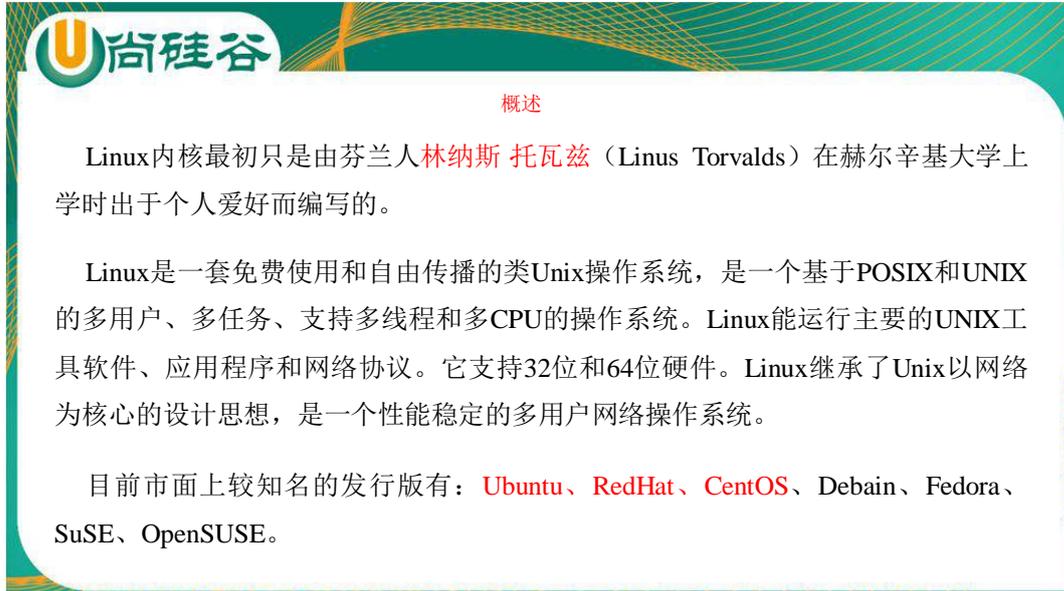


## 第 1 章 Linux 入门

### 1.1 概述



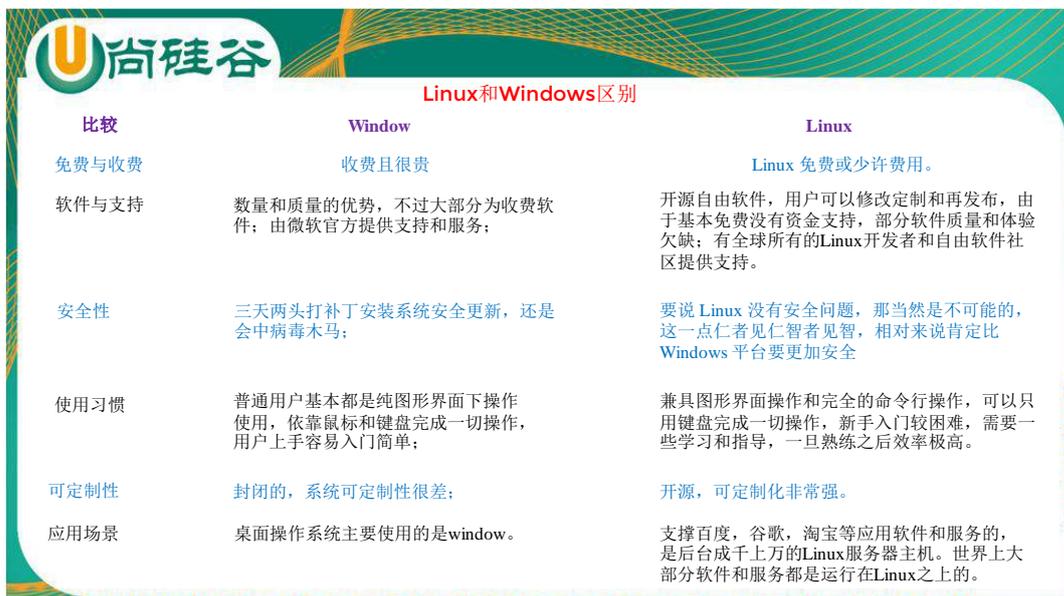
**概述**

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 Linux 和 Windows 区别



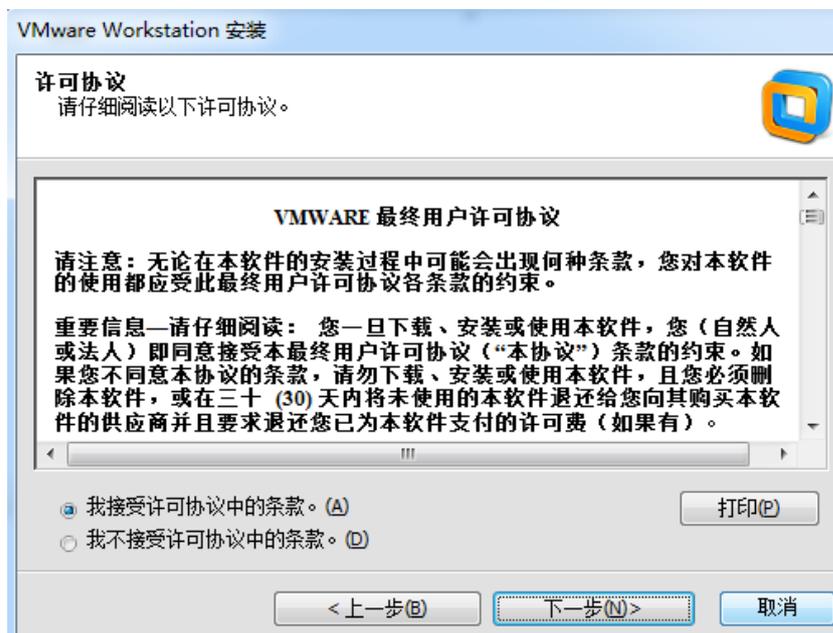
Linux和Windows区别		
比较	Window	Linux
免费与收费	收费且很贵	Linux 免费或少许费用。
软件与支持	数量和质量的劣势，不过大部分为收费软件；由微软官方提供支持和服务；	开源自由软件，用户可以修改定制和再发布，由于基本免费没有资金支持，部分软件质量和体验欠缺；有全球所有的Linux开发者和自由软件社区提供支持。
安全性	三天两头打补丁安装系统安全更新，还是会中病毒木马；	要说 Linux 没有安全问题，那当然是不可能的，这一点仁者见仁智者见智，相对来说肯定比 Windows 平台要更加安全
使用习惯	普通用户基本都是纯图形界面下操作使用，依靠鼠标和键盘完成一切操作，用户上手容易入门简单；	兼具图形界面操作和完全的命令行操作，可以只用键盘完成一切操作，新手入门较困难，需要一些学习和指导，一旦熟练之后效率极高。
可定制性	封闭的，系统可定制性很差；	开源，可定制化非常强。
应用场景	桌面操作系统主要使用的是window。	支撑百度，谷歌，淘宝等应用软件和服务的，是后台成千上万的Linux服务器主机。世界上大部分软件和服务都是运行在Linux之上的。

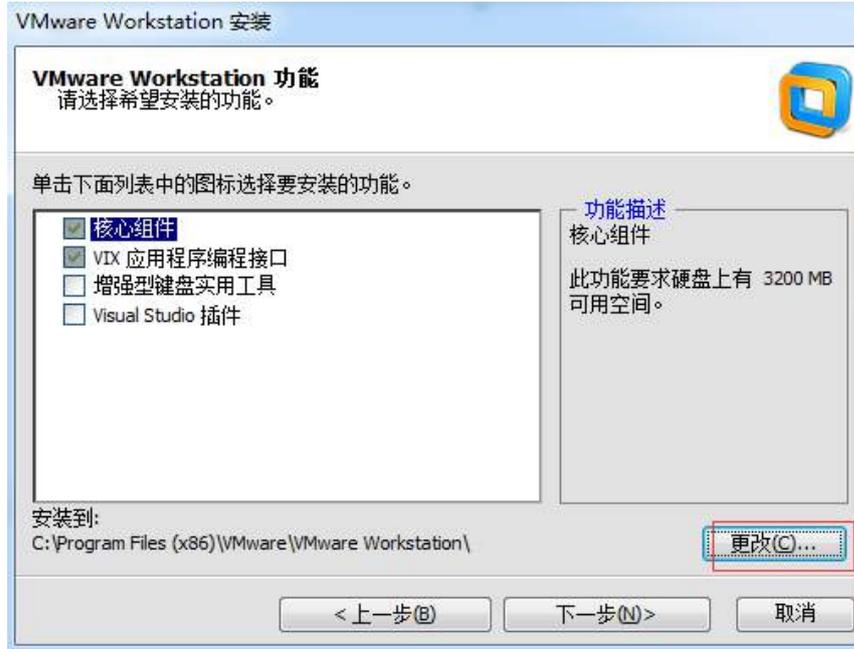
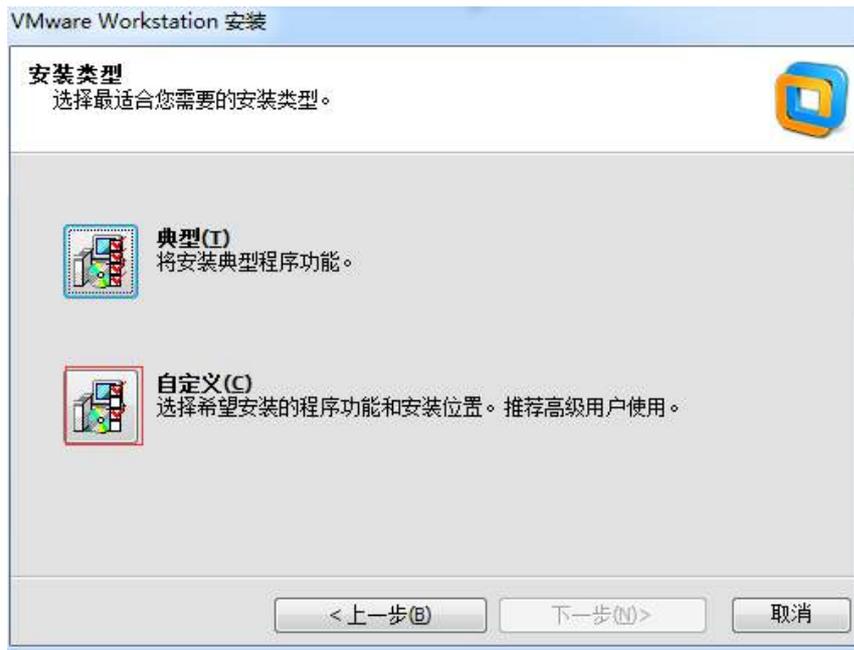
### 1.3 Centos 下载地址

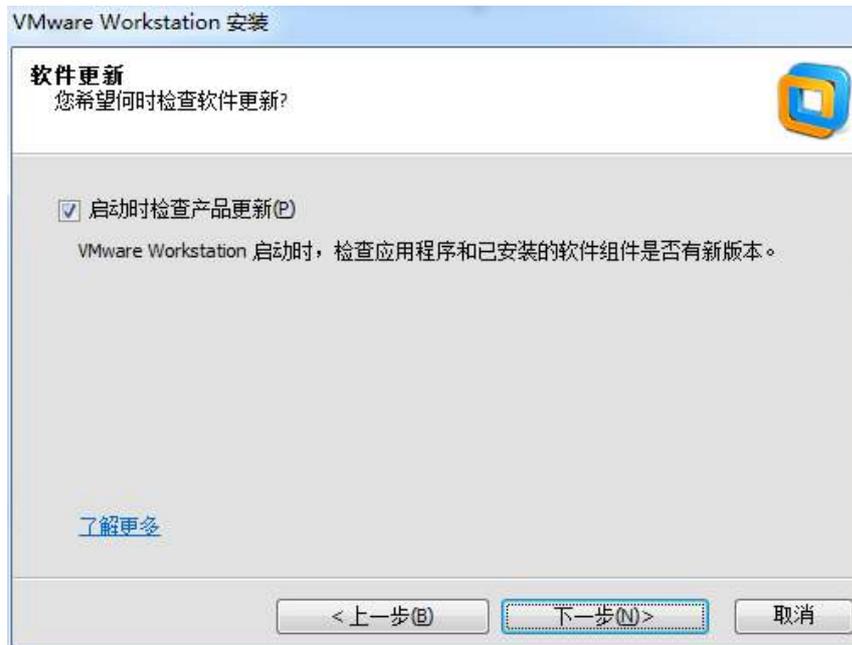
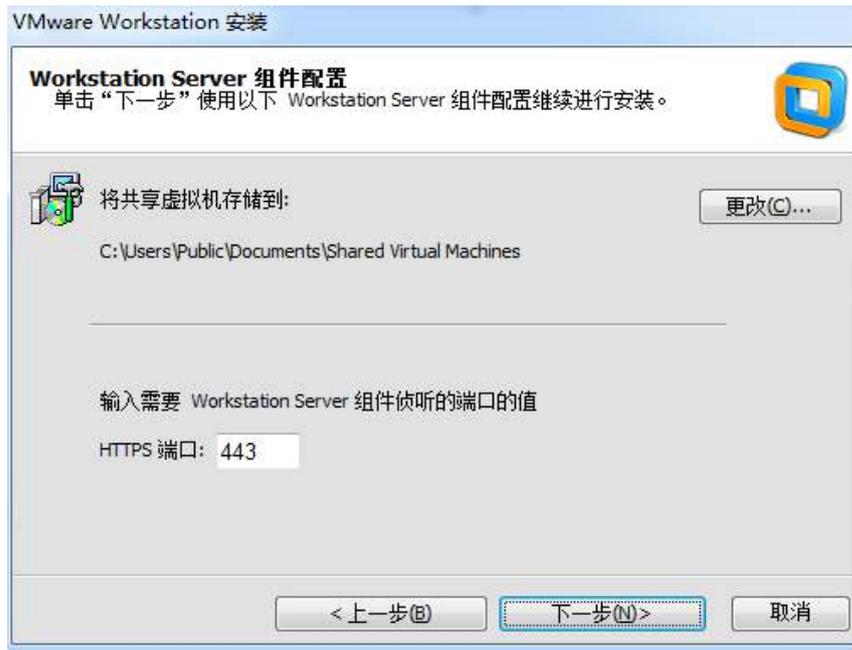


