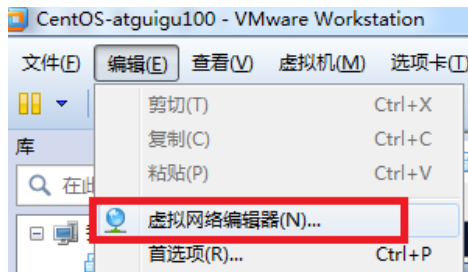
**/var:** 这个目录中存放着在不断扩充着的东西，我们习惯将那些经常被修改的目录放在这个目录下。包括各种日志文件。

【更多 Java – Android 资料下载，可访问尚硅谷（中国）官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

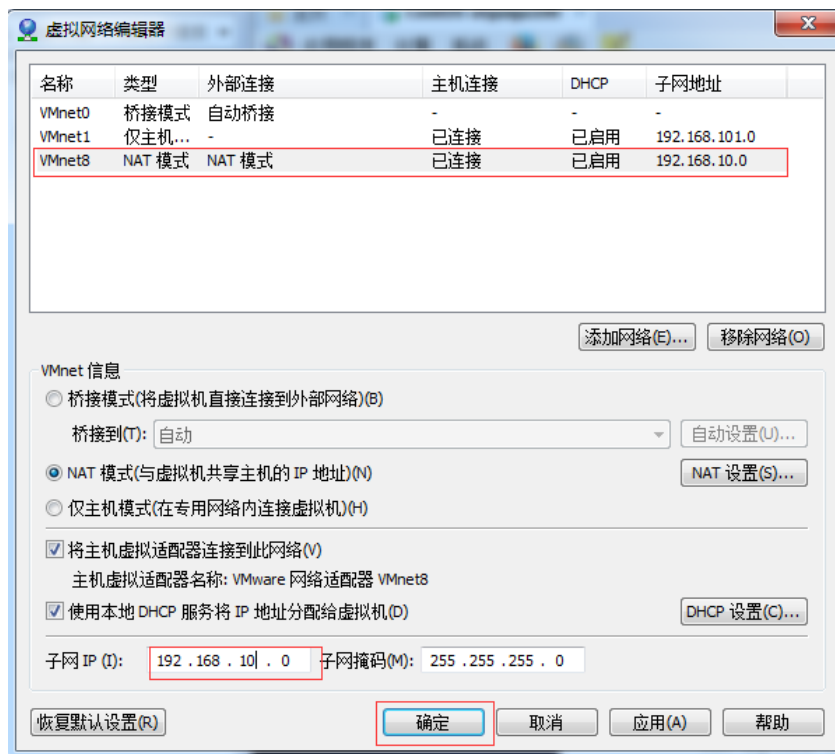
## 四、系统管理操作

### 4.1 查看网络 IP 和网关

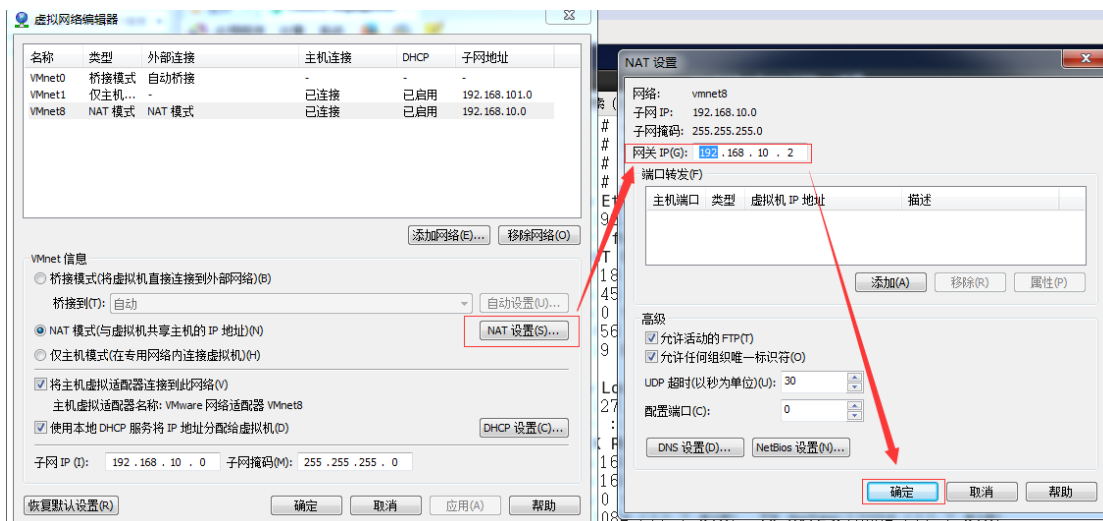
#### 1) 查看虚拟网络编辑器



#### 2) 修改 ip 地址



#### 3) 查看网关



## 4.2 配置网络 ip 地址

0) 查看当前ip基本语法:

```
[root@hadoop102 ~]# ifconfig
```

1) 在终端命令窗口中输入

```
[root@hadoop102 ~]# vim /etc/udev/rules.d/70-persistent-net.rules
```

进入如下页面，删除 eth0 该行；将 eth1 修改为 eth0，同时复制物理 ip 地址

```
# PCI device 0x1022:0x2000 (vmxnet)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="08:00:27:0c:08:34", ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"
# PCI device 0x1022:0x2000 (vmxnet)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:0c:29:34:c4:3f", ATTR{type}=="1", KERNEL=="eth*", NAME="eth1"
```

删除该行  
复制该地址  
修改为eth0

2) 修改 IP 地址

```
[root@hadoop102 ~]# vim /etc/sysconfig/network-scripts/ifcfg-eth0
```

需要修改的内容有 5 项:

IPADDR=192.168.11.106

GATEWAY=192.168.11.2

ONBOOT=yes

BOOTPROTO=static

DNS1=8.8.8.8

(1) 修改前

```
DEVICE=eth0
TYPE=Ethernet
UUID=109fb0a1-4949-4e2b-80dd-3d72ad223e33
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=none      修改为static
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME="System eth0"
HWADDR=08:0c:29:0c:08:34  地址是从刚才复制的地址中粘贴过来
IPADDR=192.168.10.102    修改为想要的IP地址
GATEWAY=192.168.10.2
LAST_CONNECT=1490351033
NETMASK=255.255.255.0
USERCTL=no
PEERDNS=yes
```

(2) 修改后

```
TYPE=Ethernet
UUID=50a63ff6-0bb5-42cd-8c9e-1386ace2608f
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static

DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME="System eth0"
HWADDR=00:0C:29:AD:6E:83
IPADDR=192.168.11.106
PREFIX=24
GATEWAY=192.168.11.2
DNS1=8.8.8.8
```

: wq 保存退出

3) 执行 service network restart

```
关闭环回接口 : [确定]
弹出环回接口 : [确定]
弹出界面 eth0 : 错误 : 激活连接失败 : The connection is not for this device.
[失败]
```

4) 如果报错, reboot, 重启虚拟机

## 4.3 配置主机名

0) 查看主机名基本语法:

```
[root@hadoop102 ~]# hostname
```

1) 修改 linux 的 hosts 文件

(1) 进入 Linux 系统查看本机的主机名。通过 hostname 命令查看

```
[root@hadoop ~]# hostname
```

【更多 Java - Android 资料下载, 可访问尚硅谷 (中国) 官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

hadoop1.atguigu.com

(2) 如果感觉此主机名不合适，我们可以进行修改。通过编辑/etc/sysconfig/network 文件

```
[root@hadoop102 /]# vi /etc/sysconfig/network
```

文件中内容

NETWORKING=yes

NETWORKING\_IPV6=no

HOSTNAME= **hadoop102**

**注意：主机名称不要有“\_”下划线**

(3) 打开此文件后，可以看到主机名。修改此主机名为我们想要修改的主机名 hadoop102。

(4) 保存退出。

(5) 打开/etc/hosts

```
[root@hadoop102 /]# vim /etc/hosts
```

添加如下内容

192.168.11.102 hadoop102

(6) 并重启设备，重启后，查看主机名，已经修改成功

## 2) 修改 window7 的 hosts 文件

(1) 进入 C:\Windows\System32\drivers\etc 路径

(2) 打开 hosts 文件并添加如下内容

192.168.11.101 hadoop101

192.168.11.102 hadoop102

192.168.11.103 hadoop103

192.168.11.104 hadoop104

192.168.11.105 hadoop105

192.168.11.106 hadoop106

192.168.11.107 hadoop107

192.168.11.108 hadoop108

## 4.4 防火墙

### 1) 基本语法:

- `service iptables status` (功能描述: 查看防火墙状态)
- `chkconfig iptables --list` (功能描述: 查看防火墙开机启动状态)
- `service iptables stop` (功能描述: 临时关闭防火墙)
- `chkconfig iptables off` (功能描述: 关闭防火墙开机启动)
- `chkconfig iptables on` (功能描述: 开启防火墙开机启动)

### 2) 扩展

Linux 系统有 7 个运行级别(runlevel)

运行级别 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

运行级别 1: 单用户工作状态, root 权限, 用于系统维护, 禁止远程登陆

运行级别 2: 多用户状态(没有 NFS)

运行级别 3: 完全的多用户状态(有 NFS), 登陆后进入控制台命令行模式

运行级别 4: 系统未使用, 保留

运行级别 5: X11 控制台, 登陆后进入图形 GUI 模式

运行级别 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

## 4.5 关机重启

在 linux 领域内大多用在服务器上, 很少遇到关机的操作。毕竟服务器上跑一个服务是永无止境的, 除非特殊情况下, 不得已才会关机。

正确的关机流程为: `sync > shutdown > reboot > halt`

### 1) 基本语法:

- (1) `sync` (功能描述: 将数据由内存同步到硬盘中)
- (2) `shutdown [选项] 时间`
  - 选项:
  - h: 关机
  - r: 重启
- (3) `halt` (功能描述: 关闭系统, 等同于 `shutdown -h now` 和 `poweroff`)
- (4) `reboot` (功能描述: 就是重启, 等同于 `shutdown -r now`)

### 2) 案例

【更多 Java – Android 资料下载, 可访问尚硅谷(中国)官网 [www.atguigu.com](http://www.atguigu.com) 下载区】



- (1) 将数据由内存同步到硬盘中

```
[root@hadoop102 ~]#sync
```

- (2) 计算机将在 10 分钟后关机，并且会显示在登录用户的当前屏幕中

```
[root@hadoop102 ~]#shutdown -h 10 'This server will shutdown after 10 mins'
```

- (3) 立马关机

```
[root@hadoop102 ~]# shutdown -h now
```

- (4) 系统立马重启

```
[root@hadoop102 ~]# shutdown -r now
```

- (5) 重启（等同于 shutdown -r now）

```
[root@hadoop102 ~]# reboot
```

- (6) 关机（等同于 shutdown -h now 和 poweroff）

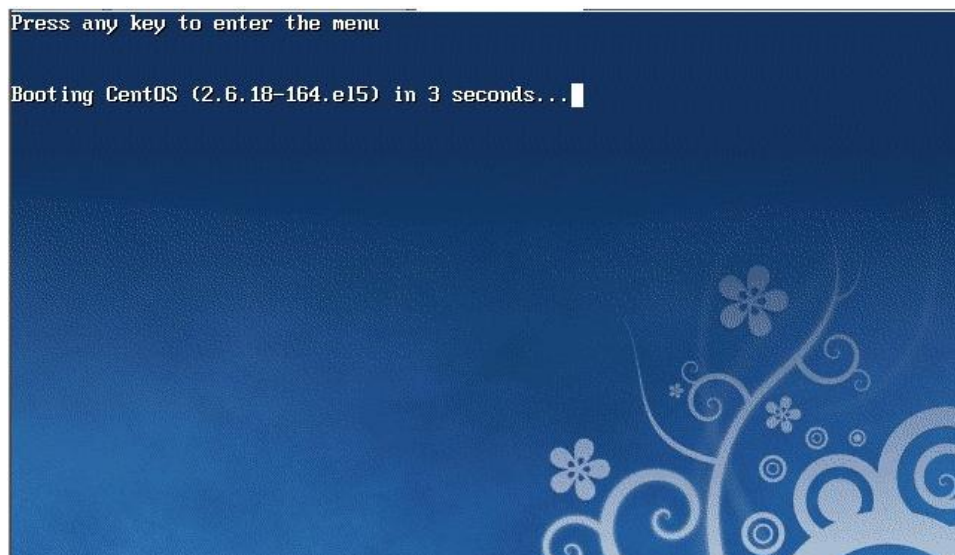
```
[root@hadoop102 ~]#halt
```

注意：不管是重启系统还是关闭系统，首先要运行 `sync` 命令，把内存中的数据写到磁盘中。

## 4.6 找回 root 密码

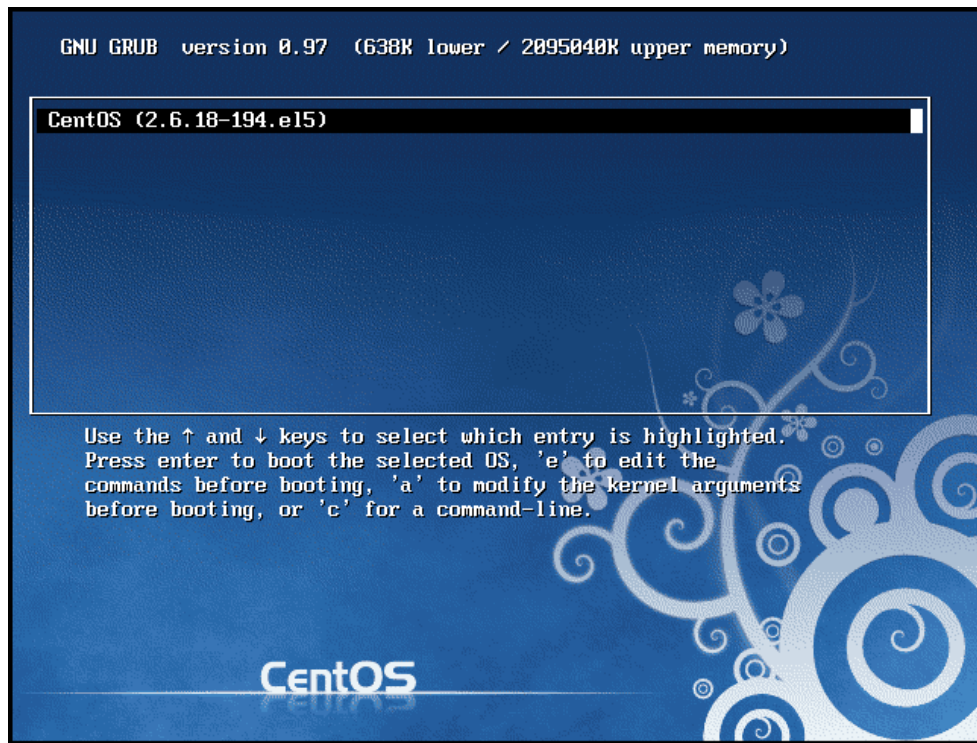
重新安装系统吗？当然不用！进入单用户模式更改一下 root 密码即可。

- 1) 重启 Linux，见到下图，在 3 秒钟之内按下回车

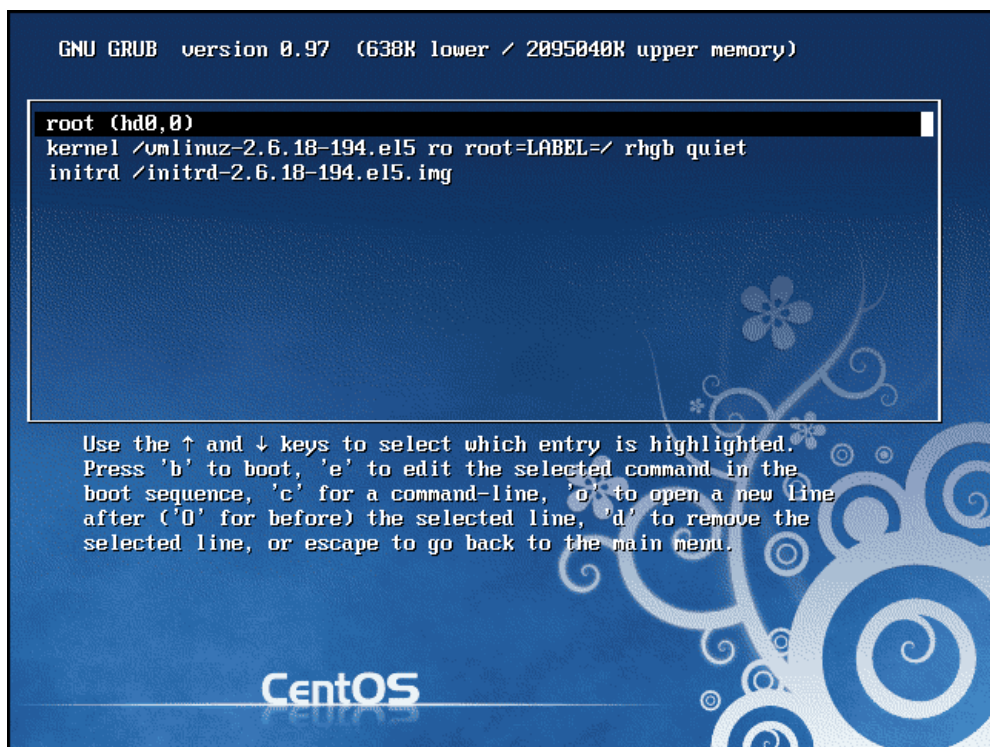


- 2) 三秒之内要按一下回车，出现如下界面

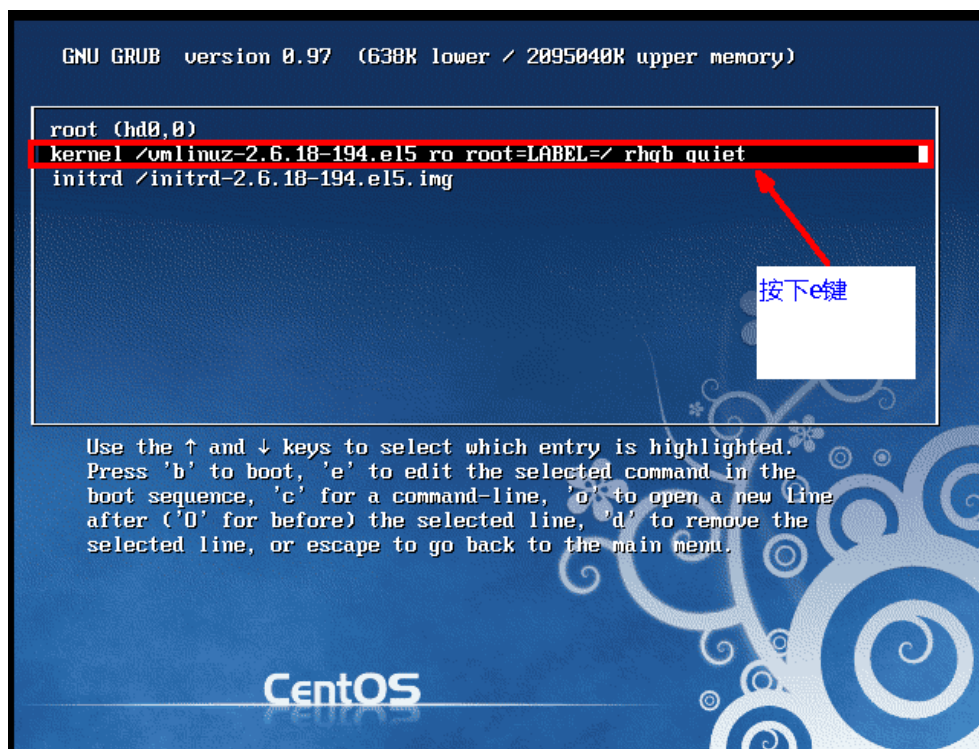




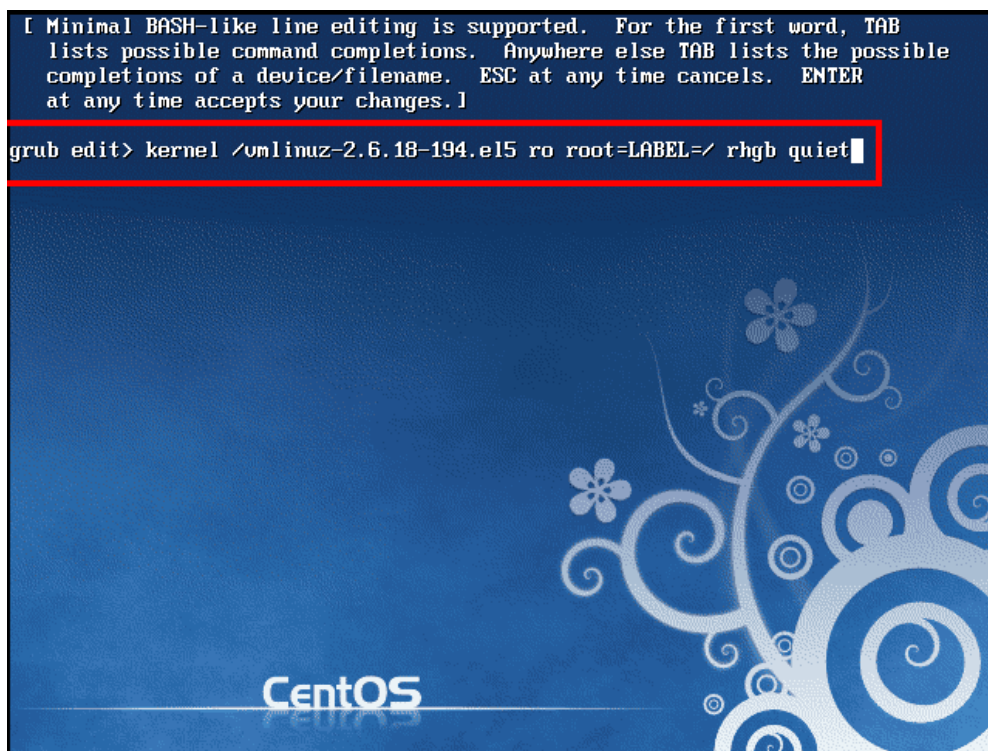
3) 按下 e 键就可以进入下图



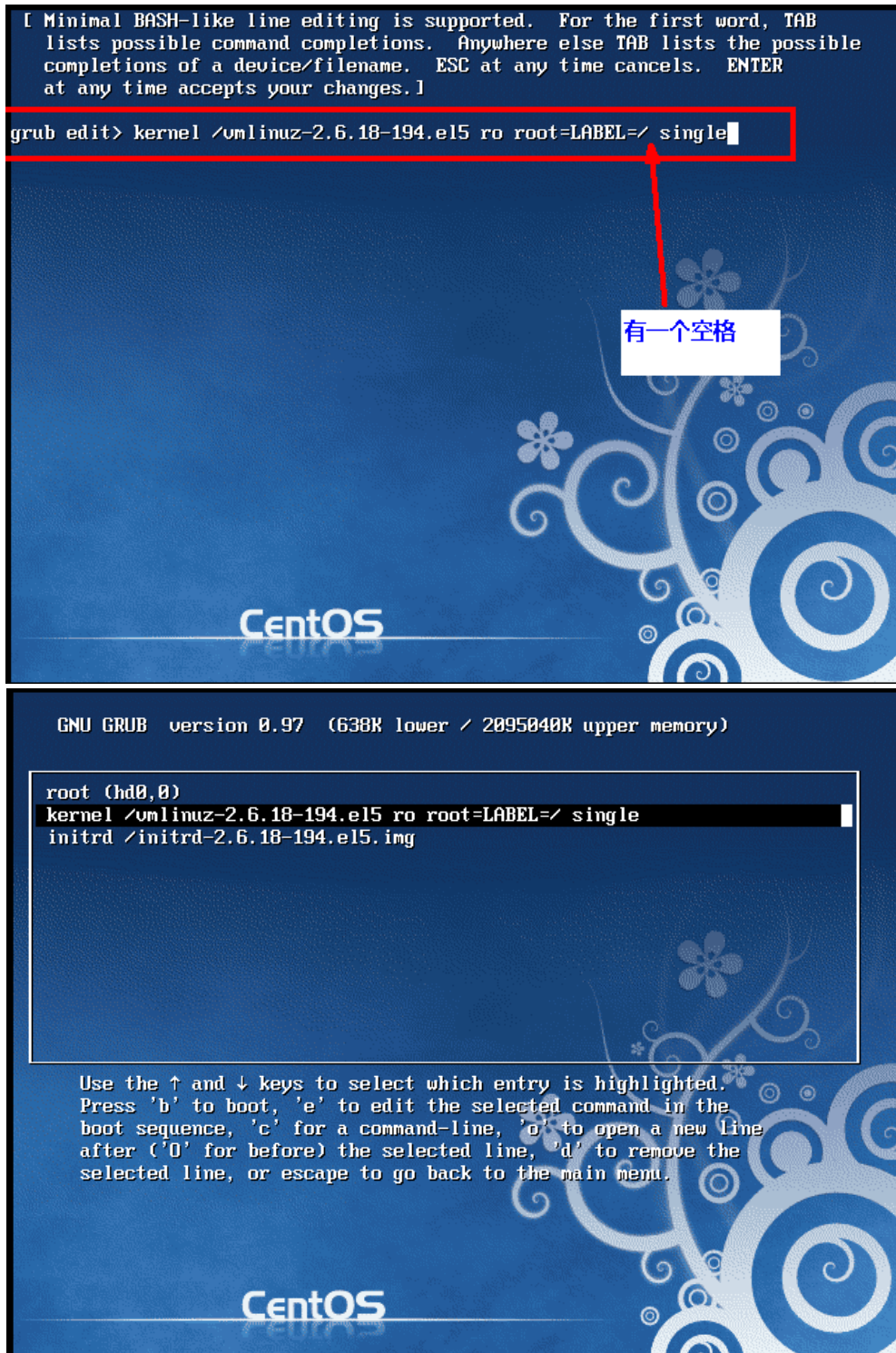
4) 移动到下一行，再次按 e 键



5) 移动到下一行，进行修改







修改完成后回车键，然后按 b 键进行重新启动进入系统

6) 移动到下一行，进行修改

```
no fstab.sys, mounting internal defaults
Switching to new root and running init.
unmounting old /dev
unmounting old /proc
unmounting old /sys
SELinux: Disabled at runtime.
type=1404 audit(1446774735.619:2): selinux=0 auid=4294967295 ses=4294967295
INIT: version 2.86 booting
        Welcome to CentOS release 5.5 (Final)
        Press 'I' to enter interactive startup.
Setting clock (utc): Fri Nov 6 09:52:17 CST 2015      [ OK ]
Starting udev:                                       [ OK ]
Loading default keymap (us):                         [ OK ]
Setting hostname atguigu.cloud:                      [ OK ]
No devices found
Setting up Logical Volume Management:                 [ OK ]
Checking filesystems
/: clean, 100006/3840192 files, 975672/3839535 blocks
/boot: clean, 35/26104 files, 14881/104388 blocks
Remounting root filesystem in read-write mode:       [ OK ]
Mounting local filesystems:                          [ OK ]
Enabling local filesystem quotas:                   [ OK ]
Enabling /etc/fstab swaps:                          [ OK ]
sh-3.2# _
```

```
sh-3.2# passwd
Changing password for user root.
New UNIX password:
BAD PASSWORD: it is too simplistic/systematic
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
sh-3.2# _
```

最终修改完密码，reboot 一下即可。

## 五、远程登录

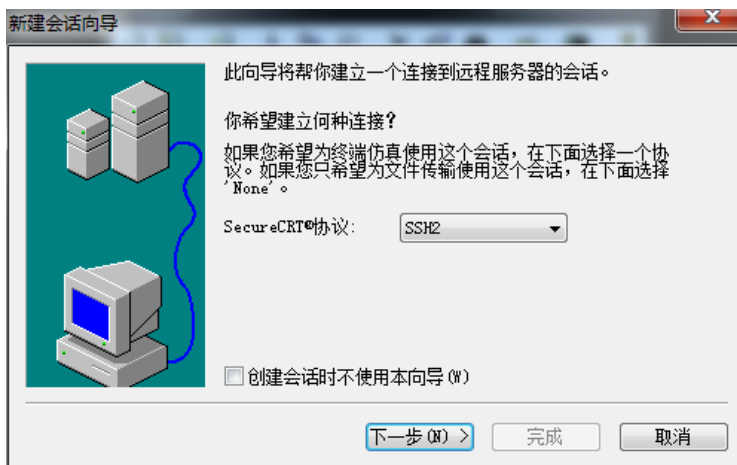
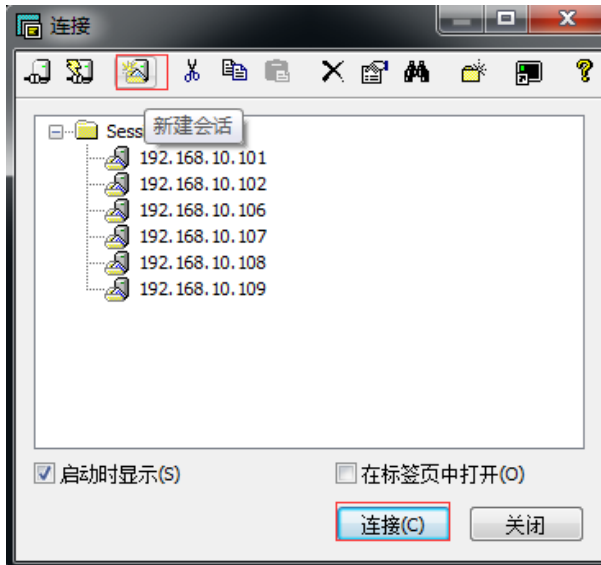
### 5.1 安装 SecureCRT