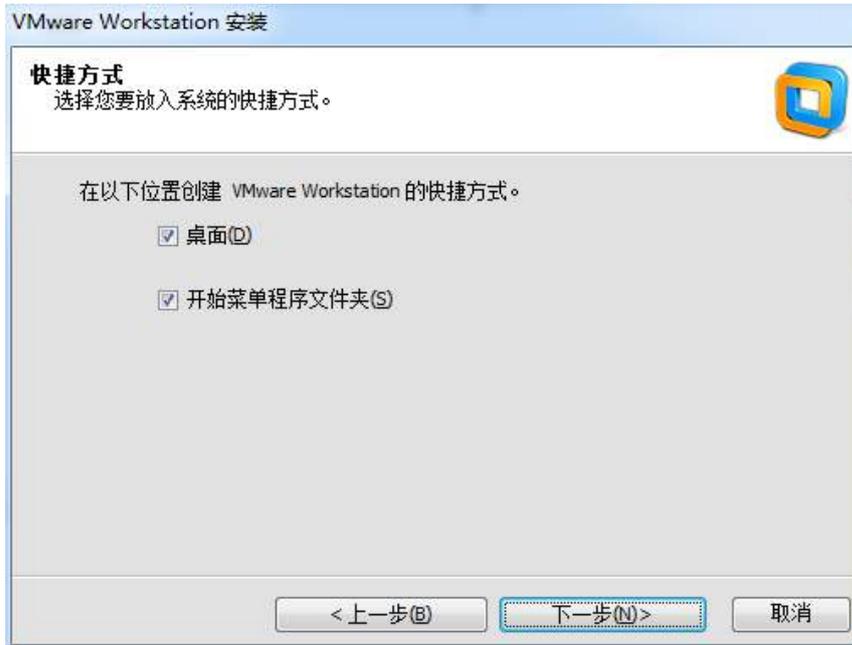
## 第 2 章 VM 与 Linux 的安装

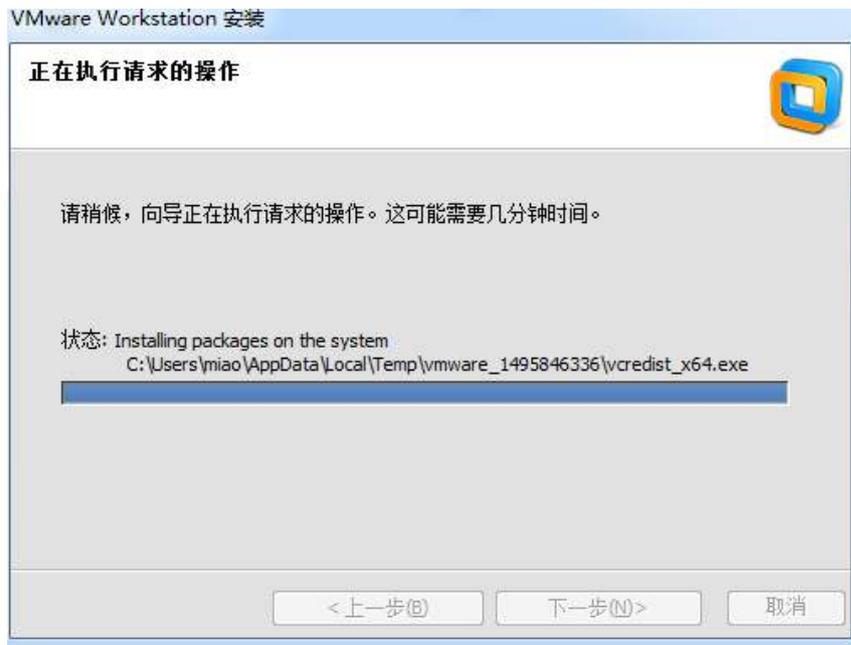
### 2.1 VMWare 安装

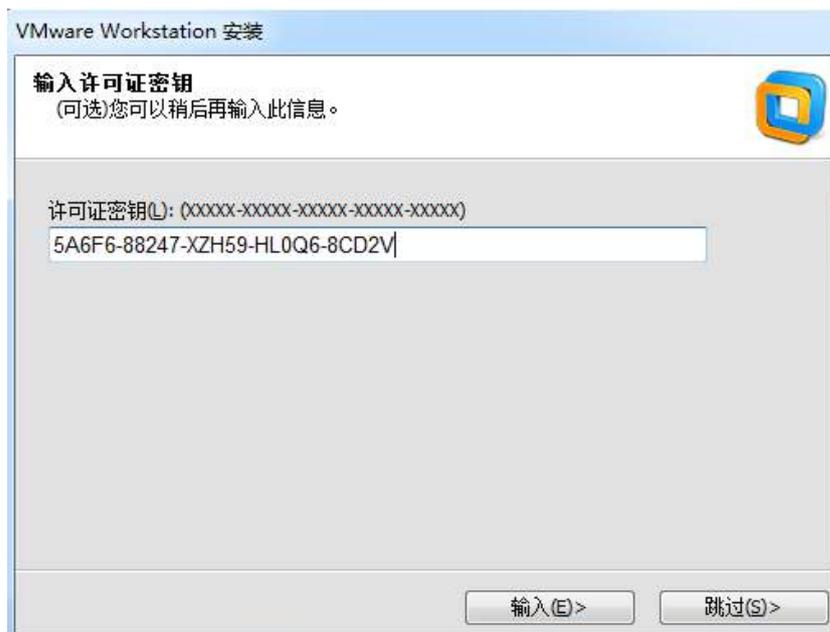
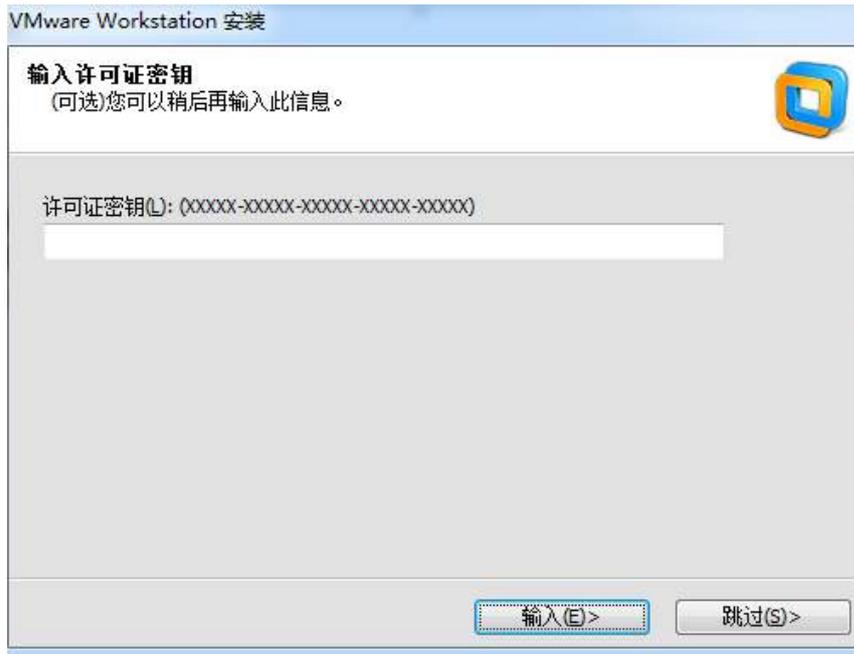


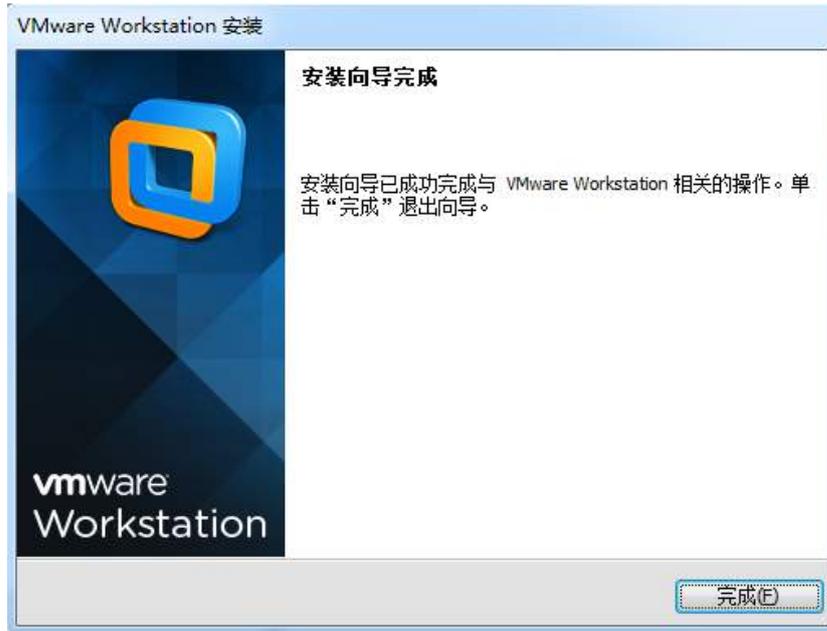






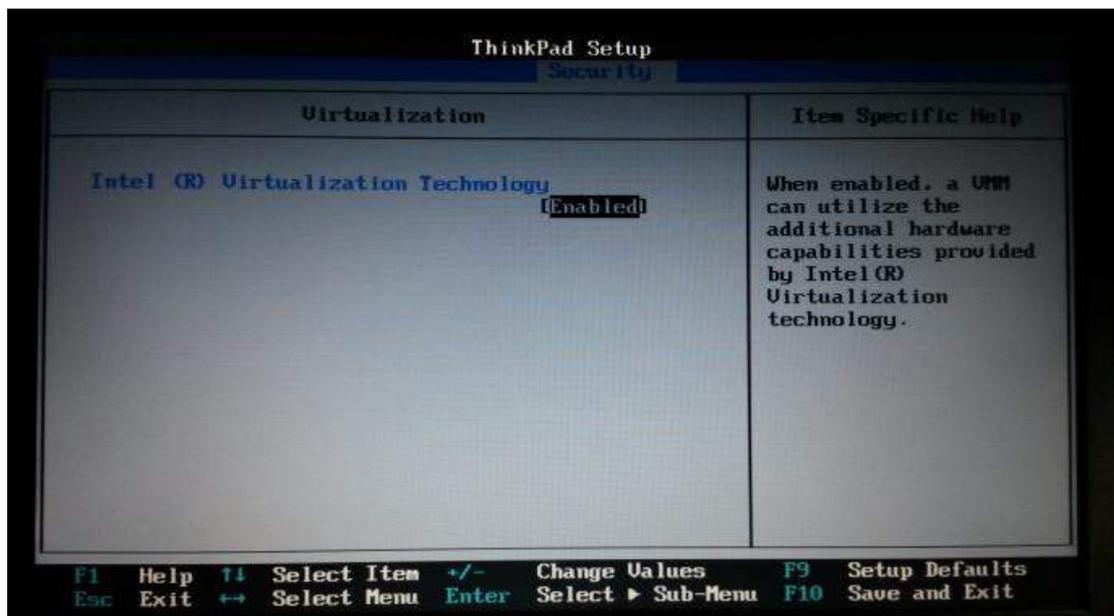




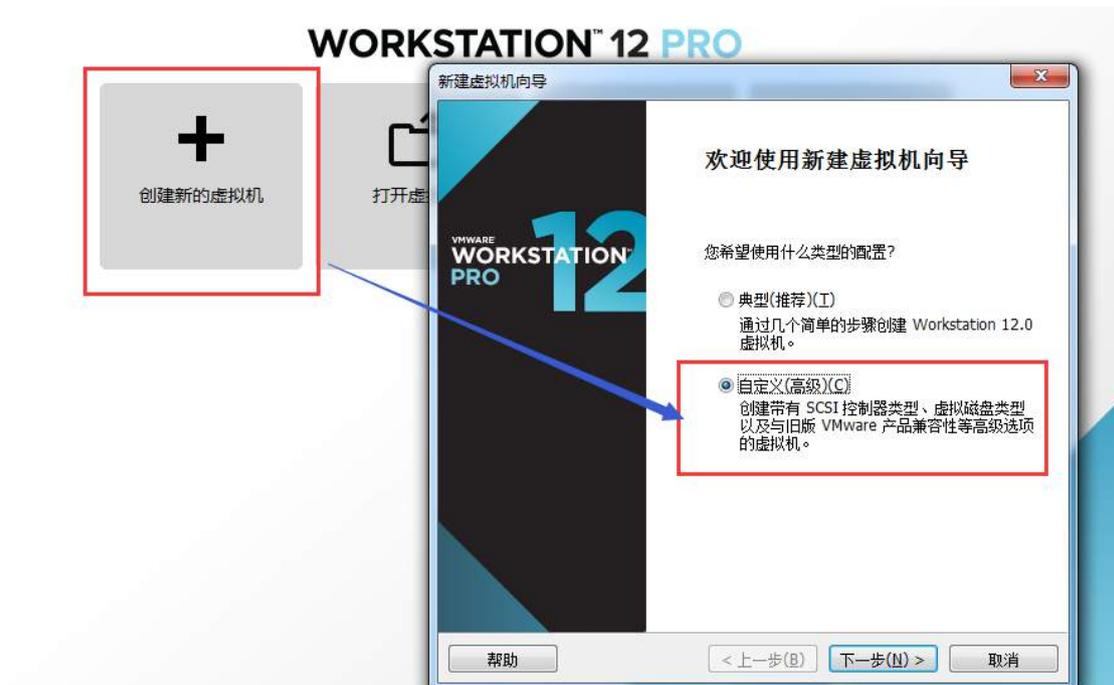


## 2.2 CentOS 安装

### 1 检查 BIOS 虚拟化支持



### 2 新建虚拟机



### 3 新建虚拟机向导



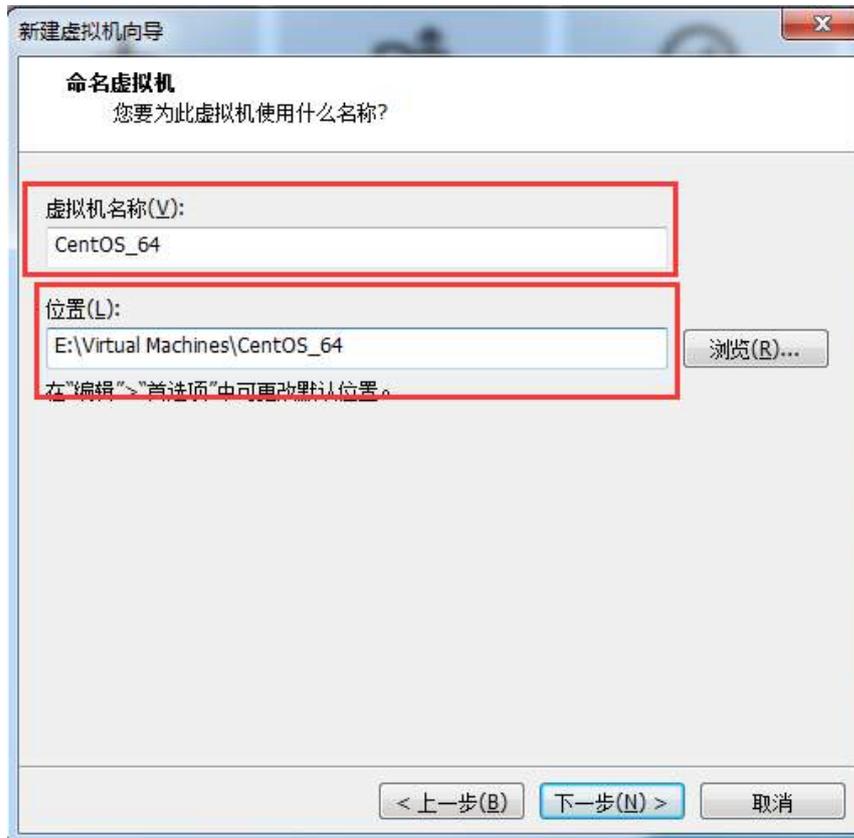
## 4 创建虚拟空白光盘



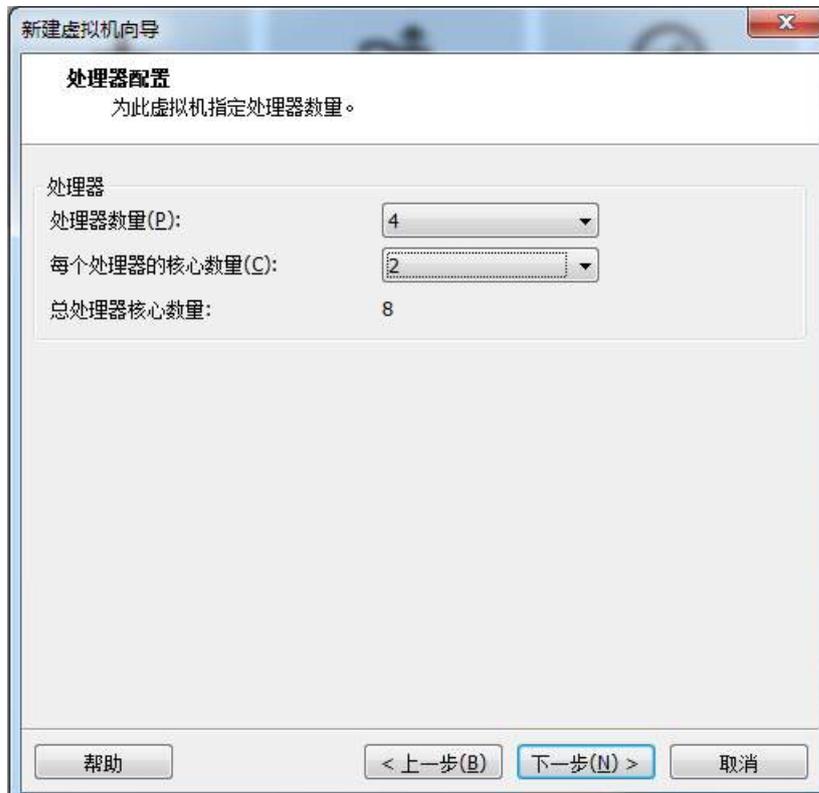
## 5 安装 Linux 系统对应的 CentOS 版



## 6 虚拟机命名和定位磁盘位置



## 7 处理器配置，看自己是否是双核、多核



## 8 设置内存为 2GB



## 9 网络设置 NAT

## 10 选择 IO 控制器类型



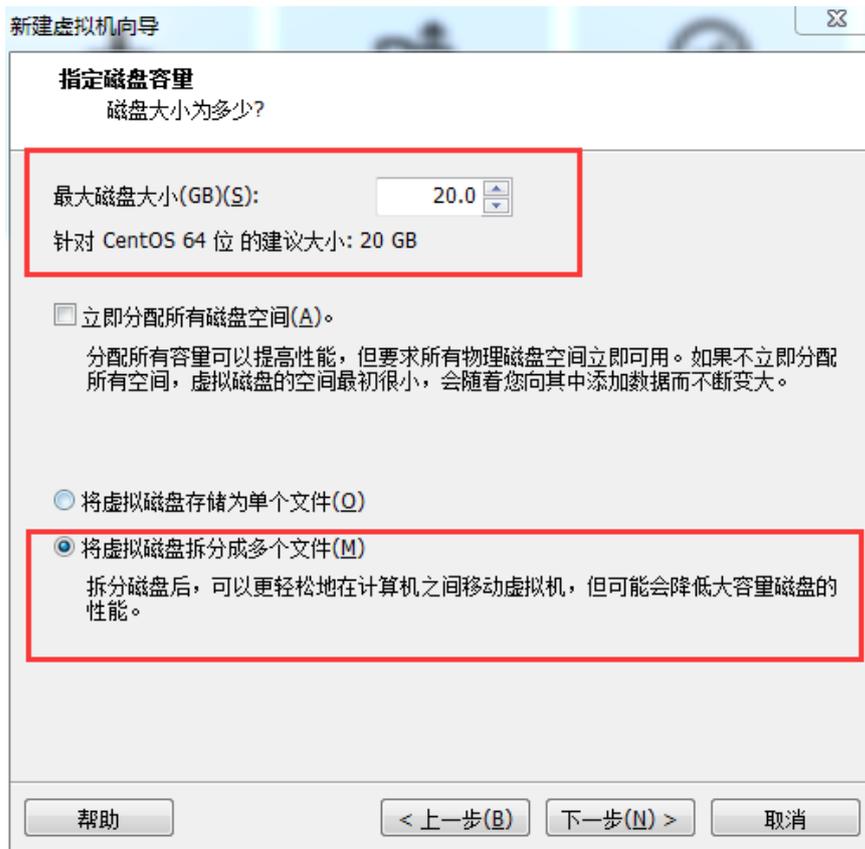
## 11 选择磁盘类型



## 12 新建虚拟磁盘



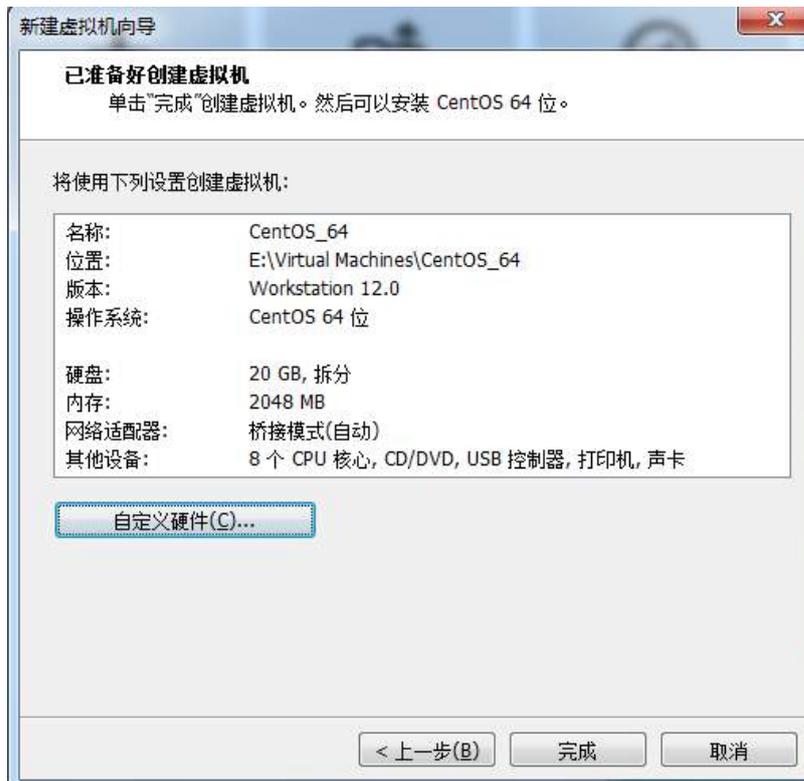