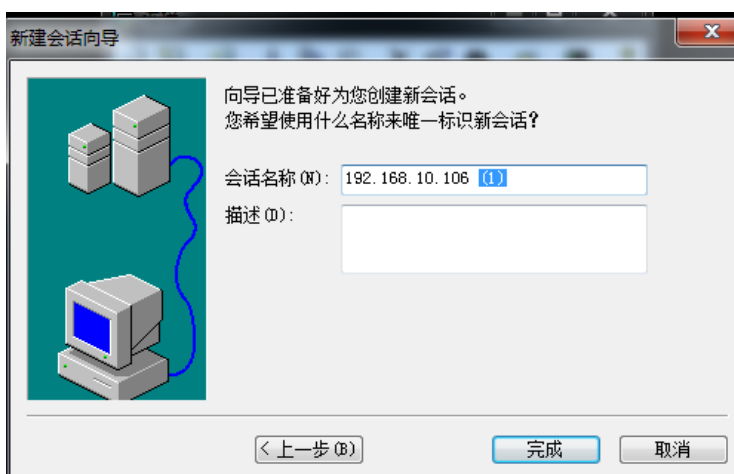
Linux 远程登录及相关工具介绍

Linux 一般作为服务器使用，而服务器一般放在机房，你不可能在机房操作你的 Linux 服务器。这时我们就需要远程登录到 Linux 服务器来管理维护系统。

Linux 系统中是通过 SSH 服务实现的远程登录功能，默认 ssh 服务端口号为 22。Window 系统上 Linux 远程登录客户端有 SecureCRT, Putty, SSH Secure Shell, XShell 等

#### 1) 安装步骤

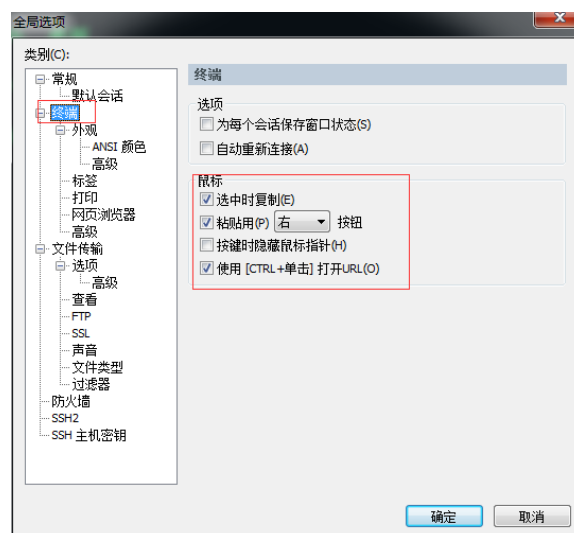
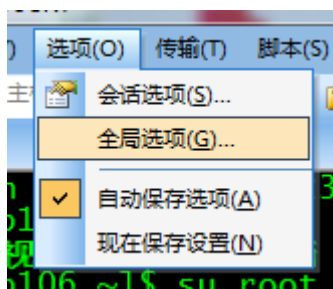




## 2) 操作

(1) 鼠标选中即为复制

(2) 鼠标右键即为粘贴



## 5.2 SecureCRT 中文乱码解决方法

1) 重新查看会话，是否中文显示正常

2) 依然无法正常显示中文，可能是由于 Linux 系统中默认的字符编码非 UTF8 所致

用 root 用户登录。输入 `cat /etc/sysconfig/i18n`

如果安装系统为中文系统，则修改【LANG="zh\_CN.UTF-8"】

如果安装系统为英文系统，则修改【LANG="en\_US.UTF-8"】

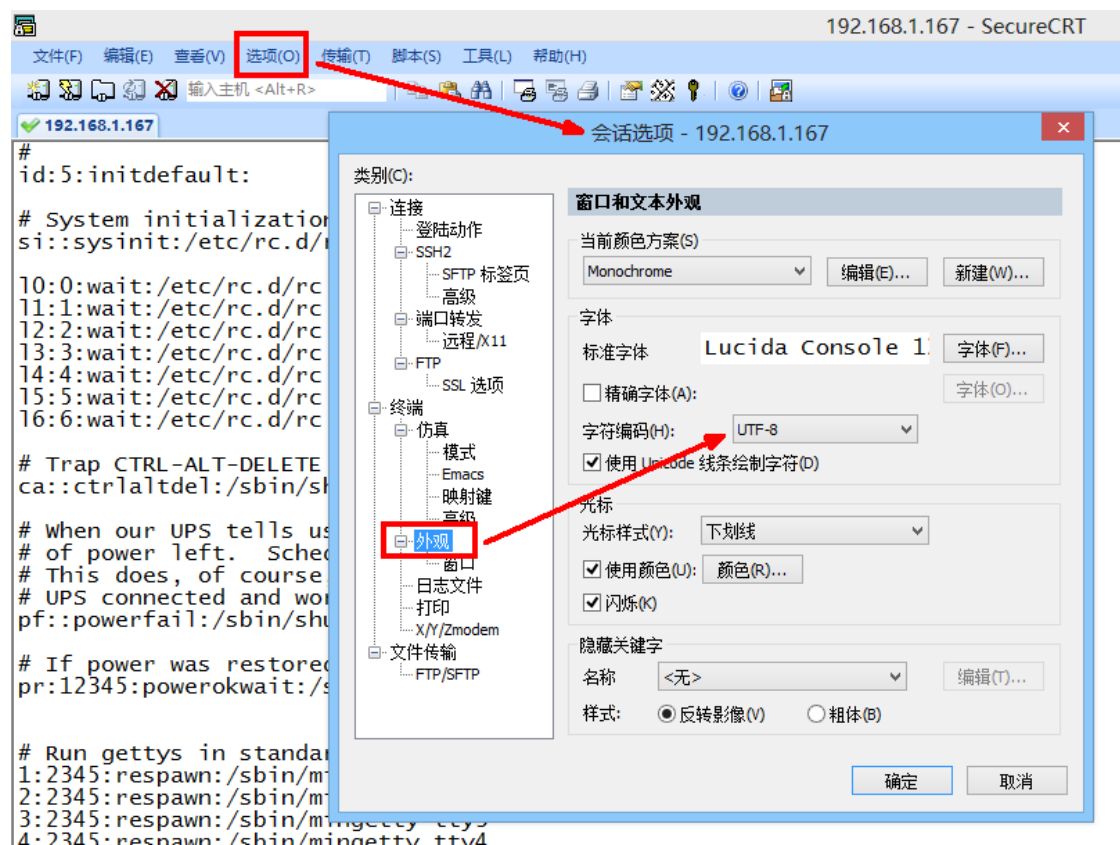
保存文件。断开 SSH，重新登录。就正常了

下面是修改后的查看

```
[root@hadoop100 ~]# cat /etc/sysconfig/i18n
```

```
LANG="zh_CN.UTF-8"
```

3) 调整设置 CRT 解决



## 六、用户和用户组管理命令

### 6.1 用户管理命令

#### 6.1.1 useradd 添加新用户

1) 基本语法:

useradd 用户名 (功能描述: 添加新用户)

2) 案例:

```
[root@hadoop101 opt]# user atguigu
```

#### 6.1.2 passwd 设置用户密码

1) 基本语法:

passwd 用户名 (功能描述: 设置用户密码)

2) 案例

```
[root@hadoop101 opt]# passwd atguigu
```

#### 6.1.3 id 判断用户是否存在

1) 基本语法:

id 用户名

2) 案例:

```
[root@hadoop101 opt]# id atguigu
```

#### 6.1.4 su 切换用户

1) 基本语法:

su 用户名称 (功能描述: 切换用户)

2) 案例

```
[root@hadoop101 opt]# su atguigu
```

#### 6.1.5 userdel 删除用户

1) 基本语法:

(1) userdel 用户名 (功能描述: 删除用户但保存用户主目录)

(2) userdel -r 用户名 (功能描述: 用户和用户主目录, 都删除)



## 2) 案例:

- (1) 删除用户但保存用户主目录

```
[root@hadoop101 opt]# userdel atguigu
```

- (2) 删除用户和用户主目录, 都删除

```
[root@hadoop101 opt]# userdel -r atguigu
```

## 6.1.6 who 查看登录用户信息

## 1) 基本语法

- (1) whoami (功能描述: 显示自身用户名称)

- (2) who am i (功能描述: 显示登录用户的用户名)

- (3) who (功能描述: 看当前有哪些用户登录到了本台机器上)

## 2) 案例

```
[root@hadoop101 opt]# whoami
```

```
[root@hadoop101 opt]# who am i
```

```
[root@hadoop101 opt]# who
```

## 6.1.7 设置 atguigu 普通用户具有 root 权限

## 1) 修改配置文件

修改 /etc/sudoers 文件, 找到下面一行, 在 root 下面添加一行, 如下所示:

```
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL
atguigu ALL=(ALL)    ALL
```

修改完毕, 现在可以用 atguigu 帐号登录, 然后用命令 `su -`, 即可获得 root 权限进行操作。

## 2) 案例

```
[atguigu@hadoop101 opt]$ sudo mkdir module
```

```
[root@hadoop101 opt]# chown atguigu:atguigu module/
```

## 6.1.8 usermod 修改用户 (讲完后面的用户组知识点后再回来讲解)

## 1) 基本语法:

```
usermod -g 用户组 用户名
```

## 2) 案例:

【更多 Java - Android 资料下载, 可访问尚硅谷 (中国) 官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

将用户atguigu加入dev用户组

```
[root@hadoop101 opt]#usermod -g dev atguigu
```

## 6.2 用户组管理命令（后面讲）

每个用户都有一个用户组，系统可以对一个用户组中的所有用户进行集中管理。不同Linux 系统对用户组的规定有所不同，

如Linux下的用户属于与它同名的用户组，这个用户组在创建用户时同时创建。

用户组的管理涉及用户组的添加、删除和修改。组的增加、删除和修改实际上就是对/etc/group文件的更新。

### 6.2.1 groupadd 新增组

1) 基本语法

```
groupadd 组名
```

2) 案例:

添加一个atguigu组

```
[root@hadoop101 opt]#groupadd atguigu
```

### 6.2.2 groupdel 删除组

1) 基本语法:

```
groupdel 组名
```

2) 案例

```
[root@hadoop101 opt]# groupdel atguigu
```

### 6.2.3 groupmod 修改组

1) 基本语法:

```
groupmod -n 新组名 老组名
```

2) 案例

修改atguigu组名称为atguigu1

```
[root@hadoop101 atguigu]# groupmod -n atguigu1 atguigu
```

### 6.2.4 综合案例

```
[root@hadoop101 atguigu]# groupadd dev
```

```
[root@hadoop101 atguigu]# groupmod -n device dev
[root@hadoop101 atguigu]# usermod -g device atguigu
[root@hadoop101 atguigu]# su atguigu
[atguigu@hadoop101 ~]$ mkdir atguigu
[atguigu@hadoop101 ~]$ ls -l
drwxr-xr-x. 2 atguigu device 4096 5月 27 16:31 atguigu
[root@hadoop101 atguigu]# usermod -g atguigu atguigu
```

## 七、常用基本命令

### 7.1 帮助命令

#### 7.1.1 man 获得帮助信息

1) 基本语法:

man [命令或配置文件] (功能描述: 获得帮助信息)

(1) 显示说明

```
[ybird@www ~]$ man date
DATE(1) User Commands DATE(1)
# 请注意上面这个括号内的数字
NAME <==这个指令的完整全名, 如下所示为 date 且说明简单用途为设定与显示日期/时间
date - print or set the system date and time

SYNOPSIS <==这个指令的基本语法如下所示
date [OPTION]... [+FORMAT]
date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

DESCRIPTION <==详细说明刚刚语法谈到的选项与参数的用法
Display the current time in the given FORMAT, or set the system date.
```

NAME 命令的名称和单行描述

SYNOPSIS 怎样使用命令

DESCRIPTION 命令功能的深入讨论

EXAMPLES 怎样使用命令的例子

SEE ALSO 相关主题 (通常是手册页)

(2) 数字说明 q

- 1.用户在 shell 环境中可以操作的命令或是可执行的文件
- 2.系统内核(kernel)可以调用的函数
- 3.常用的函数 or 函数库
- 4.设备配置文件
- 5.配置文件的格式
- 6.游戏相关
- 7.linux 网络协议和文件系统
- 8.系统管理员可以用的命令
- 9.跟内核有关系的文件

## 2) 案例

```
[root@hadoop106 home]# man ls
```

### 7.1.2 help 获得 shell 内置命令的帮助信息

#### 1) 基本语法:

help 命令      (功能描述: 获得 shell 内置命令的帮助信息)

#### 2) 案例:

```
[root@hadoop101 bin]# help cd
```

### 7.1.3 常用快捷键

- 1) ctrl + c: 停止进程
- 2) ctrl+l: 清屏
- 3) ctrl + q: 退出
- 4) 善于用 tab 键
- 5) 上下键: 查找执行过的命令
- 6) ctrl +alt: linux 和 Windows 之间切换

## 7.2 文件目录类

### 7.2.1 pwd 显示当前工作目录的绝对路径

#### 1) 基本语法:

pwd      (功能描述: 显示当前工作目录的绝对路径)

#### 2) 案例

```
[root@hadoop106 home]# pwd
```

```
/home
```

## 7.2.2 ls 列出目录的内容

### 1) 基本语法:

ls [选项] [目录或是文件]

选项:

-a : 全部的文件, 连同隐藏档( 开头为 . 的文件) 一起列出来(常用)