## 13 设置磁盘容量



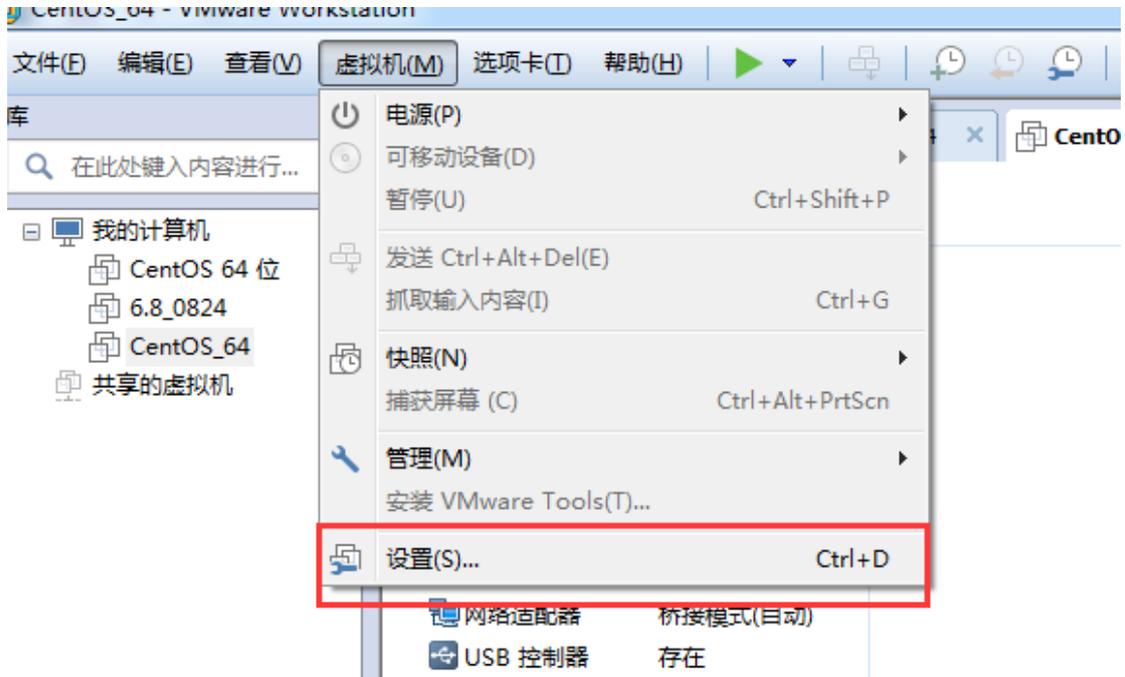
## 14 你在哪里存储这个磁盘文件



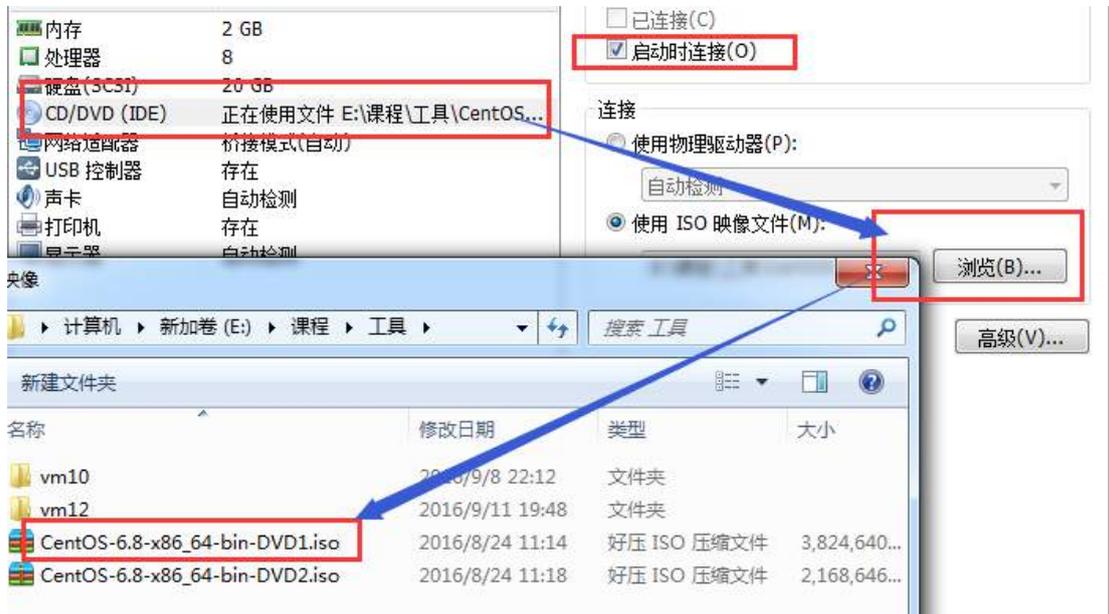
## 15 新建虚拟机向导配置完成



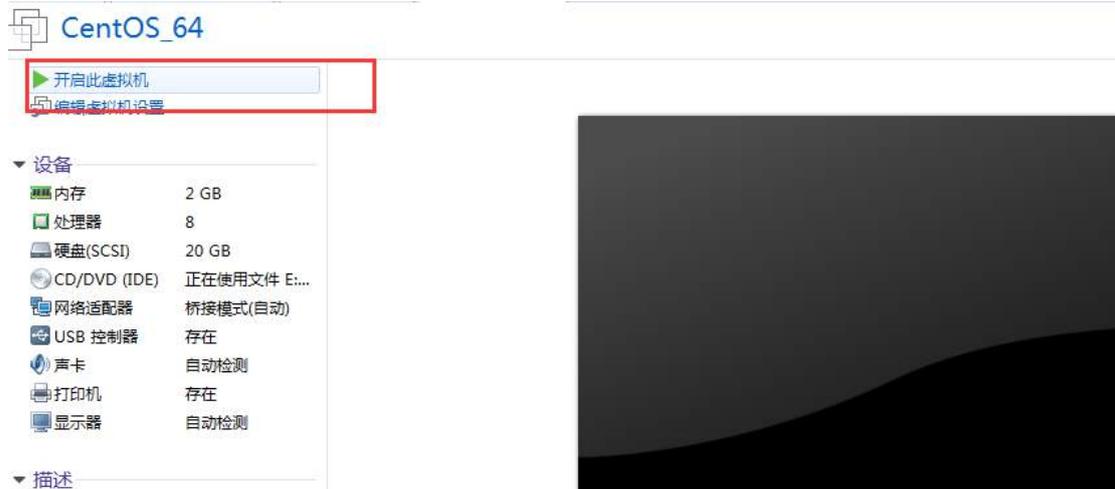
## 16 VM 设置



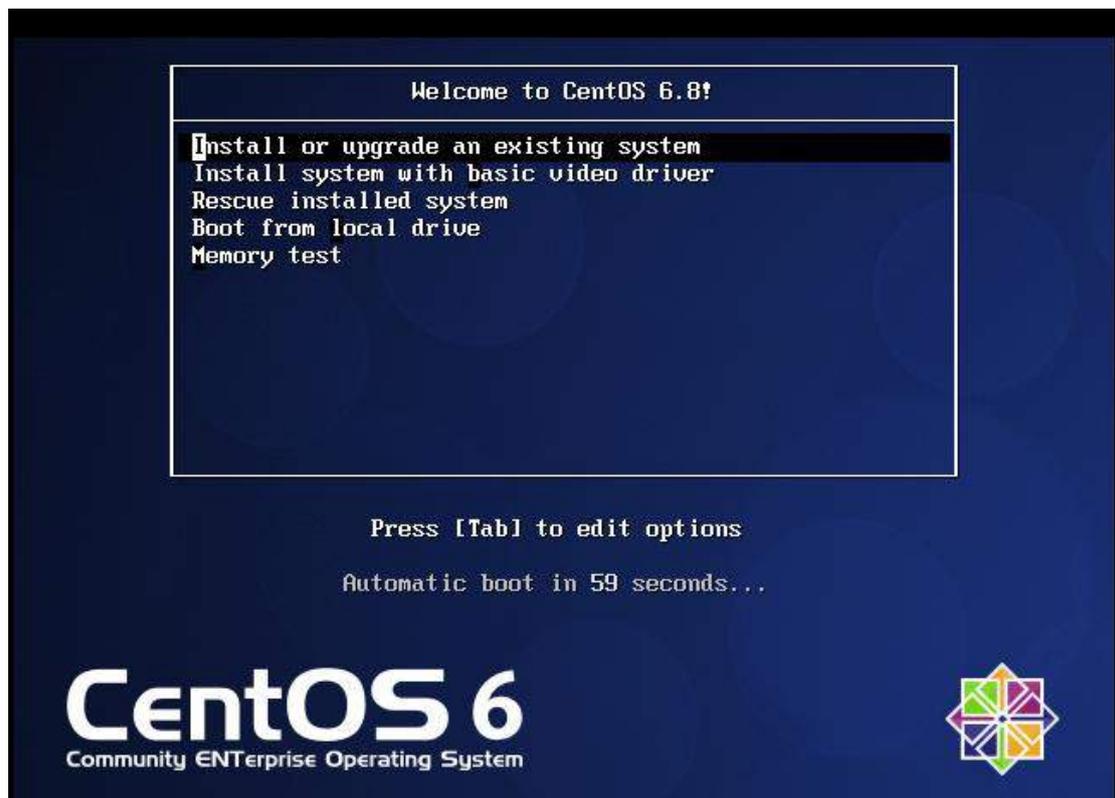
## 17 加载 ISO



## 18 加电并安装配置 CentOS



## 19 加电后初始化欢迎进入页面



回车选择第一个开始安装配置，此外，在 Ctrl+Alt 可以实现 Windows 主机和 VM 之间窗口的切换

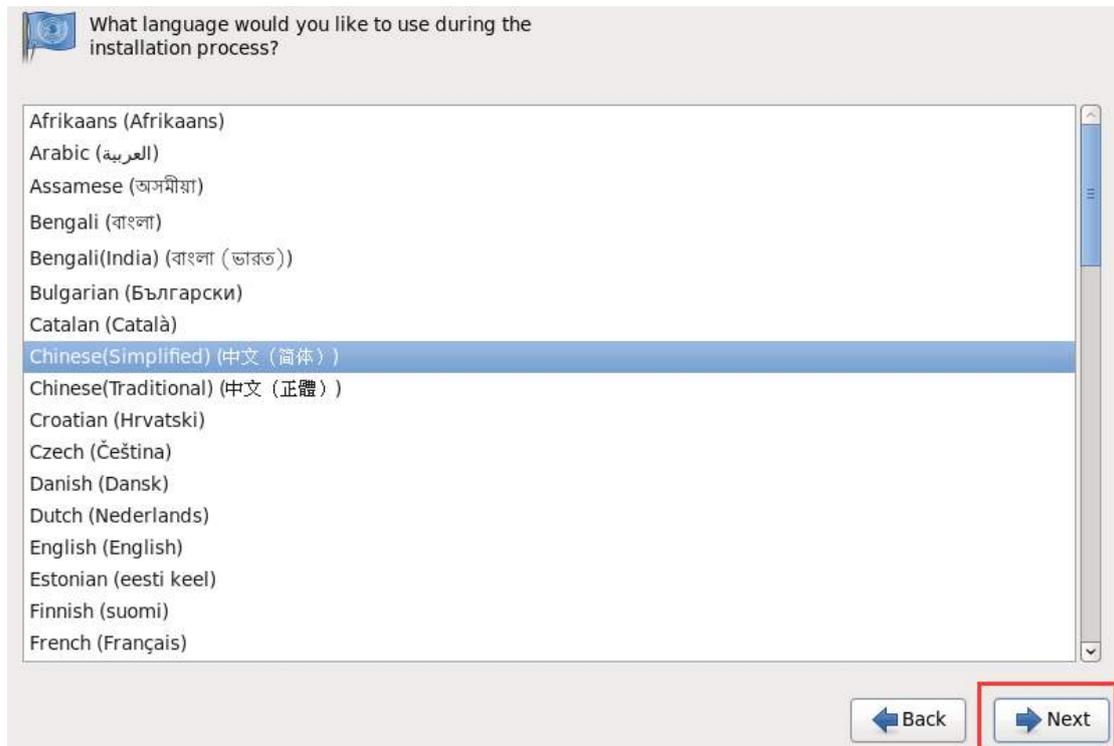
## 20 是否对 CD 媒体进行测试，直接跳过 Skip



## 21 CentOS 欢迎页面，直接点击 Next



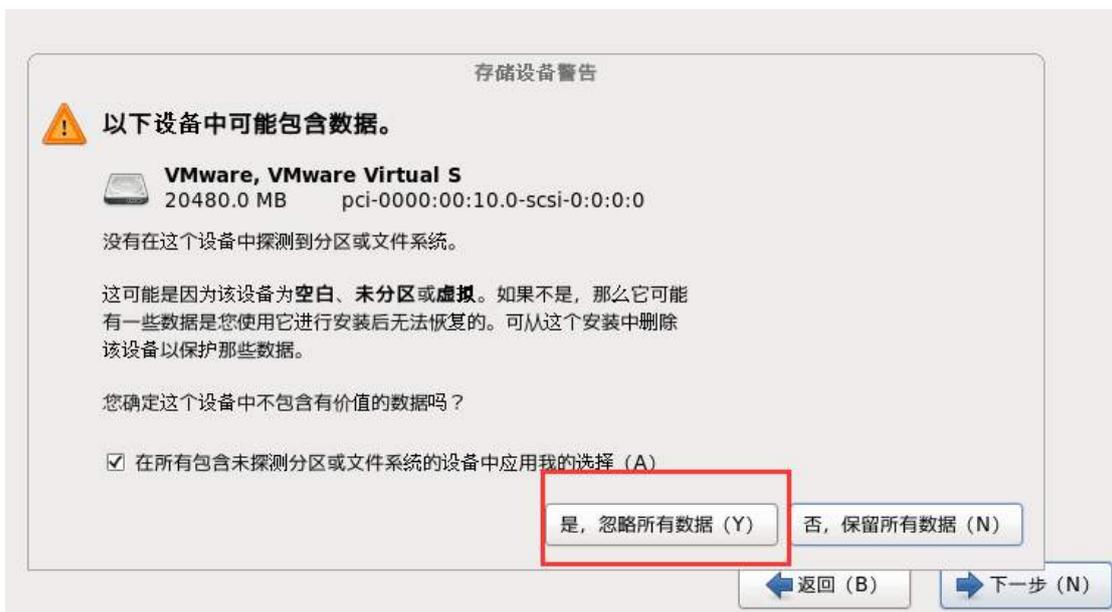
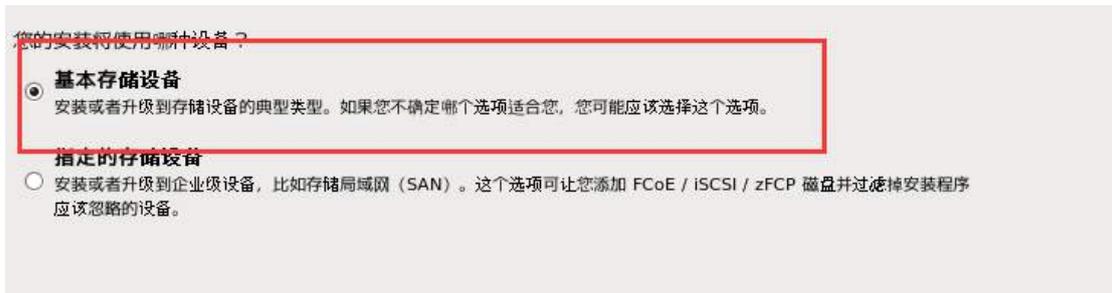
## 22 选择简体中文进行安装



## 23 选择语言键盘



## 24 选择存储设备



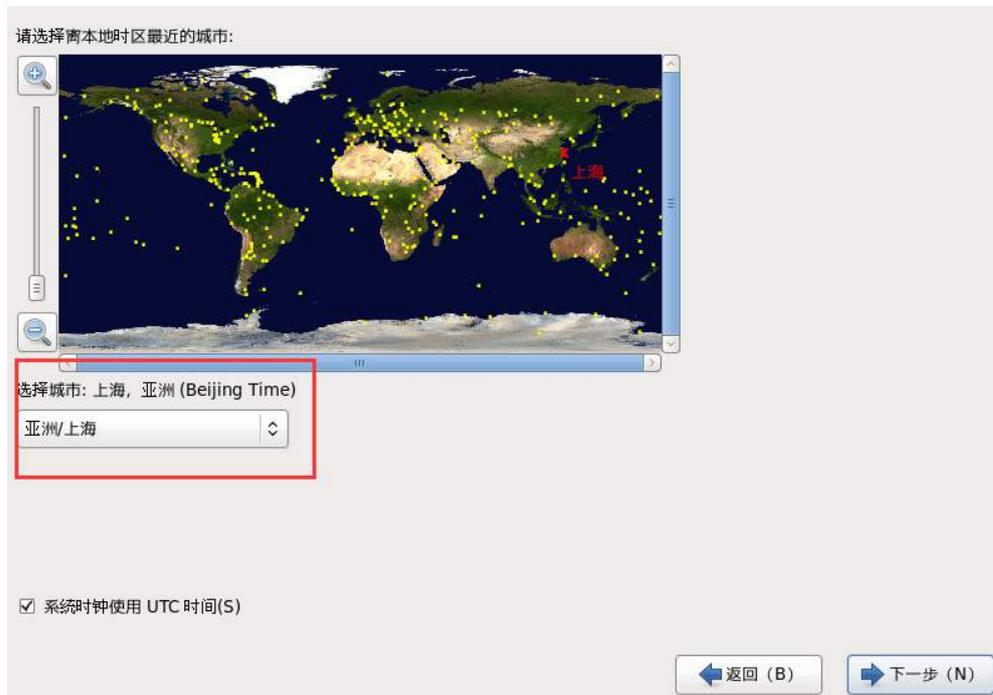
## 25 给计算机起名



## 26 设置网络环境

安装成功后再设置。

## 27 选择时区



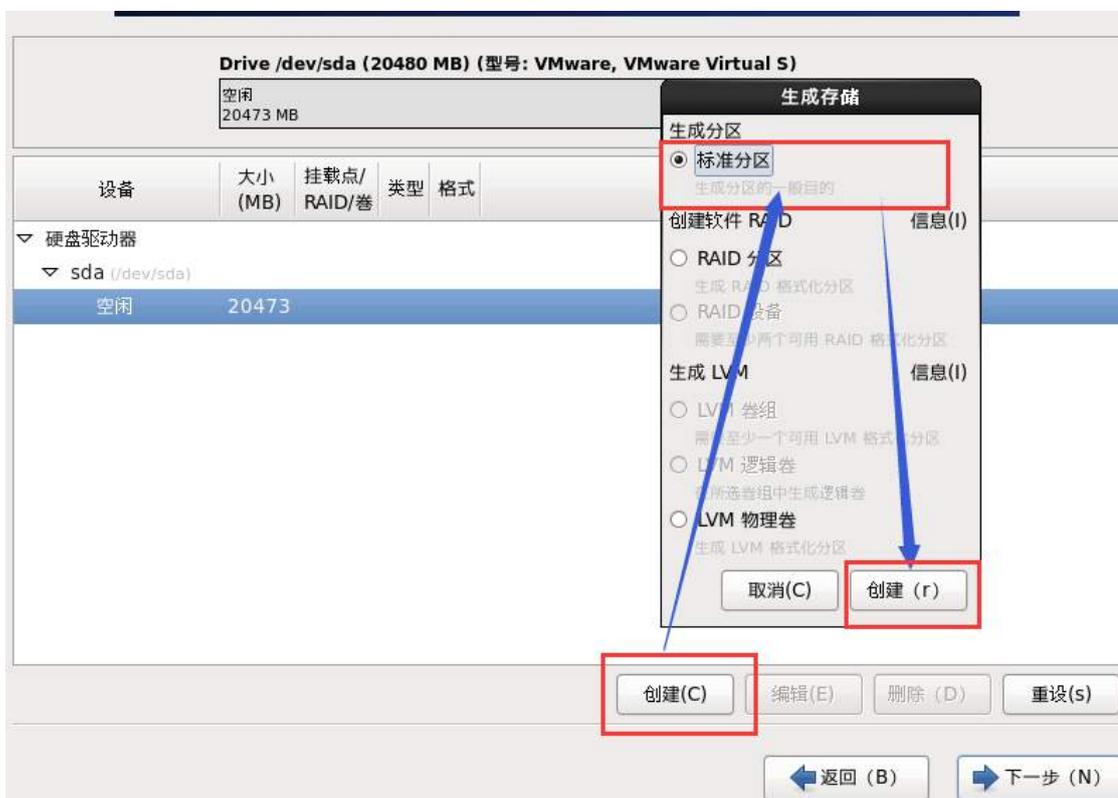
## 28 设置 root 密码 (一定记住)



## 29 硬盘分区-1



## 30 根分区新建





● Boot





- swap 分区设置



添加分区

挂载点(M) : <不适用>

文件系统类型(T) : swap

Drive	Size	Model
<input checked="" type="checkbox"/> sda	20480 MB	VMware, VMware Virtual S

允许的驱动器(D) :

大小(MB)(S) : 2048

其它大小选项

固定大小(F)

指定空间大小(MB)(u) : 1

使用全部可用空间(a)

强制为主分区(p)

加密 (E)

取消(C) 确定(O)

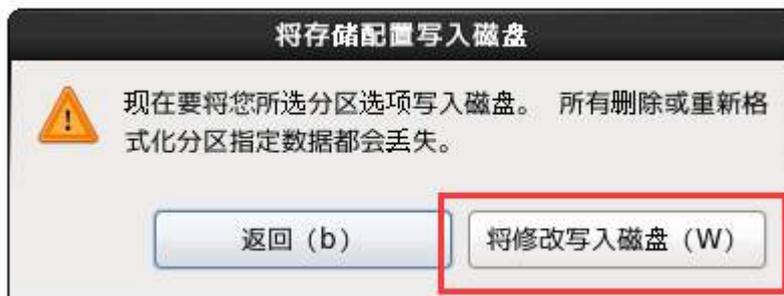
- 分区完成

请选择源驱动器

设备	大小 (MB)	挂载点/ RAID/卷	类型	格式
▼ 硬盘驱动器				
▼ sda (/dev/sda)				
sda1	100	/boot	ext4	✓
sda2	15360	/	ext4	✓
sda3	2048		swap	✓
空闲	2971			

创建(C) 编辑(E) 删除(D) 重设(s)

返回(B) 下一步(N)



## 31 程序引导，直接下一步

在 /dev/sda 中安装引导装载程序 (I)。

使用引导装载程序密码(U)

引导装载程序操作系统列表

默认	标签	设备
<input checked="" type="radio"/>	CentOS 6	/dev/sda2

## 32 现在定制系统软件

CentOS 默认安装是最小安装。您现在可以选择一些另外的软件。

Desktop  
 Minimal Desktop  
 Minimal  
 Basic Server  
 Database Server  
 Web Server  
 Virtual Host  
 Software Development Workstation

请选择您的软件安装所需要的存储库。

CentOS

或者。

以后自定义 (I)  现在自定义 (C)