-d : 仅列出目录本身, 而不是列出目录内的文件数据(常用)

-l : 长数据串列出, 包含文件的属性与权限等等数据; (常用)

每行列出的信息依次是: 文件类型与权限 链接数 文件属主 文件属组 文件大小用

byte来表示 建立或最近修改的时间 名字

### 2) 案例

```
[atguigu@hadoop101 ~]$ ls -al
总用量 44
drwx-----. 5 atguigu atguigu 4096 5月 27 15:15 .
drwxr-xr-x. 3 root    root    4096 5月 27 14:03 ..
drwxrwxrwx. 2 root    root    4096 5月 27 14:14 hello
-rwxrw-r--. 1 atguigu atguigu  34 5月 27 14:20 test.txt
```

## 7.2.3 mkdir 创建一个新的目录

### 1) 基本语法:

mkdir [-p] 要创建的目录

选项:

-p: 创建多层目录

### 2) 案例

```
[root@hadoop106 opt]# mkdir test
[root@hadoop106 opt]# mkdir -p user/atguigu
```

## 7.2.4 rmdir 删除一个空的目录

### 1) 基本语法:

rmdir 要删除的空目录

### 2) 案例

【更多 Java – Android 资料下载, 可访问尚硅谷(中国)官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

```
[root@hadoop106 opt]# mkdir test
```

```
[root@hadoop106 opt]# rmdir test
```

## 7.2.5 touch 创建空文件

### 1) 基本语法:

touch 文件名称

### 2) 案例

```
[root@hadoop106 opt]# touch test.java
```

## 7.2.6 cd 切换目录

### 1) 基本语法:

(1) cd 绝对路径

(2) cd 相对路径

(3) cd ~或者 cd (功能描述: 回到自己的家目录)

(4) cd - (功能描述: 回到上一次所在目录)

(5) cd .. (功能描述: 回到当前目录的上一级目录)

(6) cd -P (功能描述: 跳转到实际物理路径, 而非快捷方式路径)

### 2) 案例

(1) 使用 mkdir 命令创建atguigu目录

```
[root@www ~]# mkdir atguigu
```

(2) 使用绝对路径切换到atguigu目录

```
[root@www ~]# cd /root/atguigu/
```

(3) 使用相对路径切换到atguigu目录

```
[root@www ~]# cd ./atguigu/
```

(4) 表示回到自己的家目录, 亦即是 /root 这个目录

```
[root@www atguigu]# cd ~
```

(5) cd- 回到上一次所在目录

```
[root@www atguigu]# cd -
```

(6) 表示回到当前目录的上一级目录, 亦即是 /root 的上一级目录的意思;

```
[root@www ~]# cd ..
```

## 7.2.7 cp 复制文件或目录

### 1) 基本语法:

(1) `cp source dest` (功能描述: 复制source文件到dest)

(2) `cp -r sourceFolder targetFolder` (功能描述: 递归复制整个文件夹)

### 2) 案例

#### (1) 复制文件

```
[root@hadoop106 opt]# cp test.java test
```

#### (2) 递归复制整个文件夹

```
[root@hadoop106 opt]# cp -r test test1
```

## 7.2.8 rm 移除文件或目录

### 1) 基本语法

(1) `rmdir deleteEmptyFolder` (功能描述: 删除空目录)

(2) `rm -rf deleteFile` (功能描述: 递归删除目录中所有内容)

### 2) 案例

#### 1) 删除空目录

```
[root@hadoop106 opt]# rmdir test
```

#### 2) 递归删除目录中所有内容

```
[root@hadoop106 opt]# rm -rf test1
```

## 7.2.9 mv 移动文件与目录或重命名

### 1) 基本语法:

(1) `mv oldNameFile newNameFile` (功能描述: 重命名)

(2) `mv /temp/movefile /targetFolder` (功能描述: 移动文件)

### 2) 案例:

#### 1) 重命名

```
[root@hadoop106 opt]# mv test.java test1.java
```

#### 2) 移动文件

```
[root@hadoop106 opt]# mv test1.java test1
```

## 7.2.10 cat 查看文件内容

查看文件内容，从第一行开始显示。

### 1) 基本语法

cat [选项] 要查看的文件

选项：

-A : 相当于 -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

### 2) 案例

```
[atguigu@hadoop101 ~]$ cat -A test.txt
```

```
hellda $
```

```
dasadf ^I$
```

```
da^I^I^I$
```

```
das$
```

## 7.2.11 tac 查看文件内容

查看文件内容，从最后一行开始显示，可以看出 tac 是 cat 的倒著写。

### 1) 基本语法：

cat [选项参数] 要查看的文件

### 2) 案例

```
[root@hadoop106 test1]# cat test1.java
```

```
hello
```

```
atguigu
```

```
atguigu1
```

```
[root@hadoop106 test1]# tac test1.java
```



```
atguigu1
```

```
atguigu
```

```
hello
```

### 7.2.12 more 查看文件内容

查看文件内容，一页一页的显示文件内容。

#### 1) 基本语法:

```
more 要查看的文件
```

#### 2) 功能使用说明

空白键 (space): 代表向下翻一页;

Enter:代表向下翻『一行』;

q:代表立刻离开 more , 不再显示该文件内容。

Ctrl+F 向下滚动一屏

Ctrl+B 返回上一屏

= 输出当前行的行号

:f 输出文件名和当前行的行号

#### 3) 案例

```
[root@hadoop106 test1]# more test1.java
```

### 7.2.13 less 查看文件内容

less 的作用与 more 十分相似，都可以用来浏览文字档案的内容，不同的是 less 允许使用[pageup] [pagedown]来回滚动。

#### 1) 基本语法:

```
less 要查看的文件
```

#### 2) 功能使用说明

空白键 : 向下翻动一页;

[pagedown]: 向下翻动一页;

[pageup] : 向上翻动一页;

/字符串 : 向下搜寻『字符串』的功能;

?字符串 : 向上搜寻『字符串』的功能;

q : 离开 less 这个程序;

### 3) 案例

```
[root@hadoop106 test1]# less test1.java
```

## 7.2.14 head 查看文件内容

查看文件内容，只看头几行。

### 1) 基本语法

head -n 10 文件 (功能描述: 查看文件头10行内容, 10可以是任意行数)

### 2) 案例

```
[root@hadoop106 test1]# head -n 2 test1.java
```

```
hello
```

```
atguigu
```

## 7.2.15 tail 查看文件内容

查看文件内容，只看尾巴几行。

### 1) 基本语法

(1) tail -n 10 文件 (功能描述: 查看文件头10行内容, 10可以是任意行数)

(2) tail -f 文件 (功能描述: 实时追踪该文档的所有更新)

### 2) 案例

(1) 查看文件头1行内容

```
[root@hadoop106 test1]# tail -n 1 test1.java
```

```
Atguigu
```

(2) 实时追踪该档的所有更新

```
[root@hadoop106 test1]# tail -f test1.java
```

```
hello
```

```
atguigu
```

```
atguigu
```

## 7.2.16 重定向命令

### 1) 基本语法:

(1) ls -l >文件 (功能描述: 列表的内容写入文件 a.txt 中(覆盖写))

【更多 Java - Android 资料下载, 可访问尚硅谷(中国)官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

(2) `ls -al >> 文件` (功能描述: 列表的内容追加到文件 `aa.txt` 的末尾)

## 2) 案例

(1) `[root@hadoop101 opt]# ls -l>t.txt`

(2) `[root@hadoop101 opt]# ls -l>>t.txt`

(3) `[root@hadoop106 test1]# echo hello>>test1.java`

## 7.2.17 echo

### 1) 基本语法:

(1) `echo 要显示的内容 >> 存储内容的文件` (功能描述: 将要显示的内容, 存储到文件中)

(2) `echo 变量` (功能描述: 显示变量的值)

### 2) 案例

`[root@hadoop106 test1]# echo $JAVA_HOME`

`/opt/module/jdk1.7.0_79`

## 7.2.18 ln 软链接

### 1) 基本语法:

`ln -s [原文件] [目标文件]` (功能描述: 给原文件创建一个软链接, 软链接存放在目标文件目录)

### 2) 案例:

`[root@hadoop101 module]# ln -s /opt/module/test.txt /opt/t.txt`

`[root@hadoop101 opt]# ll`

`lrwxrwxrwx. 1 root root 20 6 月 17 12:56 t.txt -> /opt/module/test.txt`

创建一个软链接

`[atguigu@hadoop103 opt]$ ln -s /opt/module/hadoop-2.7.2/ /opt/software/hadoop`

`cd` 不加参数进入是软链接的地址

`[atguigu@hadoop103 software]$ cd hadoop`

`[atguigu@hadoop103 hadoop]$ pwd`

`/opt/software/hadoop`

cd 加参数进入是实际的物理地址

```
[atguigu@hadoop103 software]$ cd -P hadoop
```

```
[atguigu@hadoop103 hadoop-2.7.2]$ pwd
```

```
/opt/module/hadoop-2.7.2
```

## 7.2.19 history 查看所敲命令历史

1) 基本语法:

```
history
```

2) 案例

```
[root@hadoop106 test1]# history
```

## 7.3 时间日期类

1) 基本语法

```
date [OPTION]... [+FORMAT]
```

### 7.3.1 date 设置系统时间

1) 基本语法:

```
date -s 字符串时间
```

2) 案例

```
[root@hadoop106 /]# date -s "2017-06-19 20:52:18"
```

### 7.3.2 date 显示当前时间

1) 基本语法:

- |  |                     |
|--|---------------------|
| (1) date                               | (功能描述: 显示当前时间)      |
| (2) date +%Y                           | (功能描述: 显示当前年份)      |
| (3) date +%m                           | (功能描述: 显示当前月份)      |
| (4) date +%d                           | (功能描述: 显示当前是哪一天)    |
| (5) date +%Y%m%d    date +%Y/%m/%d ... | (功能描述: 显示当前年月日各种格式) |
| (6) date "+%Y-%m-%d %H:%M:%S"          | (功能描述: 显示年月日时分秒)    |

2) 案例

```
[root@hadoop106 /]# date
```

【更多 Java – Android 资料下载, 可访问尚硅谷(中国)官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

2017 年 06 月 19 日 星期一 20:53:30 CST

```
[root@hadoop106 /]# date +%Y%m%d
```

20170619

```
[root@hadoop106 /]# date "+%Y-%m-%d %H:%M:%S"
```

2017-06-19 20:54:58

### 7.3.3 date 显示非当前时间

#### 1) 基本语法:

- (1) `date -d '1 days ago'` (功能描述: 显示前一天日期)
- (2) `date -d yesterday +%Y%m%d` (同上)
- (3) `date -d next-day +%Y%m%d` (功能描述: 显示明天日期)
- (4) `date -d 'next monday'` (功能描述: 显示下周一时间)

#### 2) 案例:

```
[root@hadoop106 /]# date -d '1 days ago'
```

2017 年 06 月 18 日 星期日 21:07:22 CST

```
[root@hadoop106 /]# date -d next-day +%Y%m%d
```

20170620

```
[root@hadoop106 /]# date -d 'next monday'
```

2017 年 06 月 26 日 星期一 00:00:00 CST

### 7.3.4 cal 查看日历

#### 1) 基本语法:

`cal [选项]` (功能描述: 不加选项, 显示本月日历)

选项:

-3, 显示系统前一个月, 当前月, 下一个月的日历

具体某一年, 显示这一年的日历。

#### 2) 案例:

```
[root@hadoop106 /]# cal
```

```
[root@hadoop106 /]# cal -3
```

```
[root@hadoop106 /]# cal 2016
```

## 7.4 文件权限类

### 7.4.1 文件属性

Linux系统是一种典型的多用户系统，不同的用户处于不同的地位，拥有不同的权限。为了保护系统的安全性，Linux系统对不同的用户访问同一文件（包括目录文件）的权限做了不同的规定。在Linux中我们可以使用ll或者ls -l命令来显示一个文件的属性以及文件所属的用户和组。

1) 从左到右的10个字符表示：

如果没有权限，就会出现减号[-]而已。从左至右用0-9这些数字来表示：

(1) 0 首位表示类型

在Linux中第一个字符代表这个文件是目录、文件或链接文件等等

- 代表文件

d 代表目录

c 字符流，装置文件里面的串行端口设备，例如键盘、鼠标(一次性读取装置)

s socket

p 管道

l 链接文档(link file);

b 设备文件，装置文件里面的可供储存的接口设备(可随机存取装置)

(2) 第1-3位确定属主（该文件的所有者）拥有该文件的权限。---User

(3) 第4-6位确定属组（所有者的同组用户）拥有该文件的权限，---Group

(4) 第7-9位确定其他用户拥有该文件的权限 ---Other

文件类型	属主权限	属组权限	其他用户权限
0	1 2 3	4 5 6	7 8 9
d	R w x	R - x	R - x
目录文件	读 写 执行	读 写 执行	读 写 执行

2) rxw作用文件和目录的不同解释

(1) 作用到文件：

[r]代表可读(read): 可以读取，查看

[w]代表可写(write): 可以修改，但是不代表可以删除该文件,删除一个文件的前提

条件是对该文件所在的目录有写权限，才能删除该文件。

[ x ]代表可执行(execute):可以被系统执行

(2) 作用到目录:

[ r ]代表可读(read): 可以读取，ls查看目录内容

[ w ]代表可写(write): 可以修改，目录内创建+删除+重命名目录

[ x ]代表可执行(execute):可以进入该目录

3) 案例

```
[atguigu@hadoop101 ~]$ ls -l
```

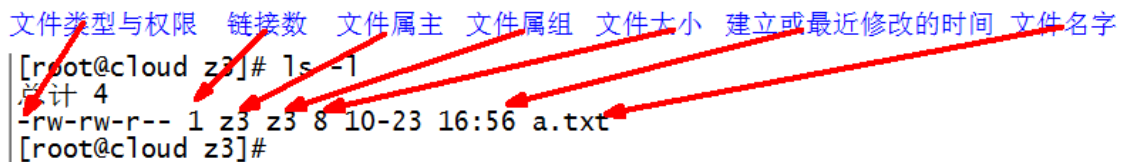
总用量 8

```
drwxrwxr-x. 2 atguigu atguigu 4096 5月 27 14:14 hello
```

```
-rw-rw-r--. 1 atguigu atguigu 34 5月 27 14:20 test.txt
```

文件类型与权限 链接数 文件属主 文件属组 文件大小 建立或最近修改的时间 文件名字

```
[root@cloud z3]# ls -l
总计 4
-rw-rw-r-- 1 z3 z3 8 10-23 16:56 a.txt
[root@cloud z3]#
```



## 7.4.2 chmod 改变权限

1) 基本语法:

```
chmod [{ugoa}]{+|=}{rwx} [文件或目录] [mode=421] [文件或目录]
```

2) 功能描述

改变文件或者目录权限

文件: r-查看; w-修改; x-执行文件

目录: r-列出目录内容; w-在目录中创建和删除; x-进入目录

删除一个文件的前提条件:该文件所在的目录有写权限，你才能删除该文件。

3) 案例

```
[root@hadoop106 test1]# chmod u+x test1.java
```

```
[root@hadoop106 test1]# chmod g+x test1.java
```

```
[root@hadoop106 test1]# chmod o+x test1.java
```

```
[root@hadoop106 test1]# chmod 777 test1.java
```

```
[root@hadoop106 test1]# chmod -R 777 testdir
```



### 7.4.3 chgrp 改变所属组

1) 基本语法:

chgrp [最终用户组] [文件或目录]      (功能描述: 改变文件或者目录的所属组)

2) 案例

```
[root@hadoop106 test1]# chgrp atguigu test1.java
```

```
[root@hadoop106 test1]# ls -al
```

```
-rwxr-xr-x. 1 root atguigu  551 5月  23 13:02 test1.java
```

### 7.4.4 chown 改变所有者

1) 基本语法:

chown [最终用户] [文件或目录]      (功能描述: 改变文件或者目录的所有者)

2) 案例

```
[root@hadoop106 test1]# chown atguigu test1.java
```

```
[root@hadoop106 test1]# ls -al
```

```
-rwxr-xr-x. 1 atguigu atguigu  551 5月  23 13:02 test1.java
```

### 7.4.5 su 切换用户

1) 基本语法:

su -username      (功能描述: 切换用户)

2) 案例

```
[root@hadoop101 atguigu]# su atguigu
```

```
[atguigu@hadoop101 ~]$
```

```
[atguigu@hadoop101 ~]$ su root
```

密码:

```
[root@hadoop101 atguigu]#
```

## 7.5 磁盘分区类

### 7.5.1 fdisk 查看分区

1) 基本语法:

`fdisk -l` (功能描述: 查看磁盘分区详情)

**注意:** 在 `root` 用户下才能使用

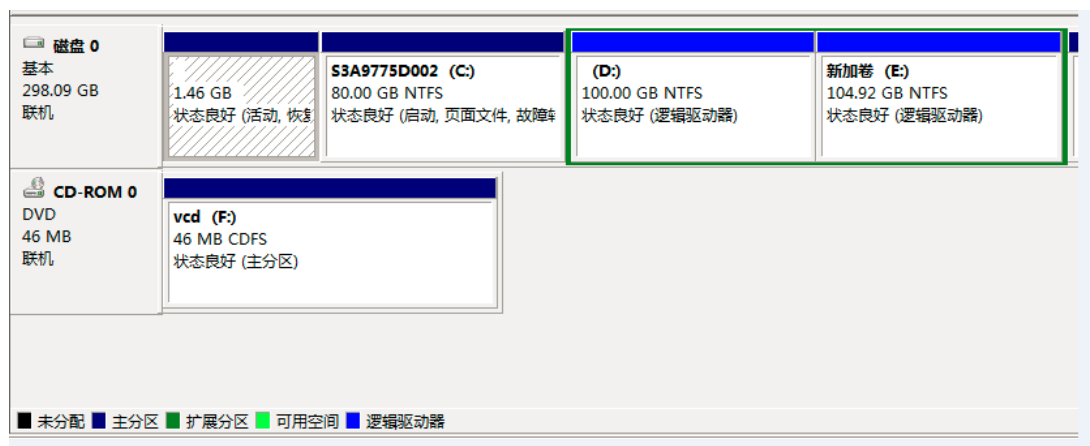
2) 功能说明:

(1) Linux 分区

这个硬盘是20G的, 有255个磁面; 63个扇区; 2610个磁柱; 每个 cylinder (磁柱) 的容量是  $8225280 \text{ bytes} = 8225.280 \text{ K}$  (约为)  $= 8.225280 \text{ M}$  (约为);

Device	Boot	Start	End	Blocks	Id	System
分区序列	引导	从X磁柱开始	到Y磁柱结束	容量	分区类型ID	分区类型

(2) Win7分区



3) 案例

```
[root@hadoop101 ~]# fdisk -l
```

```
Disk /dev/sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0005e654
```

```
Device Boot      Start      End      Blocks  Id  System
/dev/sda1  *            1         26      204800   83   Linux
```

Partition 1 does not end on cylinder boundary.

【更多 Java – Android 资料下载, 可访问尚硅谷 (中国) 官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

/dev/sda2	26	1332	10485760	83	Linux
/dev/sda3	1332	1593	2097152	82	Linux swap / Solaris

## 7.5.2 df 查看硬盘

1) 基本语法:

**df 参数** (功能描述: 列出文件系统的整体磁盘使用量, 检查文件系统的磁盘空间占用情况)

参数:

- a : 列出所有的文件系统, 包括系统特有的 /proc 等文件系统;
- k : 以 KBytes 的容量显示各文件系统;
- m : 以 MBytes 的容量显示各文件系统;
- h : 以人们较易阅读的 GBytes, MBytes, KBytes 等格式自行显示;
- H : 以 M=1000K 取代 M=1024K 的进位方式;
- T : 显示文件系统类型, 连同该 partition 的 filesystem 名称 (例如 ext3) 也列出;
- i : 不用硬盘容量, 而以 inode 的数量来显示

2) 案例

```
[root@hadoop106 ~]# df -h
```

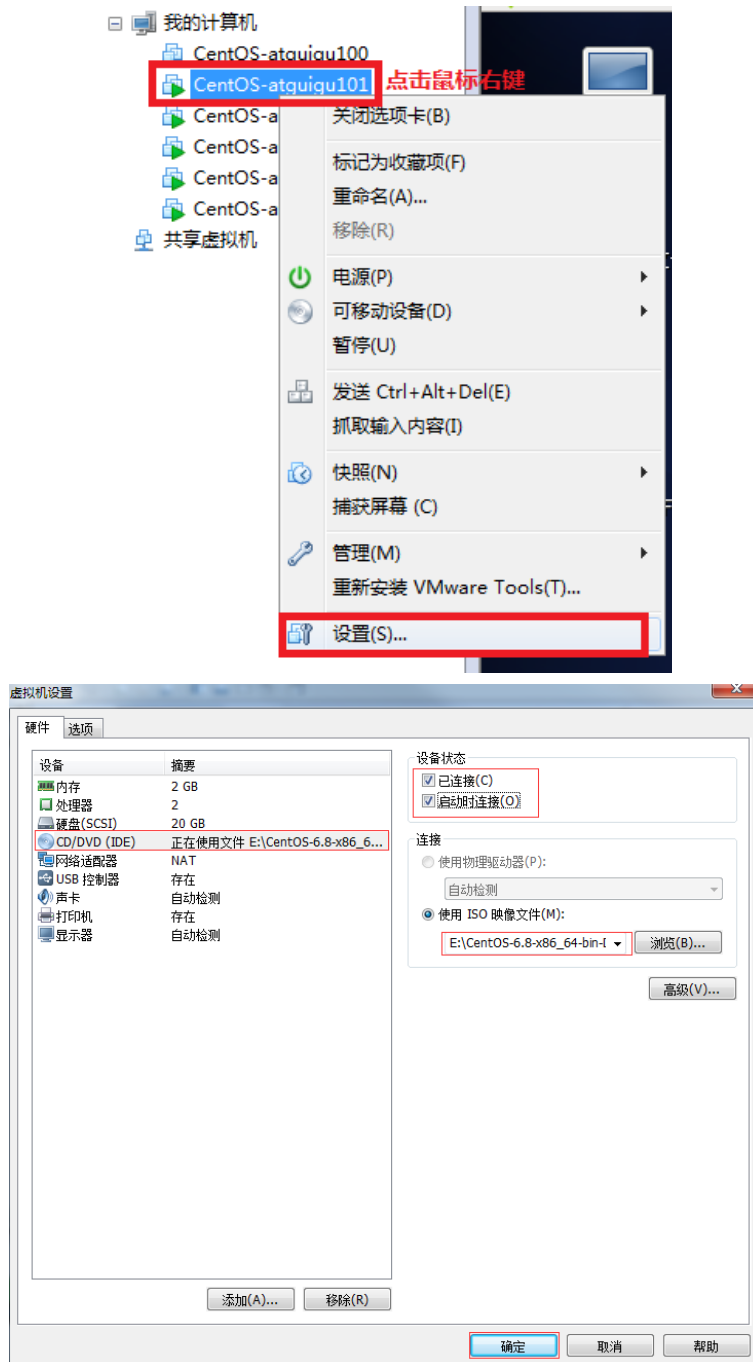
Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda2	15G	3.5G	11G	26%	/
tmpfs	939M	224K	939M	1%	/dev/shm
/dev/sda1	190M	39M	142M	22%	/boot

## 7.5.3 mount/umount 挂载/卸载

对于Linux用户来讲, 不论有几个分区, 分别分给哪一个目录使用, 它总归就是一个根目录、一个独立且唯一的文件结构

Linux中每个分区都是用来组成整个文件系统的一部分, 她在用一种叫做“挂载”的处理方法, 它整个文件系统中包含了一整套的文件和目录, 并将一个分区和一个目录联系起来, 要载入的那个分区将使它的存储空间在这个目录下获得。

0) 挂载前准备 (必须要有光盘或者已经连接镜像文件)



### 1) 挂载光盘语法:

`mount [-t vfstype] [-o options] device dir`

(1) `-t vfstype` 指定文件系统的类型，通常不必指定。`mount` 会自动选择正确的类型。

常用类型有:

光盘或光盘镜像: `iso9660`

DOS fat16文件系统: `msdos`

Windows 9x fat32文件系统: `vfat`

【更多 Java – Android 资料下载，可访问尚硅谷（中国）官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

Windows NT ntfs文件系统：ntfs

Mount Windows文件网络共享：smbfs

UNIX(LINUX) 文件网络共享：nfs

(2) -o options 主要用来描述设备或档案的挂接方式。常用的参数有：

loop: 用来把一个文件当成硬盘分区挂接上系统

ro: 采用只读方式挂接设备

rw: 采用读写方式挂接设备

iocharset: 指定访问文件系统所用字符集

(3) device 要挂接(mount)的设备

(4) dir设备在系统上的挂接点(mount point)

## 2) 案例

(1) 光盘镜像文件的挂载

```
[root@localhost ~]# mkdir /mnt/cdrom/
```

建立挂载点

```
[root@localhost ~]# mount -t iso9660 /dev/cdrom /mnt/cdrom/
```

设备/dev/cdrom挂载

到 挂载点： /mnt/cdrom中

```
[root@hadoop101 ~]# ll /mnt/cdrom/
```

## 3) 卸载光盘语法：

```
[root@localhost ~]# umount 设备文件名或挂载点
```

## 4) 案例

```
[root@localhost ~]# umount /mnt/cdrom
```

## 5) 开机自动挂载语法：

```
[root@hadoop101 ~]# vi /etc/fstab
```

添加红框中内容，保存退出。

```
#  
# /etc/fstab  
# Created by anaconda on Sat May 27 18:55:56 2017  
#  
# Accessible filesystems, by reference, are maintained under '/dev/disk'  
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info  
#  
UUID=b687374c-43b9-4b40-b9bf-26d3e77e33c2 / ext4 default  
ts 1 1  
UUID=f9b160f5-3ef1-4051-ae14-9a17aed2760c /boot ext4 default  
ts 1 2  
UUID=6f3b8eec-a932-474a-b7df-cbbfaa2df815 swap swap default  
ts 0 0  
tmpfs /dev/shm tmpfs defaults 0 0  
devpts /dev/pts devpts gid=5,mode=620 0 0  
sysfs /sys sysfs defaults 0 0  
proc /proc proc defaults 0 0  
/dev/cdrom /mnt/cdrom iso9660 defaults 0 0
```

## 7.6 搜索查找类

### 7.6.1 find 查找文件或者目录

1) 基本语法:

```
find [搜索范围] [匹配条件]
```

2) 案例

(1) 按文件名: 根据名称查找/目录下的filename.txt文件。

```
[root@hadoop106 ~]# find /opt/ -name '*.txt'
```

(2) 按拥有者: 查找/opt目录下, 用户名称为-user的文件

```
[root@hadoop106 ~]# find /opt/ -user atguigu
```

(3) 按文件大小: 在/home目录下查找大于200m的文件 (+n 大于 -n小于 n等于)

```
[root@hadoop106 ~]# find /home -size +204800
```

### 7.6.2 grep 在文件内搜索字符串匹配的行并输出

1) 基本语法

```
grep+参数+查找内容+源文件
```

参数:

-c: 只输出匹配行的计数。

-I: 不区分大小写(只适用于单字符)。

-h: 查询多文件时不显示文件名。

-l: 查询多文件时只输出包含匹配字符的文件名。

- n: 显示匹配行及行号。
- s: 不显示不存在或无匹配文本的错误信息。
- v: 显示不包含匹配文本的所有行。

## 2) 案例

```
[root@hadoop106 opt]# ls | grep -n test  
4:test1  
5:test2
```

## 7.6.3 which 文件搜索命令

### 1) 基本语法:

which 命令 (功能描述: 搜索命令所在目录及别名信息)

### 2) 案例

```
[root@hadoop101 opt]# which ls  
/bin/ls
```

## 7.7 进程线程类

进程是正在执行的一个程序或命令, 每一个进程都是一个运行的实体, 都有自己的地址空间, 并占用一定的系统资源。

### 7.7.1 ps 查看系统中所有进程

#### 1) 基本语法:

ps -aux (功能描述: 查看系统中所有进程)

#### 2) 功能说明

USER: 该进程是由哪个用户产生的

PID: 进程的 ID 号

%CPU: 该进程占用 CPU 资源的百分比, 占用越高, 进程越耗费资源;

%MEM: 该进程占用物理内存的百分比, 占用越高, 进程越耗费资源;

VSZ: 该进程占用虚拟内存的大小, 单位 KB;

RSS: 该进程占用实际物理内存的大小, 单位 KB;

TTY: 该进程是在哪个终端中运行的。其中 tty1-tty7 代表本地控制台终端, tty1-tty6 是本地的字符界面终端, tty7 是图形终端。pts/0-255 代表虚拟终端。