### 33 Web 环境



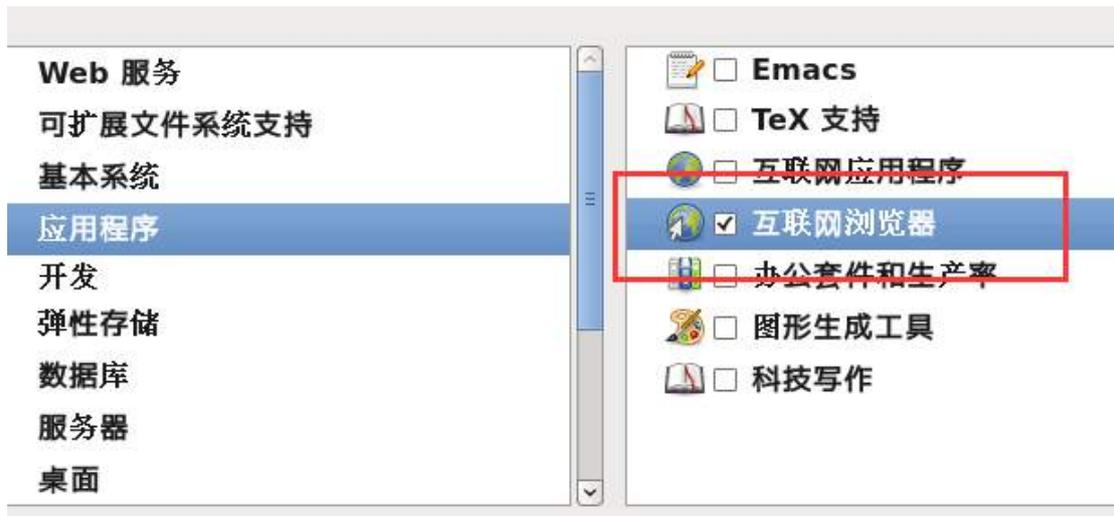
### 34 可扩展文件系统支持



### 35 基本系统

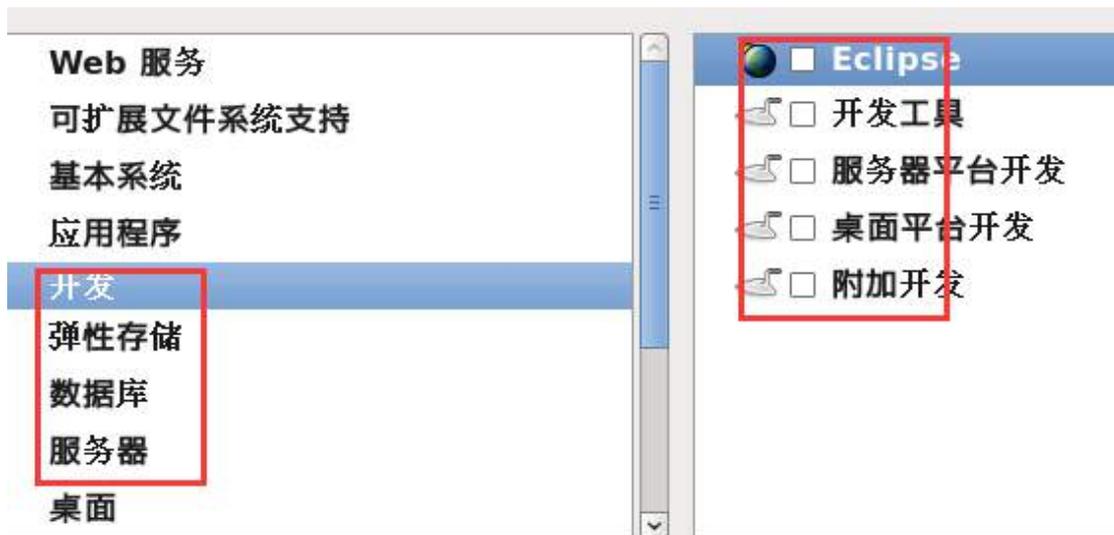


## 36 应用程序



## 37 开发、弹性存储、数据库、服务器

可以都不勾，有需要，以后使用中有需要再手动安装

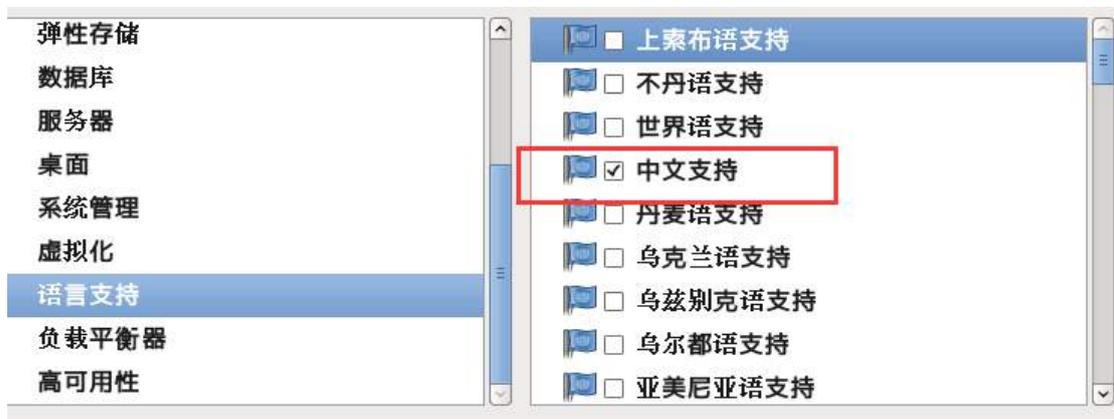


## 38 桌面

除了 KDE，其他都选就可以了。



### 39 语言支持



40 系统管理、虚拟化、负载均衡器、高可用性可以都不选

41 完成配置，开始安装 CentOS



## 42 等待安装完成，等待等待等待等待.....20 分钟左右



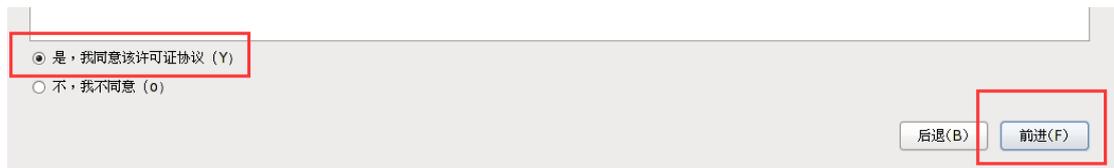
## 43 安装完成，重新引导



## 44 欢迎引导页面



## 45 许可证



## 46 创建用户，可以先不创建，用 root 账户登录就行

# 创建用户

您必须为您的系统创建一个常规使用的（非管理）‘用户名’。要创建系统‘用户名’，请提供以下所需信息。

用户名 (U) :

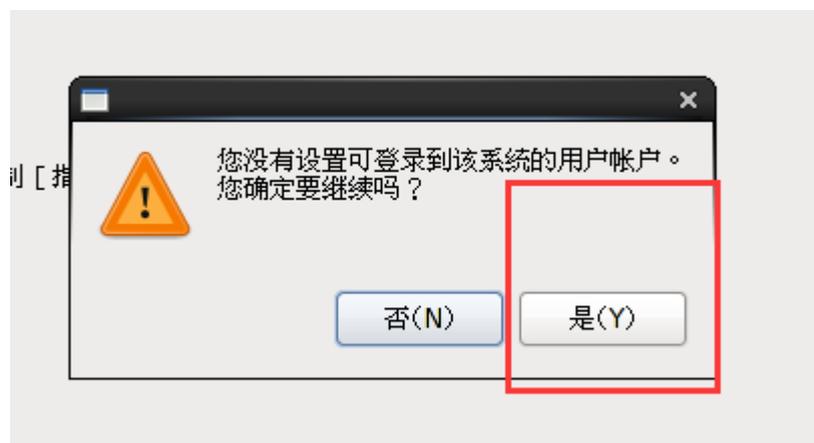
全名 (e) :

密码 (P) :

确认密码 (m) :

如果您需要使用网络验证，比如 Kerberos 或者 NIS，请点击“使用网络登录”按钮。

如果您需要在创建该用户时有更多控制 [指定主目录和 (/或者) UID]，请点击高级按钮。



## 47 时间和日期

### 日期和时间

请为系统设置日期和时间。

日期和时间 (T)

当前日期和时间 2016年08月30日 星期二 19时56分51秒

在网络上同步日期和时间 (y)

手动设置您系统的日期和时间

日期 (D)	时间																																																	
<input type="text" value="2016"/> < 2016 > < 八月 >	时 (H) : <input type="text" value="19"/>																																																	
<table border="1"><thead><tr><th>日</th><th>一</th><th>二</th><th>三</th><th>四</th><th>五</th><th>六</th></tr></thead><tbody><tr><td>31</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td></tr><tr><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr><tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td></tr><tr><td>28</td><td>29</td><td>30</td><td>31</td><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr></tbody></table>	日	一	二	三	四	五	六	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	分 (M) : <input type="text" value="51"/>
日	一	二	三	四	五	六																																												
31	1	2	3	4	5	6																																												
7	8	9	10	11	12	13																																												
14	15	16	17	18	19	20																																												
21	22	23	24	25	26	27																																												
28	29	30	31	1	2	3																																												
4	5	6	7	8	9	10																																												
	秒 (S) : <input type="text" value="49"/>																																																	

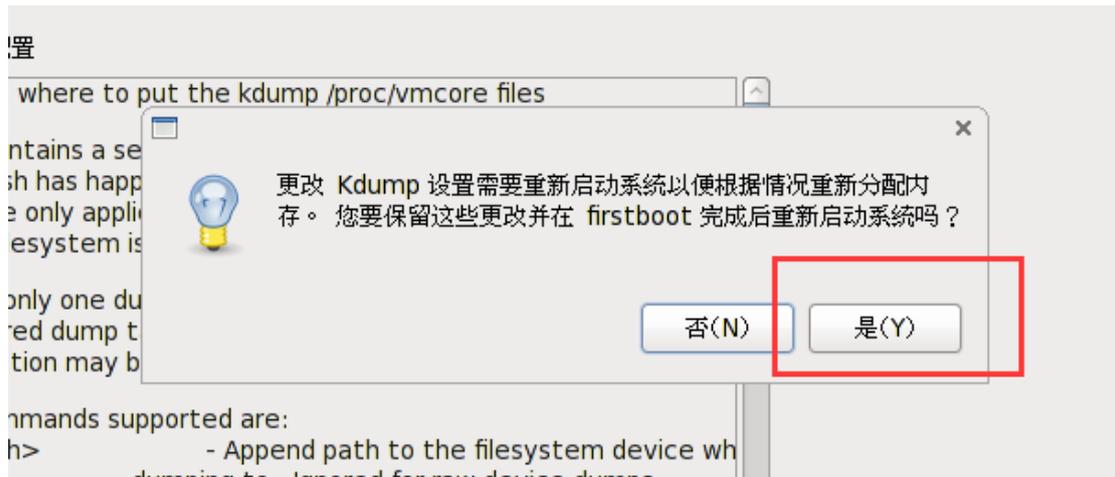
## 48 Kdump, 去掉

### Kdump

Kdump 是一个内核崩溃转储机制。在系统崩溃的时候，kdump 将捕获系统信息，这对于诊断崩溃的原因非常有用。注意，kdump 需要预留一部分系统内存，且这部分内存对于其他用户是不可用的。

启用 kdump (E) ?

总共系统内存 (MB) :	1990
Kdump 内存 (MB) :	<input type="text" value="128"/>
可用系统内存 (MB) :	1862



## 49 重启后用 root 登录

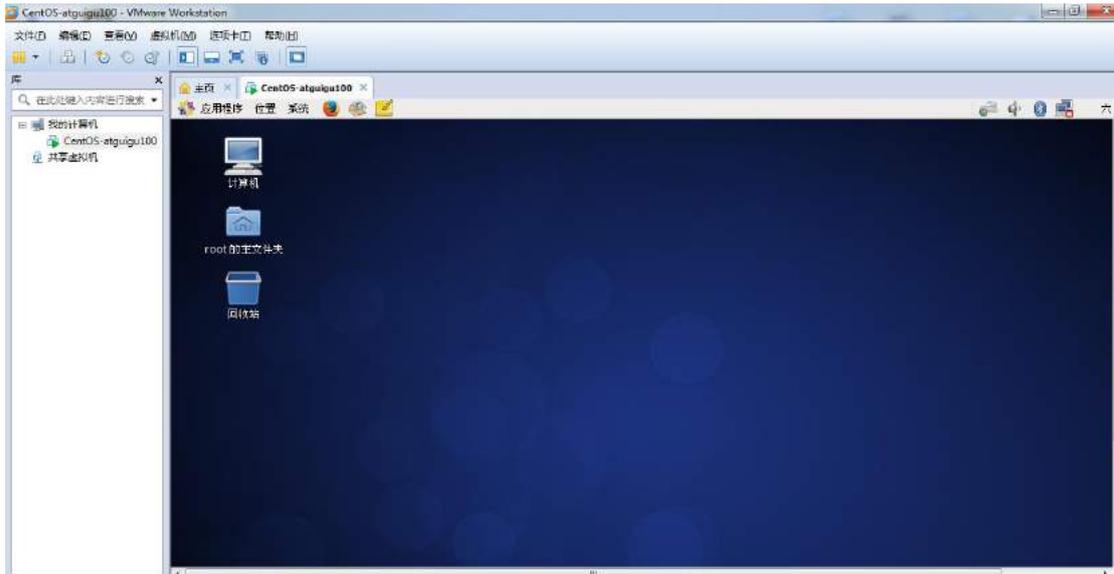


## 2.3 VMTools 安装

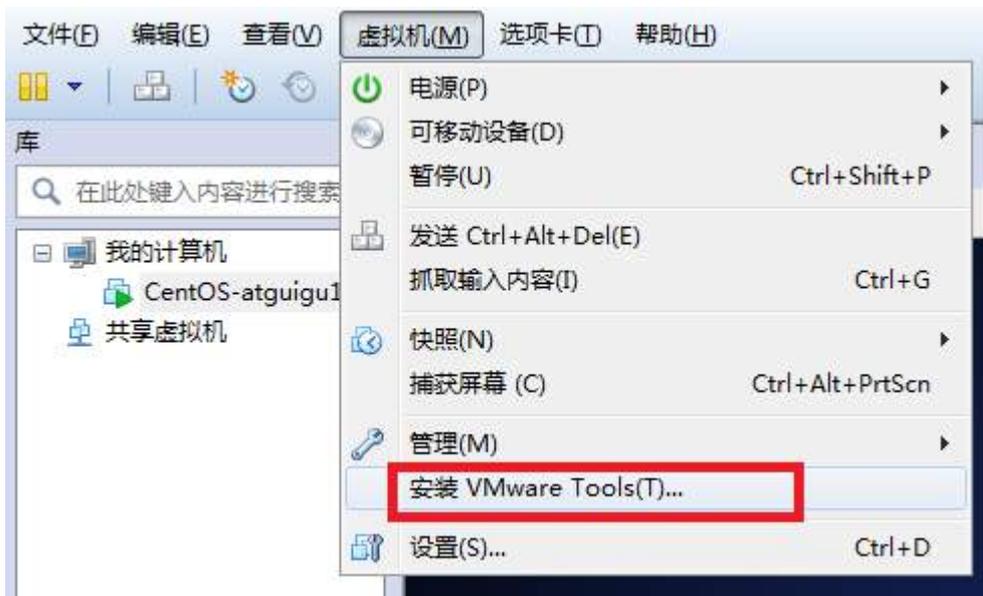
### 1) 什么是 VMtools

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。

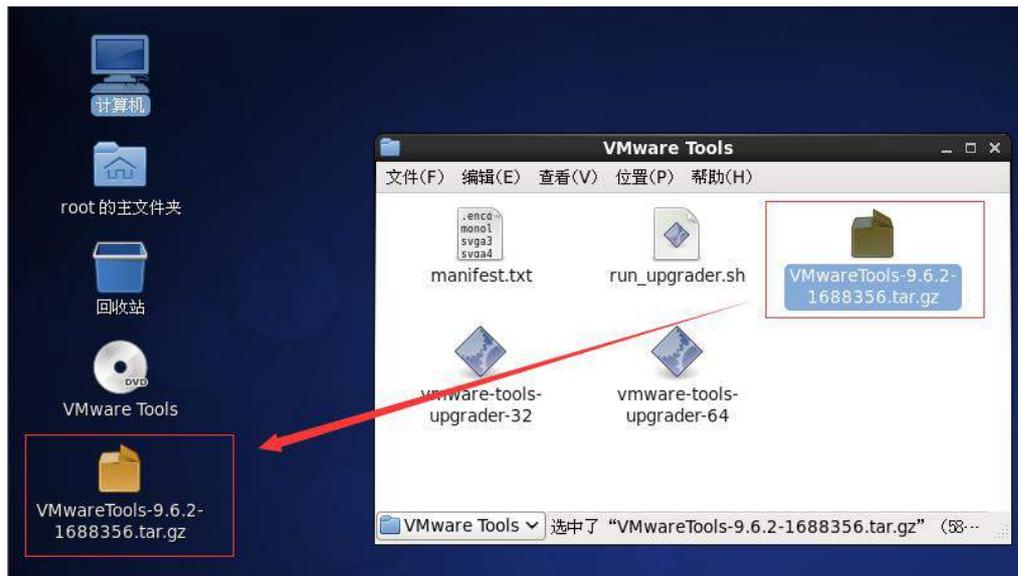
### 2) 先启动 CentOS 并成功登录如下图，发现底部提示且窗口中等大小，准备安装