STAT: 进程状态。常见的状态有: R: 运行、S: 睡眠、T: 停止状态、s: 包含子进程、

+: 位于后台

START: 该进程的启动时间

TIME: 该进程占用 CPU 的运算时间, 注意不是系统时间

COMMAND: 产生此进程的命令名

### 3) 案例

```
[root@hadoop102 datas]# ps -aux
```

```
Warning: bad syntax, perhaps a bogus '-'? See /usr/share/doc/procps-3.2.8/FAQ
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  19348  1544 ?        Ss   10:55   0:01 /sbin/init
root         2  0.0  0.0      0     0 ?        S    10:55   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    10:55   0:00 [migration/0]
root         4  0.0  0.0      0     0 ?        S    10:55   0:00 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S    10:55   0:00 [stopper/0]
```

## 7.7.2 top 查看系统健康状态

### 1) 基本命令

top [选项]

#### (1) 选项:

-d 秒数: 指定 top 命令每隔几秒更新。默认是 3 秒在 top 命令的交互模式当中可以

执行的命令:

-i: 使 top 不显示任何闲置或者僵死进程。

-p: 通过指定监控进程 ID 来仅仅监控某个进程的状态。

-s: 使 top 命令在安全模式中运行。这将去除交互命令所带来的潜在危险。

#### (2) 操作选项:

P: 以 CPU 使用率排序, 默认就是此项

M: 以内存的使用率排序

N: 以 PID 排序

q: 退出 top

#### (3) 查询结果字段解释

第一行信息为任务队列信息

内容	说明
12:26:46	系统当前时间
up 1 day, 13:32	系统的运行时间, 本机已经运行 1 天

	13 小时 32 分钟
2 users	当前登录了两个用户
load average: 0.00, 0.00, 0.00	系统在之前 1 分钟, 5 分钟, 15 分钟的平均负载。一般认为小于 1 时, 负载较小。如果大于 1, 系统已经超出负荷。

第二行为进程信息

Tasks: 95 total	系统中的进程总数
1 running	正在运行的进程数
94 sleeping	睡眠的进程
0 stopped	正在停止的进程
0 zombie	僵尸进程。如果不是 0, 需要手工检查僵尸进程

第三行为 CPU 信息

Cpu(s): 0.1%us	用户模式占用的 CPU 百分比
0.1%sy	系统模式占用的 CPU 百分比
0.0%ni	改变过优先级的用户进程占用的 CPU 百分比
99.7%id	空闲 CPU 的 CPU 百分比
0.1%wa	等待输入/输出的进程的占用 CPU 百分比
0.0%hi	硬中断请求服务占用的 CPU 百分比
0.1%si	软中断请求服务占用的 CPU 百分比
0.0%st	st (Steal time) 虚拟时间百分比。就是当有虚拟机时, 虚拟 CPU 等待实际 CPU 的时间百分比。

第四行为物理内存信息

Mem: 625344k total	物理内存的总量, 单位 KB
571504k used	已经使用的物理内存数量
53840k free	空闲的物理内存数量, 我们使用的是虚拟机, 总共只分配了 628MB 内存, 所以只有 53MB 的空闲内存了
65800k buffers	作为缓冲的内存数量

第五行为交换分区 (swap) 信息

Swap: 524280k total	交换分区 (虚拟内存) 的总大小
0k used	已经使用的交互分区的大小
524280k free	空闲交换分区的大小
409280k cached	作为缓存的交互分区的大小

2) 案例

```
[root@hadoop101 atguigu]# top -d 1
[root@hadoop101 atguigu]# top -i
[root@hadoop101 atguigu]# top -p 2575
[root@hadoop101 atguigu]# top -s
```

【更多 Java - Android 资料下载, 可访问尚硅谷 (中国) 官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

执行上述命令后，可以按 P、M、N 对查询出的进程结果进行排序。

### 7.7.3 pstree 查看进程树

1) 基本语法:

`pstree [选项]`

选项

`-p:` 显示进程的 PID

`-u:` 显示进程的所属用户

2) 案例:

```
[root@hadoop102 datas]# pstree -u
```

```
[root@hadoop102 datas]# pstree -p
```

### 7.7.4 kill 终止进程

1) 基本语法:

`kill -9 pid 进程号`

选项

`-9` 表示强迫进程立即停止

2) 案例:

启动mysql程序

切换到root用户执行

```
[root@hadoop102 桌面]# kill -9 5102
```

### 7.7.5 netstat 显示网络统计信息

1) 基本语法:

`netstat -anp` (功能描述: 此命令用来显示整个系统目前的网络情况。例如目前的连接、数据包传递数据、或是路由表内容)

选项:

`-an` 按一定顺序排列输出

`-p` 表示显示哪个进程在调用

`-nltp` 查看 tcp 协议进程端口号

2) 案例

【更多 Java – Android 资料下载，可访问尚硅谷（中国）官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

查看端口50070的使用情况

```
[root@hadoop106 hadoop-2.7.2]# netstat -anp | grep 50070
tcp        0      0 0.0.0.0:50070      0.0.0.0:*          LISTEN      6816/java
           端口号                进程号
```

## 7.8 压缩和解压类

### 7.8.1 gzip/gunzip 压缩

1) 基本语法:

gzip+文件 (功能描述: 压缩文件, 只能将文件压缩为\*.gz 文件)

gunzip+文件.gz (功能描述: 解压缩文件命令)

2) 特点:

(1) 只能压缩文件不能压缩目录

(2) 不保留原来的文件

3) 案例

(1) gzip压缩

```
[root@hadoop106 opt]# ls
test.java
[root@hadoop106 opt]# gzip test.java
[root@hadoop106 opt]# ls
test.java.gz
```

(2) gunzip解压缩文件

```
[root@hadoop106 opt]# gunzip test.java.gz
[root@hadoop106 opt]# ls
test.java
```

### 7.8.2 tar 打包

1) 基本语法:

tar + 参数 + XXX.tar.gz + 将要打包进去的内容 (功能描述: 打包目录, 压缩后的文件格式.tar.gz)

参数:

-c 产生.tar打包文件

【更多 Java – Android 资料下载, 可访问尚硅谷(中国)官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

- v 显示详细信息
- f 指定压缩后的文件名
- z 打包同时压缩
- x 解包.tar文件

## 2) 案例

(1) 压缩: `tar -zcvf XXX.tar.gz n1.txt n2.txt`

压缩多个文件

```
[root@hadoop106 opt]# tar -zcvf test.tar.gz test1.java test.java  
test1.java  
test.java  
[root@hadoop106 opt]# ls  
test1.java  test.java  test.tar.gz
```

压缩目录

```
[root@hadoop106 opt]# tar -zcvf test.java.tar.gz test1  
test1/  
test1/hello  
test1/test1.java  
test1/test/  
test1/test/test.java  
[root@hadoop106 opt]# ls  
test1 test.java.tar.gz
```

(2) 解压: `tar -zxvf XXX.tar.gz`

解压到当前目录

```
[root@hadoop106 opt]# tar -zxvf test.tar.gz
```

解压到/opt目录

```
[root@hadoop106 opt]# tar -zxvf test.tar.gz -C /opt
```

## 7.8.3 zip/unzip 压缩

1) 基本语法:

`zip + 参数 + XXX.zip + 将要压缩的内容` (功能描述: 压缩文件和目录的命令, window/linux 通用且可以压缩目录且保留源文件)

参数:

- r 压缩目录

2) 案例:

【更多 Java – Android 资料下载, 可访问尚硅谷(中国)官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

- (1) 压缩 1.txt 和 2.txt, 压缩后的名称为 mypackage.zip

```
[root@hadoop106 opt]# zip test.zip test1.java test.java
adding: test1.java (stored 0%)
adding: test.java (stored 0%)
```

```
[root@hadoop106 opt]# ls
test1.java test.java test.zip
```

- (2) 解压 mypackage.zip

```
[root@hadoop106 opt]# unzip test.zip
Archive: test.zip
  extracting: test1.java
  extracting: test.java
```

```
[root@hadoop106 opt]# ls
test1.java test.java test.zip
```

## 7.9 后台服务管理类

### 7.9.1 service 后台服务管理

- 1) service network status 查看指定服务的状态
- 2) service network stop 停止指定服务
- 3) service network start 启动指定服务
- 4) service network restart 重启指定服务
- 5) service --status-all 查看系统中所有的后台服务

### 7.9.2 chkconfig 设置后台服务的自启配置

- 1) chkconfig 查看所有服务器自启配置
- 2) chkconfig iptables off 关掉指定服务的自动启动
- 3) chkconfig iptables on 开启指定服务的自动启动

## 7.10 crond 系统定时任务

### 7.10.1 crond 服务管理

```
[root@localhost ~]# service crond restart (重新启动服务)
```

### 7.10.2 crontab 定时任务设置

- 1) 基本语法

crontab [选项]

【更多 Java – Android 资料下载, 可访问尚硅谷(中国)官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

选项:

- e: 编辑 crontab 定时任务
- l: 查询 crontab 任务
- r: 删除当前用户所有的 crontab 任务

## 2) 参数说明

```
[root@localhost ~]# crontab -e
```

(1) 进入 crontab 编辑界面。会打开 vim 编辑你的工作。

\*\*\*\*\* 执行的任务

项目	含义	范围
第一个 “*”	一小时当中的第几分钟	0-59
第二个 “*”	一天当中的第几小时	0-23
第三个 “*”	一个月当中的第几天	1-31
第四个 “*”	一年当中的第几个月	1-12
第五个 “*”	一周当中的星期几	0-7 (0 和 7 都代表星期日)

(2) 特殊符号

特殊符号	含义
*	代表任何时间。比如第一个 “*” 就代表一小时中每分钟都执行一次的意思。
,	代表不连续的时间。比如 “0 8,12,16 *** 命令”，就代表在每天的 8 点 0 分，12 点 0 分，16 点 0 分都执行一次命令
-	代表连续的时间范围。比如 “0 5 * * 1-6 命令”，代表在周一到周六的凌晨 5 点 0 分执行命令
*/n	代表每隔多久执行一次。比如 “*/10 * * * * 命令”，代表每隔 10 分钟就执行一遍命令

(3) 特定时间执行命令

时间	含义
45 22 *** 命令	在 22 点 45 分执行命令
0 17 ** 1 命令	每周 1 的 17 点 0 分执行命令
0 5 1,15 ** 命令	每月 1 号和 15 号的凌晨 5 点 0 分执行命令
40 4 ** 1-5 命令	每周一到周五的凌晨 4 点 40 分执行命令
*/10 4 *** 命令	每天的凌晨 4 点，每隔 10 分钟执行一次命令
0 0 1,15 * 1 命令	每月 1 号和 15 号，每周 1 的 0 点 0 分都会执行命令。注意：星期几和几号最好不要同时出现，因为他们定义的都是天。非常容易让管理员混乱。

## 3) 案例:

```
*/5 * * * * /bin/echo "11" >> /tmp/test
```



## 八、VI/VIM 编辑器

### 8.1 概述

所有的 Unix Like 系统都会内建 vi 文书编辑器，其他的文书编辑器则不一定会存在。但是目前我们使用比较多的是 vim 编辑器。

Vim 具有程序编辑的能力，可以主动的以字体颜色辨别语法的正确性，方便程序设计。Vim 是从 vi 发展出来的一个文本编辑器。代码补完、编译及错误跳转等方便编程的功能特别丰富，在程序员中被广泛使用。

简单的来说 vi 是老式的字处理器，不过功能已经很齐全了，但是还是有可以进步的地方。vim 则可以说是程序开发者的一项很好用的工具。连 vim 的官方网站 (<http://www.vim.org>) 自己也说 vim 是一个程序开发工具而不是文字处理软件。

### 8.2 测试数据准备



vi测试数据.txt

### 8.3 一般模式

以 vi 打开一个档案就直接进入一般模式了(这是默认的模式)。在这个模式中，你可以使用『上下左右』按键来移动光标，你可以使用『删除字符』或『删除整行』来处理档案内容，也可以使用『复制、贴上』来处理你的文件数据。

#### 常用语法

- 1) yy            (功能描述: 复制光标当前一行)
- y数字y      (功能描述: 复制一段(从第几行到第几行))
- 2) p            (功能描述: 箭头移动到目的行粘贴)
- 3) u            (功能描述: 撤销上一步)
- 4) dd           (功能描述: 删除光标当前行)
- d数字d      (功能描述: 删除光标(含)后多少行)
- 5) x            (功能描述: 删除一个字母, 相当于del)
- X            (功能描述: 删除一个字母, 相当于Backspace)

- 6) yw            (功能描述: 复制一个词)
- 7) dw            (功能描述: 删除一个词)
- 8) shift+^        (功能描述: 移动到行头)
- 9) shift+\$        (功能描述: 移动到行尾)
- 10) 1+shift+g      (功能描述: 移动到页头, 数字)
- 11) shift+g        (功能描述: 移动到页尾)
- 12) 数字N+shift+g    (功能描述: 移动到目标行)

## 8.4 编辑模式

在一般模式中可以进行删除、复制、贴上等等的动作,但是却无法编辑文件内容的! 要等到你按下『i, I, o, O, a, A, r, R』等任何一个字母之后才会进入编辑模式。

注意了! 通常在 Linux 中, 按下这些按键时, 在画面的左下方会出现『INSERT 或 REPLACE』的字样, 此时才可以进行编辑。而如果要回到一般模式时, 则必须要按下『Esc』这个按键即可退出编辑模式。

### 常用语法

- 1) 进入编辑模式
  - (1) i      当前光标前
  - (2) a      当前光标后
  - (3) o      当前光标行的下一行
- 2) 退出编辑模式
  - 按『Esc』键

## 8.5 指令模式

在一般模式当中, 输入『:/?』3个中的任何一个按钮, 就可以将光标移动到最底下那一行。

在这个模式当中, 可以提供你『搜寻资料』的动作, 而读取、存盘、大量取代字符、离开 vi、显示行号等动作是在此模式中达成的!

### 常用语法

- 1) 基本语法
  - (1) : 选项
  - 选项:

w 保存

q 退出

! 感叹号强制执行

(2) / 查找, /被查找词, n是查找下一个, shift+n是往上查找

(3) ? 查找, ?被查找词, n是查找上一个, shift+n是往下查找

## 2) 案例

:wq! 强制保存退出

## 九、rpm

### 9.1 概述

RPM (RedHat Package Manager), Redhat软件包管理工具, 类似windows里面的setup.exe是Linux这系列操作系统里面的打包安装工具, 它虽然是RedHat的标志, 但理念是通用的。

RPM包的名称格式

Apache-1.3.23-11.i386.rpm

- “apache” 软件名称
- “1.3.23-11”软件的版本号, 主版本和此版本
- “i386”是软件所运行的硬件平台
- “rpm”文件扩展名, 代表RPM包

### 9.2 常用命令

#### 9.2.1 查询

1) 基本语法:

(1) rpm -qa (功能描述: 查询所安装的所有 rpm 软件包)

过滤

rpm -qa | more

rpm -qa | grep X

rpm -qa | grep -l 软件名

(2) rpm -q 软件包名 (功能描述: 查询软件包是否安装)

(3) rpm -qi 软件包名 (功能描述: 查询软件包信息)

(4) `rpm -ql` 软件包名 (功能描述: 查询软件包中文件)

(5) `rpm -qf` 文件全路径名 (功能描述: 查询文件所属的软件包)

```
rpm -qf /etc/passwd
```

```
rpm -qf /root/install.log
```

(6) `rpm -qp` 包文件名 (功能描述: 查询包的信息对这个软件包的介绍)

```
rpm -qp jdk-1_5_0-linux-i586.rpm
```

```
rpm -qpi jdk-1_5_0-linux-i586.rpm
```

```
rpm -qpl jdk-1_5_0-linux-i586.rpm
```

## 9.2.2 安装

1) 基本语法:

```
rpm -ivh RPM 包全名
```

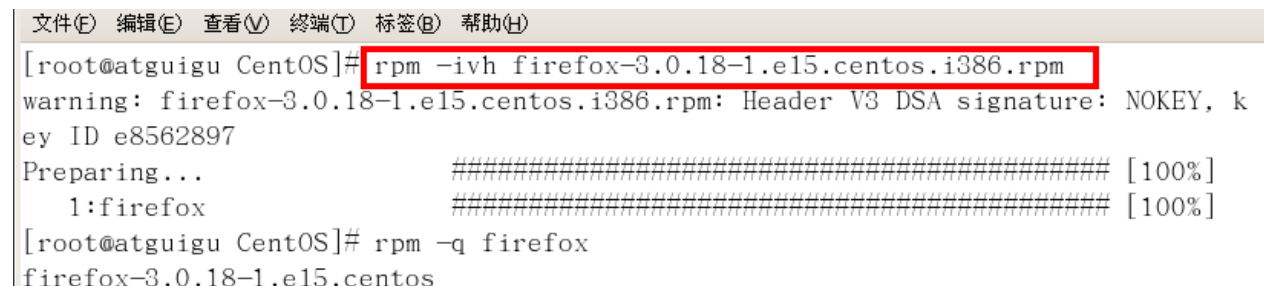
-i=install, 安装

-v=verbose, 显示详细信息

-h=hash, 进度条

--nodeps, 不检测依赖进度

2) 案例



```
文件(E) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)
[root@atguigu CentOS]# rpm -ivh firefox-3.0.18-1.el5.centos.i386.rpm
warning: firefox-3.0.18-1.el5.centos.i386.rpm: Header V3 DSA signature: NOKEY, k
ey ID e8562897
Preparing... ##### [100%]
   1:firefox ##### [100%]
[root@atguigu CentOS]# rpm -q firefox
firefox-3.0.18-1.el5.centos
```

## 9.2.3 卸载

1) 基本语法:

(1) `rpm -e RPM软件包`

或者 (2) `rpm -e --nodeps 软件包`

--nodeps 如果该RPM包的安装依赖其它包, 即使其它包没装, 也强迫安装。

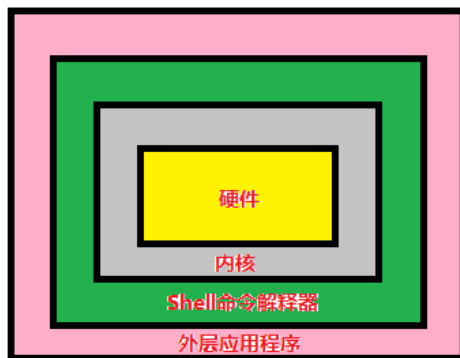
2) 案例

```
[root@atguigu ~]# rpm -q firefox
firefox-3.0.18-1.el5.centos
[root@atguigu ~]# rpm -e firefox-3.0.18-1.el5.centos
[root@atguigu ~]# rpm -q firefox
package firefox is not installed
[root@atguigu ~]#
```

## 十、shell 编程

### 10.1 概述

Shell 是一个命令行解释器，它为用户提供了一个向 Linux 内核发送请求以便运行程序的界面系统级程序，用户可以用 Shell 来启动、挂起、停止甚至是编写一些程序。



Shell 还是一个功能相当强大的编程语言，易编写、易调试、灵活性强。Shell 是解释执行的脚本语言，在 Shell 中可以调用 Linux 系统命令。

### 10.2 shell 脚本的执行方式

#### 1) echo 输出命令

##### (1) 基本语法：

echo [选项] [输出内容]

选项：

-e: 支持反斜线控制的字符转换

控制字符	作用
\\	输出\本身
\a	输出警告音
\b	退格键，也就是向左删除键
\c	取消输出行末的换行符。和“-n”选项一致
\e	ESCAPE 键
\f	换页符

\n	换行符
\r	回车键
\t	制表符，也就是 Tab 键
\v	垂直制表符
\0nnn	按照八进制 ASCII 码表输出字符。其中 0 为数字零，nnn 是三位八进制数
\xhh	按照十六进制 ASCII 码表输出字符。其中 hh 是两位十六进制数

## (2) 案例

```
[atguigu@hadoop102 sbin]$ echo "helloworld"
```

helloworld

## 2) 第一个 Shell 脚本

(1) 需求：创建一个 Shell 脚本，输出 helloworld

(2) 实操：

```
[atguigu@hadoop102 datas]$ touch helloworld.sh  
[atguigu@hadoop102 datas]$ vi helloworld.sh
```

在 helloworld.sh 中输入如下内容

```
#!/bin/bash  
echo "helloworld"
```

## 3) 脚本的常用执行方式

第一种：输入脚本的绝对路径或相对路径

(1) 首先要赋予 helloworld.sh 脚本的 +x 权限

```
[atguigu@hadoop102 datas]$ chmod 777 helloworld.sh
```

(2) 执行脚本

```
/root/helloWorld.sh
```

```
./helloWorld.sh
```

第二种：bash 或 sh + 脚本（不用赋予脚本 +x 权限）

```
sh /root/helloWorld.sh
```

```
sh helloWorld.sh
```

## 10.3 shell 中的变量

1) Linux Shell 中的变量分为“系统变量”和“用户自定义变量”，可以通过 set 命令查看系统变量。

2) 系统变量：\$HOME、\$PWD、\$SHELL、\$USER 等等

【更多 Java – Android 资料下载，可访问尚硅谷（中国）官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

3) 显示当前 shell 中所有变量: set

### 10.3.1 定义变量

1) 基本语法:

变量=值

2) 变量定义规则

- (1) 变量名称可以由字母、数字和下划线组成, 但是不能以数字开头。
- (2) 等号两侧不能有空格
- (3) 变量名称一般习惯为大写
- (4) 双引号和单引号有区别, 双引号仅将空格脱意, 单引号会将所有特殊字符脱意

3) 案例

STR="hello world"

A=8            定义变量A

unset A        撤销变量A

readonly B=2   声明静态的变量B=2, 不能unset

export 变量名   可把变量提升为全局环境变量, 可供其他shell程序使用

### 10.3.2 将命令的返回值赋给变量

A=`ls -la` 反引号, 运行里面的命令, 并把结果返回给变量 A

A=\$(ls -la) 等价于反引号

### 10.3.3 设置环境变量

1) 基本语法:

export 变量名=变量值            (功能描述: 设置环境变量的值)

echo \$变量名                    (功能描述: 查询环境变量的值)

source 配置文件                (功能描述: 让修改后的配置信息立即生效)

2) 案例:

在/etc/profile 文件中定义 JAVA\_HOME 环境变量

export JAVA\_HOME=/opt/module/jdk1.7.0\_79

export PATH=\$PATH:\$JAVA\_HOME/bin



查看环境变量 JAVA\_HOME 的值

```
[atguigu@hadoop102 datas]$ echo $JAVA_HOME
```

```
/opt/module/jdk1.7.0_79
```

### 10.3.4 位置参数变量

#### 1) 基本语法

**\$n** （功能描述：n 为数字，\$0 代表命令本身，\$1-\$9 代表第一到第九个参数，十以上的参数，十以上的参数需要用大括号包含，如\${10}）

**\$\*** （功能描述：这个变量代表命令行中所有的参数，\$\*把所有的参数看成一个整体）

**\$@** （功能描述：这个变量也代表命令行中所有的参数，不过\$@把每个参数区分对待）

**\$#** （功能描述：这个变量代表命令行中所有参数的个数）

#### 2) 案例

(1) 计算输入的参数 1 和参数 2 的两个数的和，并输出到控制台

```
#!/bin/bash
num1=$1
num2=$2
sum=$(( $num1 + $num2))
#变量 sum 的和是 num1 加 num2
echo $sum
#打印变量 sum 的值
```

(2) 打印输入的参数总数、所有参数

```
#!/bin/bash
echo "A total of $# parameters"
#使用$#代表所有参数的个数
echo "The parameters is: $*"
#使用$*代表所有的参数
echo "The parameters is: $@"
#使用$@也代表所有参数
```

(3) \$\*与\$@的区别

```
#!/bin/bash
for i in "$*"
#*$中的所有参数看成是一个整体，所以这个 for 循环只会循环一次
do
    echo "The parameters is: $i"
done
x=1
for y in "$@"
```

```
# $@ 中的每个参数都看成是独立的，所以 "$@" 中有几个参数，就会循环几次
do
    echo "The parameter $x is: $y"
    x=$(( $x + 1 ))
done
```

a) `$*`和`$@`都表示传递给函数或脚本的所有参数，不被双引号“”包含时，都以`$1 $2 ... $n`的形式输出所有参数

b) 当它们被双引号“”包含时，“`$*`”会将所有的参数作为一个整体，以“`$1 $2 ... $n`”的形式输出所有参数；“`$@`”会将各个参数分开，以“`$1`” “`$2`” ... “`$n`”的形式输出所有参数

### 10.3.5 预定义变量

#### 1) 基本语法：

`$?` （功能描述：最后一次执行的命令的返回状态。如果这个变量的值为0，证明上一个命令正确执行；如果这个变量的值为非0（具体是哪个数，由命令自己来决定），则证明上一个命令执行不正确了。）

`$$` （功能描述：当前进程的进程号（PID））

`$!` （功能描述：后台运行的最后一个进程的进程号（PID））

#### 2) 案例

```
#!/bin/bash

#输出当前进程的PID，这个PID就是当前这个脚本执行时，生成的进程的PID
echo "The current process is $$"

#使用find命令在root目录下查找hello.sh文件，符号&的意思是把命令放入后台执行
find /root -name hello.sh &

echo "The last one Daemon process is $!"

echo "$?"
```

## 10.4 运算符

#### 1) 基本语法：

(1) “`$((运算式))`” 或 “`${运算式}`”

(2) `expr m + n`

注意 `expr` 运算符间要有空格

2) 案例：计算  $(2+3) \times 4$  的值

(1) 采用 `$(运算式)` 方式

```
[root@hadoop102 datas]# S=$[(2+3)*4]
```

```
[root@hadoop102 datas]# echo $S
```

(2) `expr` 分布计算

```
S=`expr 2 + 3`
```

```
expr $S \* 4
```

(3) `expr` 一步完成计算

```
expr `expr 2 + 3` \* 4
```

```
echo `expr `expr 2 + 3` \* 4`
```

## 10.5 条件判断

### 10.5.1 判断语句

1) 基本语法：

`[ condition ]`（注意 `condition` 前后要有空格）

#非空返回 `true`，可使用 `$?` 验证（0 为 `true`，>1 为 `false`）

2) 案例：

```
[atguigu]      返回 true
```

```
[]            返回 false
```

```
[condition] && echo OK || echo notok      条件满足，执行后面的语句
```

### 10.5.2 常用判断条件

1) 两个整数之间比较

`=` 字符串比较

`-lt` 小于

`-le` 小于等于

`-eq` 等于

`-gt` 大于

-ge 大于等于

-ne 不等于

## 2) 按照文件权限进行判断

-r 有读的权限

-w 有写的权限

-x 有执行的权限

## 3) 按照文件类型进行判断

-f 文件存在并且是一个常规的文件

-e 文件存在

-d 文件存在并是一个目录

## 4) 案例

```
[root@localhost ~]# [ 23 -ge 22 ]
```

```
[root@localhost ~]# [ -w student.txt ]
```

```
[root@localhost ~]# [ -e /root/install.log ]
```

# 10.6 流程控制

## 10.6.1 if 判断

### 1) 基本语法:

```
if [ 条件判断式 ];then
```

```
    程序
```

```
fi
```

或者

```
if [ 条件判断式 ]
```

```
then
```

```
    程序
```

```
fi
```

注意事项: (1) [ 条件判断式 ], 中括号和条件判断式之间必须有空格

### 2) 案例

```
#!/bin/bash
read -p "please input your name:" NAME
```

【更多 Java – Android 资料下载, 可访问尚硅谷(中国)官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

```
#printf '%s\n' $NAME
if[ $NAME = root ]
    then
        echo "hello ${NAME}, welcome !"
    elif [ $NAME = atguigu ]
        then
            echo "hello ${NAME}, welcome !"
        else
            echo "sorry "
    fi
```

### 10.6.2 case 语句

#### 1) 基本语法:

```
case $变量名 in
    "值 1")
        如果变量的值等于值 1，则执行程序 1
        ;;
    "值 2")
        如果变量的值等于值 2，则执行程序 2
        ;;
    ...省略其他分支...
    *)
        如果变量的值都不是以上的值，则执行此程序
        ;;
esac
```

#### 2) 案例

```
case $1 in
start)
    echo "starting"
    ;;
stop)
    echo "stoping"
    ;;
```

```
*)  
  
    echo "Usage: {start|stop}"  
  
esac
```

### 10.6.3 for 循环

#### 1) 基本语法 1:

for 变量 in 值 1 值 2 值 3...

do

程序

done

#### 2) 案例:

##### (1) 打印时间

```
#!/bin/bash  
#打印时间  
  
for time in morning noon afternoon evening  
do  
    echo "This time is $time!"  
done
```

#### 3) 基本语法 2:

for (( 初始值;循环控制条件;变量变化 ))

do

程序

done

#### 4) 案例

##### (1) 从 1 加到 100

```
#!/bin/bash  
#从 1 加到 100  
  
s=0  
for (( i=1;i<=100;i=i+1 ))  
do  
    s=$(( $s+$i ))  
done  
echo "The sum is : $s"
```