3) 选择虚拟机菜单栏--安装 VMware tools



4) 光驱自动挂载 VMTools



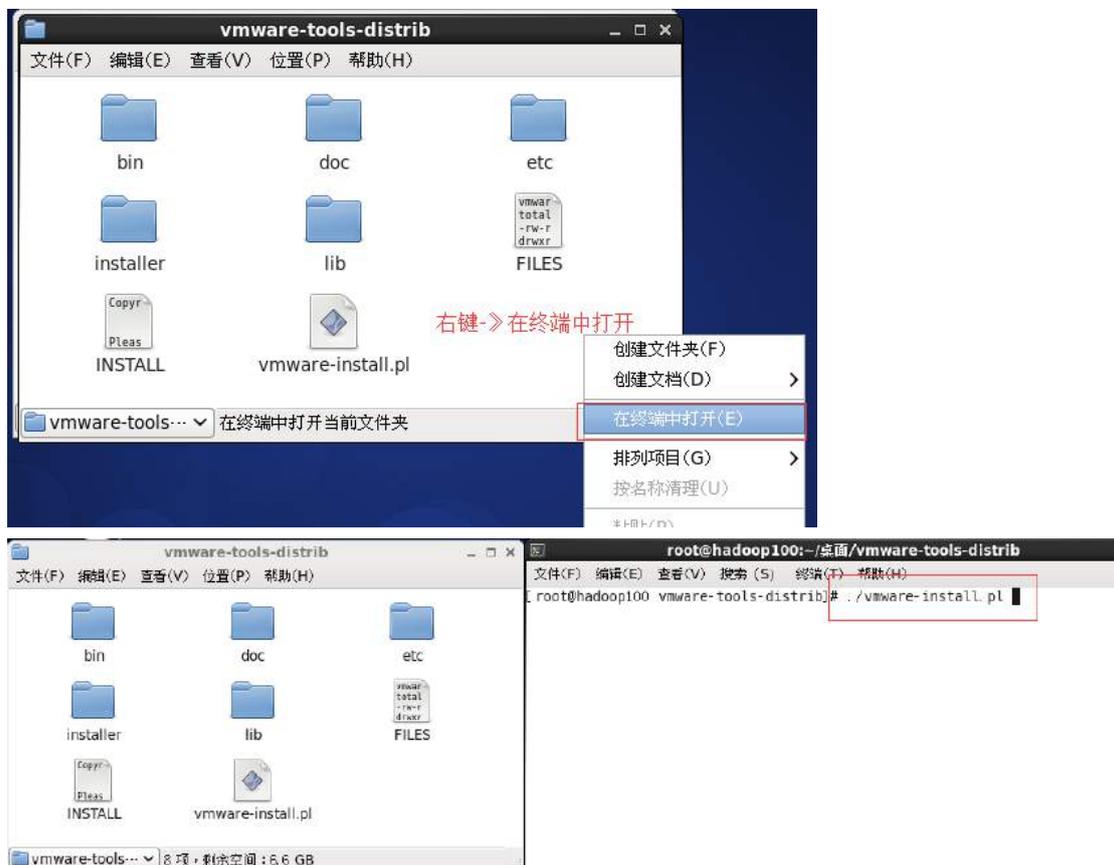
5) 右键解压 VMwaretools-9.6.2-1688356.tar.gz

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.

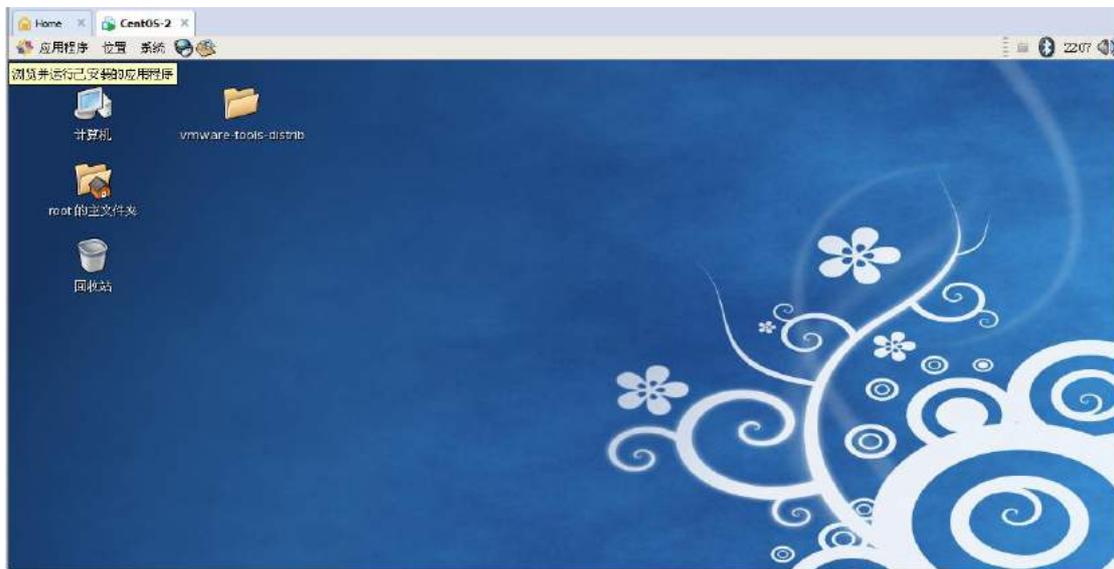
to use the vmxnet driver, either reboot or
1. stop networking or stop any interface using the vmxnet or pcnet32 driver
2. remove the pcnet32 module with 'rmmod pcnet32'
3. remove the vmxnet module with 'rmmod vmxnet'
4. load the vmxnet module with 'modprobe -v vmxnet'
5. and restart networking or restart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

Enjoy,

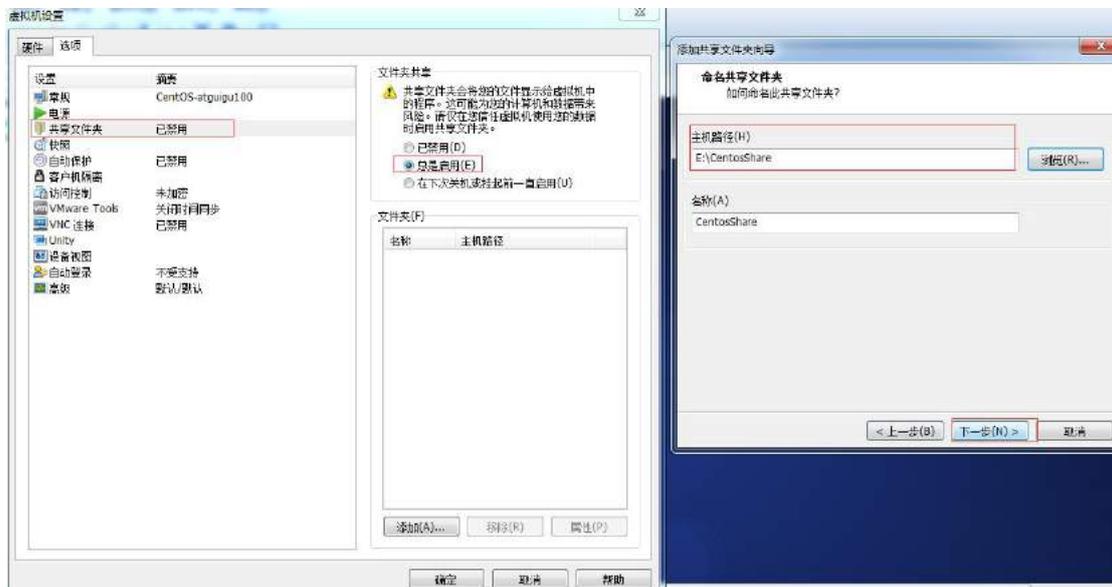
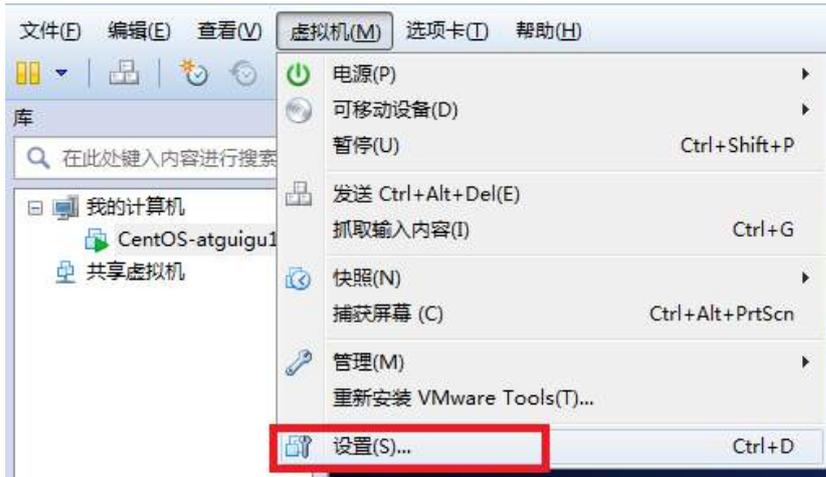
--the VMware team

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



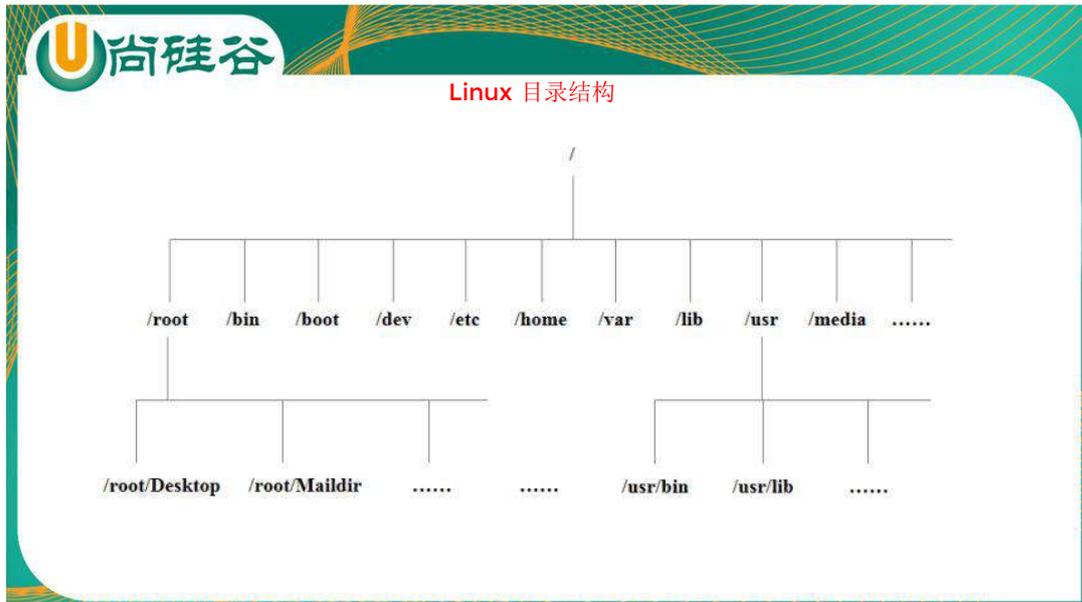
```
root@atguigu:/mnt/hgfs/centos2-share-window
文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)
[root@atguigu centos2-share-window]# pwd
/mnt/hgfs/centos2-share-window
[root@atguigu centos2-share-window]# ls -ll
总计 1
-rwxrwxrwx 1 root root 17 10-19 22:16 abc.txt
[root@atguigu centos2-share-window]# more abc.txt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

## 第 3 章 Linux 文件与目录结构

### 3.1 Linux 文件

Linux 系统中一切皆文件。

### 3.2 Linux 目录结构



### Linux 目录结构

- **/bin**
  - 是Binary的缩写,这个目录存放着最经常使用的命令
- **/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。
- **/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。
- **/root**
  - 该目录为系统管理员,也称作超级权限者的用户主目录。



### Linux 目录结构

- /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。
- /etc
  - 所有的系统管理所需要的配置文件和子目录。
- /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，类似于windows下的program files目录。



### Linux 目录结构

- /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。
- /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。
- /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。



### Linux 目录结构

- /tmp
  - 这个目录是用来存放一些临时文件的。
- /dev
  - 类似于windows的设备管理器，把所有的硬件用文件的形式存储。
- /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。
- /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，我们可以将外部的存储挂载在/mnt/上，然后进入该目录就可以查看里的内容了。



### Linux 目录结构

- /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mysql数据库则就可以放到这个目录下。默认是空的。
- /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。
- /selinux
  - SELinux是一种安全子系统,它能控制程序只能访问特定文件。

## 第 4 章 VI/VIM 编辑器

### 4.1 是什么

VI 是 Unix 操作系统和类 Unix 操作系统中最通用的文本编辑器。

VIM 编辑器是从 VI 发展出来的一个性能更强大的文本编辑器。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 与 VI 编辑器完全兼容。

### 4.2 测试数据准备

1) 拷贝/etc/smartd.conf 数据到/root 目录下