### 10.6.4 while 循环

#### 1) 基本语法:

```
while [ 条件判断式 ]  
  
do  
  
    程序  
  
done
```

#### 2) 案例

##### (1) 从 1 加到 100

```
#!/bin/bash  
#从 1 加到 100  
  
i=1  
s=0  
while [ $i -le 100 ]  
#如果变量 i 的值小于等于 100，则执行循环  
do  
    s=$(( $s+$i ))  
    i=$(( $i+1 ))  
done  
echo "The sum is: $s"
```

### 10.7 read 读取控制台输入

#### 1) 基本语法:

read(选项)(参数)

选项:

- p: 指定读取值时的提示符;
- t: 指定读取值时等待的时间（秒）。

参数

变量: 指定读取值的变量名

#### 2) 案例

```
[atguigu@hadoop101 etc]$ read -p "please input your name:" NAME  
please input your name:lilei  
[atguigu@hadoop101 etc]$ echo $NAME
```

lilei

## 10.8 函数

### 10.8.1 系统函数

#### 1) basename 基本语法

`basename [pathname] [suffix]`

`basename [string] [suffix]`      （功能描述： `basename` 命令会删掉所有的前缀包括最后一个（‘/’）字符，然后将字符串显示出来。

选项：

`suffix` 为后缀，如果 `suffix` 被指定了，`basename` 会将 `pathname` 或 `string` 中的 `suffix` 去掉。

#### 2) 案例

```
[atguigu@hadoop102 opt]$ basename /opt/test.txt
```

```
test.txt
```

```
[atguigu@hadoop102 opt]$ basename /opt/test.txt .txt
```

```
test
```

#### 3) dirname 基本语法

`dirname` 文件绝对路径      （功能描述：从给定的包含绝对路径的文件名中去除文件名

（非目录的部分），然后返回剩下的路径（目录的部分））

#### 4) 案例

```
[atguigu@hadoop102 opt]$ dirname /opt/test.txt
```

```
/opt
```

### 10.8.2 自定义函数

#### 1) 基本语法：

```
[ function ] funname()  
{  
    Action;  
    [return int;]  
}
```



function start() / function start / start()

注意：

(1) 必须在调用函数地方之前，先声明函数，shell 脚本是逐行运行。不会像其它语言一样先编译。

(2) 函数返回值，只能通过\$?系统变量获得，可以显示加：return 返回，如果不加，将以最后一条命令运行结果，作为返回值。return 后跟数值 n(0-255)

## 2) 案例

(1) 打印出比你输入小的所有数（单参）

```
#!/bin/bash

function LoopPrint()
{
    count=0;
    while [ $count -lt $1 ];
    do
        echo $count;
        expr ++count;
        sleep 1;
    done
    return 0;
}

read -p "Please input the number: " n;
LoopPrint $n;
```

(2) 多参

```
#!/bin/bash

function LoopPrint()
{
    echo $2
    count=0;
    while [ $count -lt $1 ];
```

```
do
    echo $count;
    expr ++count;
    sleep 1;
done
return 0;
}

read -p "Please input the num1: " n;
read -p "Please input the num2: " m;
LoopPrint $n $m;
```

## 十一、yum 仓库配置

### 11.1 概述

YUM（全称为 Yellow dog Updater, Modified）是一个在 Fedora 和 RedHat 以及 CentOS 中的 Shell 前端软件包管理器。基于 RPM 包管理，能够从指定的服务器自动下载 RPM 包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软件包，无须繁琐地一次次下载、安装。

在 Linux 上使用源码的方式安装软件非常满分，使用 yum 可以简化安装的过程

### 11.2 yum 的常用命令

#### 1) 基本语法:

- |                                   |                                    |
|-----------------------------------|------------------------------------|
| <code>yum install -y httpd</code> | （功能描述：安装 httpd 并确认安装）              |
| <code>yum list</code>             | （功能描述：列出所有可用的 package 和 package 组） |
| <code>yum clean all</code>        | （功能描述：清除所有缓冲数据）                    |
| <code>yum deplist httpd</code>    | （功能描述：列出一个包所有依赖的包）                 |
| <code>yum remove httpd</code>     | （功能描述：删除 httpd）                    |

#### 2) 案例实操

```
yum install -y tree
```

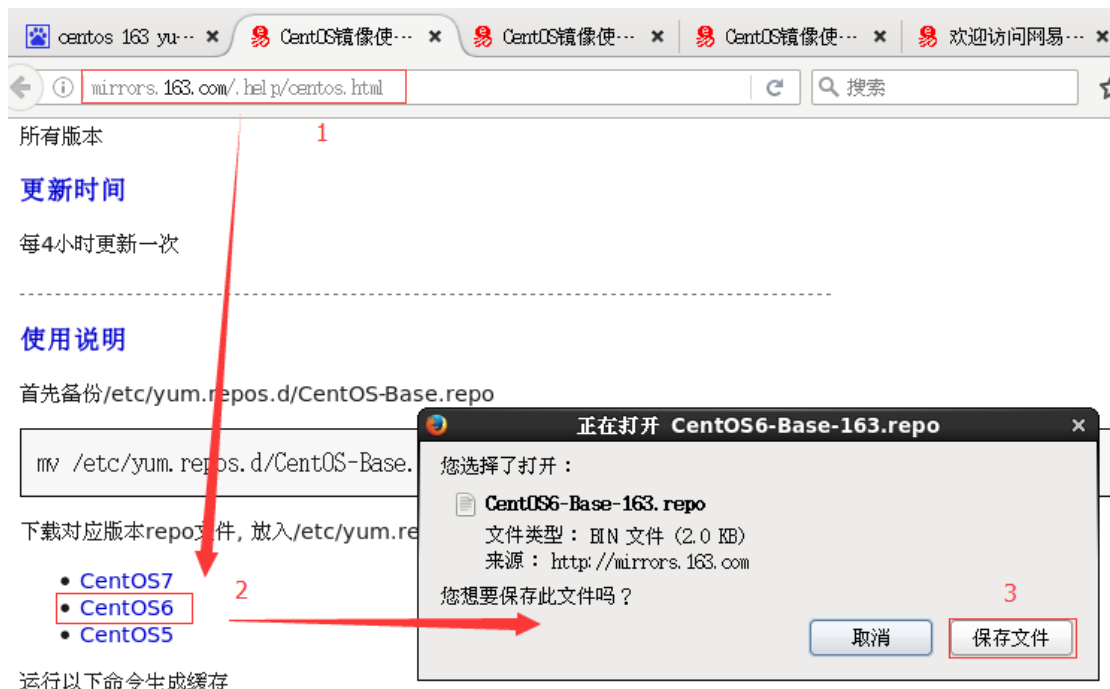
### 11.3 关联网络 yum 源

#### 1) 前期文件准备

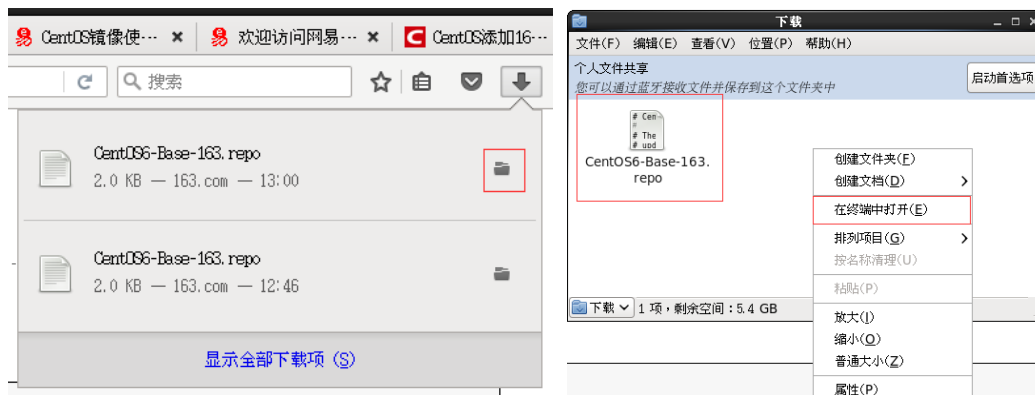
（1）前提条件 linux 系统必须可以联网

【更多 Java – Android 资料下载，可访问尚硅谷（中国）官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

(2) 在 Linux 环境环境中访问该网络地址: <http://mirrors.163.com/help/centos.html>, 在使用说明中点击 CentOS6->再点击保存



(3) 查看文件保存的位置



在打开的终端中输入如下命令, 就可以找到文件的保存位置。

```
[atguigu@hadoop101 下载]$ pwd
```

```
/home/atguigu/下载
```

2) 替换本地 yum 文件

(1) 把下载的文件移动到/etc/yum.repos.d/目录

```
[root@hadoop101 下载]# mv CentOS6-Base-163.repo /etc/yum.repos.d/
```

(2) 进入到/etc/yum.repos.d/目录

```
[root@hadoop101 yum.repos.d]# pwd
```

【更多 Java - Android 资料下载, 可访问尚硅谷 (中国) 官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

/etc/yum.repos.d

(3) 用 CentOS6-Base-163.repo 替换 CentOS-Base.repo

```
[root@hadoop101 yum.repos.d]# mv CentOS6-Base-163.repo CentOS-Base.repo
```

3) 安装命令

(1) `[root@hadoop101 yum.repos.d]#yum clean all`

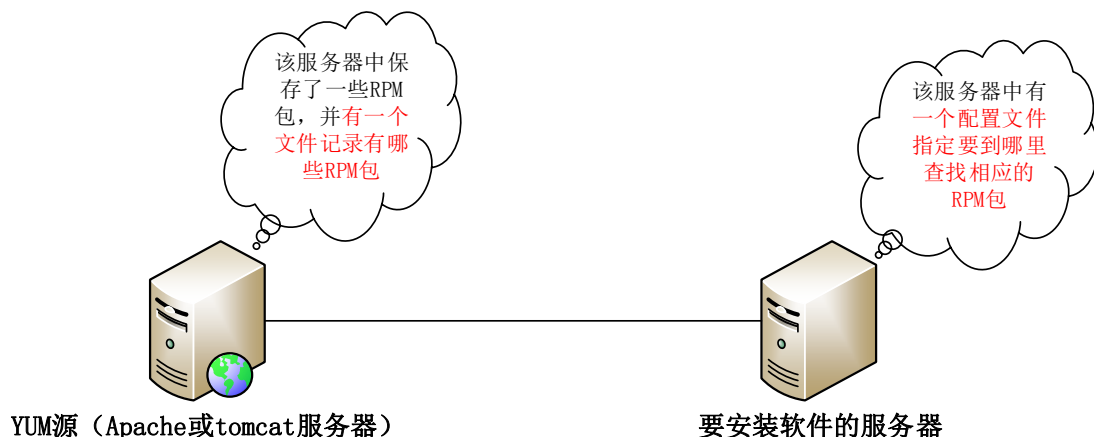
(2) `[root@hadoop101 yum.repos.d]#yum makecache`

## 11.4 制作本地 yum 源

1) 为什么要制作本地 YUM 源

YUM 源虽然可以简化我们在 Linux 上安装软件的过程，但是生成环境通常无法上网，不能连接外网的 YUM 源，所以接就无法使用 yum 命令安装软件了。为了在内网中也可以使用 yum 安装相关的软件，就要配置 yum 源。

YUM 源其实就是一个保存了多个 RPM 包的服务器，可以通过 http 的方式来检索、下载并安装相关的 RPM 包



2) 制作本地 YUM 源

(1) 准备一台 Linux 服务器，版本 CentOS-6.8-x86\_64-bin-DVD1.iso

(2) 配置好这台服务器的 IP 地址

(3) 将 CentOS-6.8-x86\_64-bin-DVD1.iso 镜像挂载到/mnt/cdrom 目录

```
[root@hadoop101 /]# mkdir /mnt/cdrom
```

```
[root@hadoop101 /]# mount -t iso9660 /dev/cdrom /mnt/cdrom
```

(4) 修改本机上的 YUM 源配置文件，将源指向自己

备份原有的 YUM 源的配置文件

```
[root@hadoop101 /]# cd /etc/yum.repos.d/
```

```
[root@hadoop101 yum.repos.d]# cp CentOS-Base.repo CentOS-Base.repo.bak
```

编辑 CentOS-Base.repo 文件

```
[root@hadoop101 yum.repos.d]# vi CentOS-Base.repo
```

```
[base]
name=CentOS-Local
baseurl=file:///var/iso
gpgcheck=1
enabled=1    #增加改行，使能
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
```

添加上面内容保存退出

(6) 清除 YUM 缓冲

```
[root@hadoop101 yum.repos.d]# yum clean all
```

(7) 列出可用的 YUM 源

```
[root@hadoop101 yum.repos.d]# yum repolist
```

(8) 安装相应的软件

```
[root@hadoop101 yum.repos.d]# yum install -y httpd
```

(9) 开启 httpd 使用浏览器访问 <http://192.168.11.101:80> (如果访问不通, 检查防火墙是否开启了 80 端口或关闭防火墙)

```
[root@hadoop101 yum.repos.d]# service httpd start
```

(10) 将 YUM 源配置到 httpd (Apache Server) 中, 其他的服务器即可通过网络访问这个内网中的 YUM 源了

```
[root@hadoop101 yum.repos.d]# cp -r /mnt/cdrom/ /var/www/html/CentOS
```

(11) 取消先前挂载的镜像

```
[root@hadoop101 yum.repos.d]# umount /mnt/cdrom
```

(12) 在浏览器中访问 <http://192.168.11.101/CentOS/>



(13) 让其他需要安装 RPM 包的服务器指向这个 YUM 源，准备一台新的服务器，备份或删除原有的 YUM 源配置文件

备份原有的 YUM 源的配置文件

```
[root@hadoop102 /]# cd /etc/yum.repos.d/
```

```
[root@hadoop102 yum.repos.d]# cp CentOS-Base.repo CentOS-Base.repo.bak
```

编辑 CentOS-Base.repo 文件

```
[root@hadoop102 yum.repos.d]# vi CentOS-Base.repo
```

```
[base]

name=CentOS-hadoop101

baseurl=http://192.168.11.101/CentOS

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
```

添加上面内容保存退出

(14) 在这台新的服务器上执行 YUM 的命令

```
[root@hadoop102 yum.repos.d]# yum clean all
```

```
[root@hadoop102 yum.repos.d]# yum repolist
```