```
[root@hadoop100 桌面]# cp /etc/smartd.conf /root
```

```
[root@hadoop100 桌面]# cd /root/
```

### 4.3 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

语法	功能描述
yy	复制光标当前一行
y 数字 y	复制一段（从第几行到第几行）
p	箭头移动到目的行粘贴
u	撤销上一步
dd	删除光标当前行
d 数字 d	删除光标（含）后多少行
x	删除一个字母，相当于 del
X	删除一个字母，相当于 Backspace
yw	复制一个词
dw	删除一个词
shift+^	移动到行头
shift+\$	移动到行尾

l+shift+g	移动到页头，数字
shift+g	移动到页尾
数字 N+shift+g	移动到目标行



## 4.4 编辑模式

在一般模式中可以进行删除、复制、粘贴等的动作，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

按键	功能
i	当前光标前
a	当前光标后
o	当前光标行的下一行
I	光标所在行最前
A	光标所在行最后

O	当前光标行的上一行
---	-----------

2) 退出编辑模式

按『Esc』键

## 4.5 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！

1) 基本语法

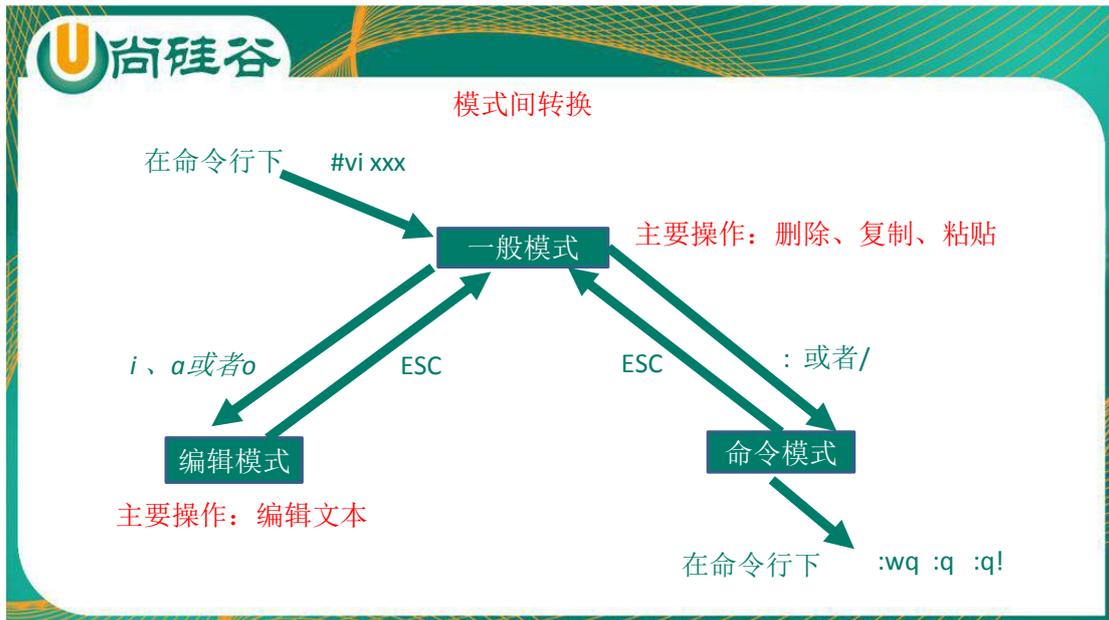
命令	功能
:w	保存
:q	退出
:!	强制执行
/要查找的词	n 查找下一个，N 往上查找
? 要查找的词	n是查找上一个，shift+n是往下查找
:set nu	显示行号
:set nonu	关闭行号

2) 案例实操

(1) 强制保存退出

:wq!

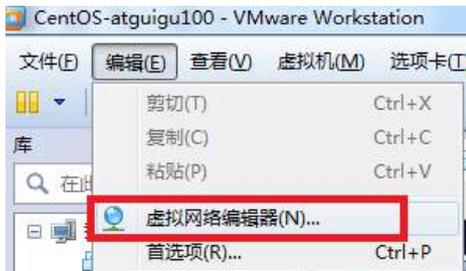
### 4.6 模式间转换



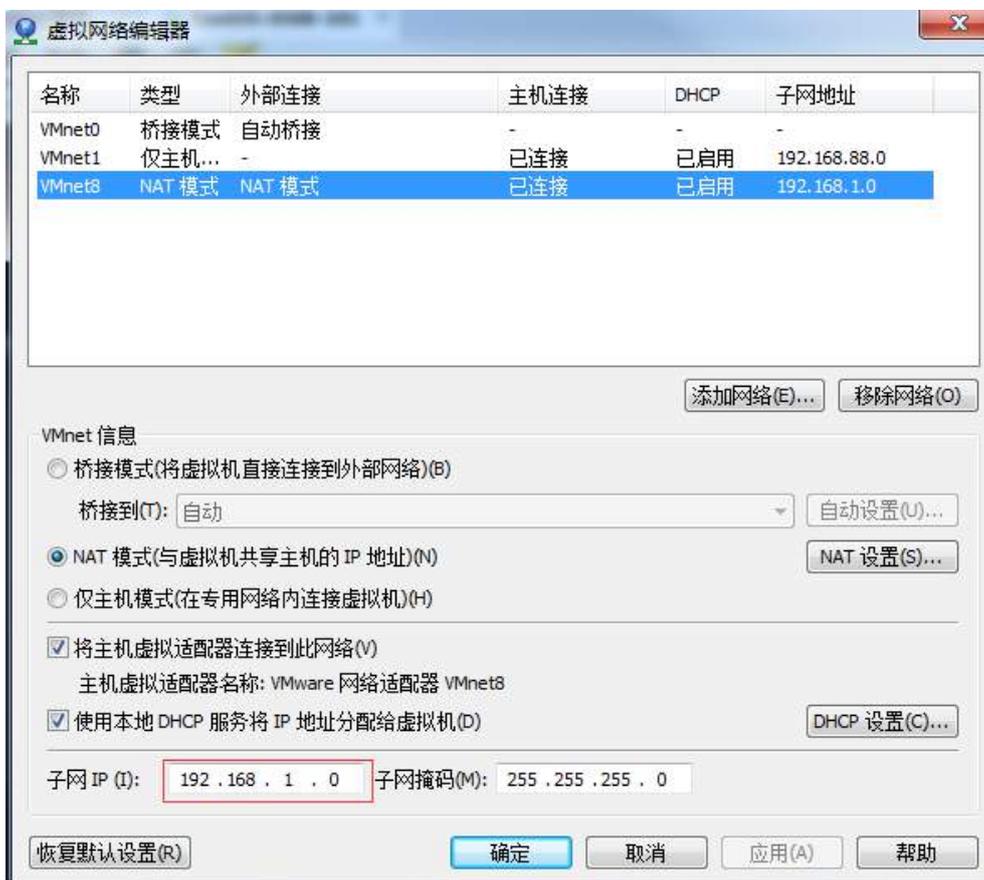
## 第 5 章 网络配置和系统管理操作

### 5.1 查看网络 IP 和网关

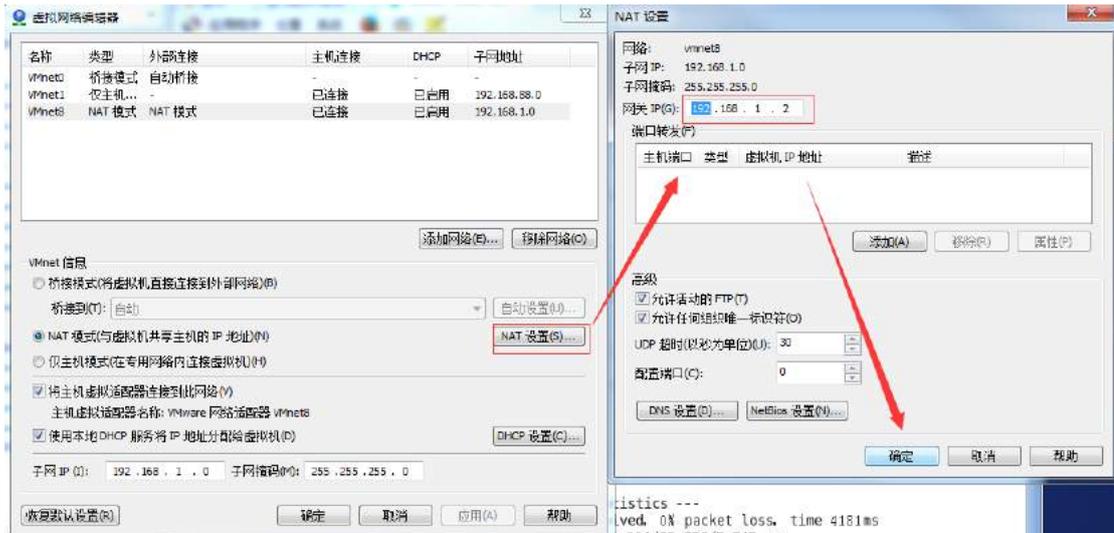
1) 查看虚拟网络编辑器



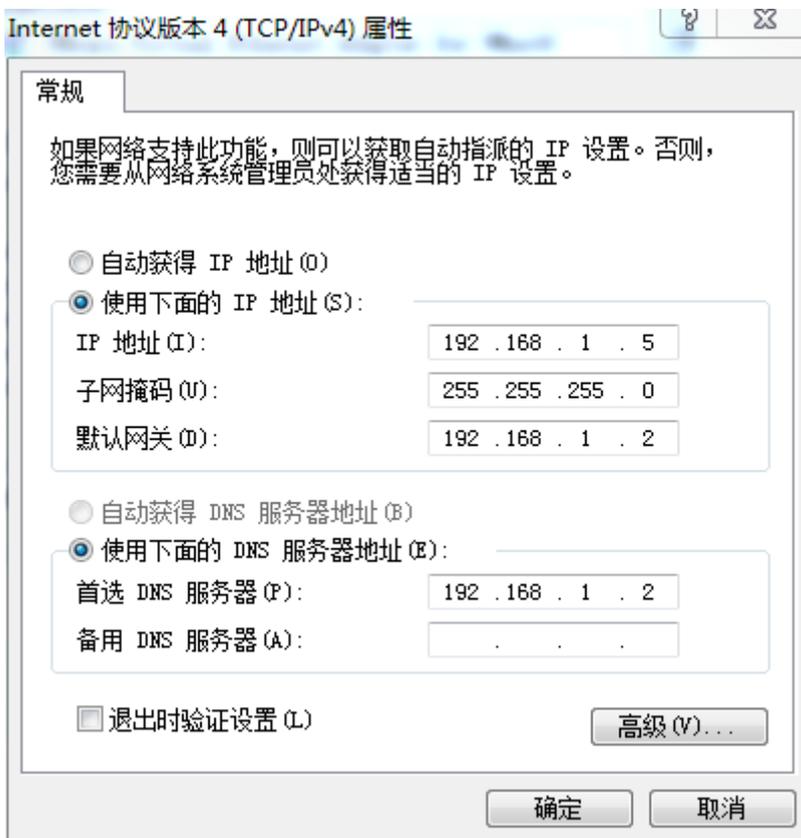
2) 修改 ip 地址



3) 查看网关



4) 查看 windows 环境中 VMnet8 网络配置



## 5.2 配置网络 ip 地址

### 5.2.1 ifconfig 配置网络接口

ifconfig :network interfaces configuring 网络接口配置

1) 基本语法:

ifconfig (功能描述: 显示所有网络接口的配置信息)

更多 Java - 大数据 - 前端 - python 人工智能资料下载, 可百度访问: 尚硅谷官网

## 2) 案例实操:

(1) 查看当前网络 ip

```
[root@hadoop100 桌面]# ifconfig
```

## 5.2.2 ping 测试主机之间网络连通性

## 1) 基本语法:

ping 目的主机 (功能描述: 测试当前服务器是否可以连接目的主机)

## 2) 案例实操:

(1) 测试当前服务器是否可以连接百度

```
[root@hadoop100 桌面]# ping www.baidu.com
```

## 5.2.3 修改 IP 地址

## 1) 修改 IP 地址

```
[root@hadoop100 桌面]# vim /etc/sysconfig/network-scripts/ifcfg-eth0
```



```
root@hadoop100:~/桌面
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
DEVICE=eth0
HWADDR=00:0C:29:65:62:6A
TYPE=Ethernet
UUID=926a57ba-92c6-4231-bacb-f27e5e6a9f44
ONBOOT=no
NM_CONTROLLED=yes
BOOTPROTO=dhcp
```

以下标红的项必须修改, 有值的按照下面的值修改, 没有该项的要增加。

DEVICE=eth0	#接口名 (设备,网卡)
HWADDR=00:0C:2x:6x:0x:xx	#MAC 地址
TYPE=Ethernet	#网络类型 (通常是 Ethemet)
UUID=926a57ba-92c6-4231-bacb-f27e5e6a9f44	#随机 id
ONBOOT=yes	#系统启动的时候网络接口是否有效 (yes/no)
BOOTPROTO=static	# IP 的配置方法[none static bootp dhcp] (引导时不使用协议 静态分配 IP BOOTP 协议 DHCP 协议)
	#IP 地址

```

IPADDR=192.168.1.101

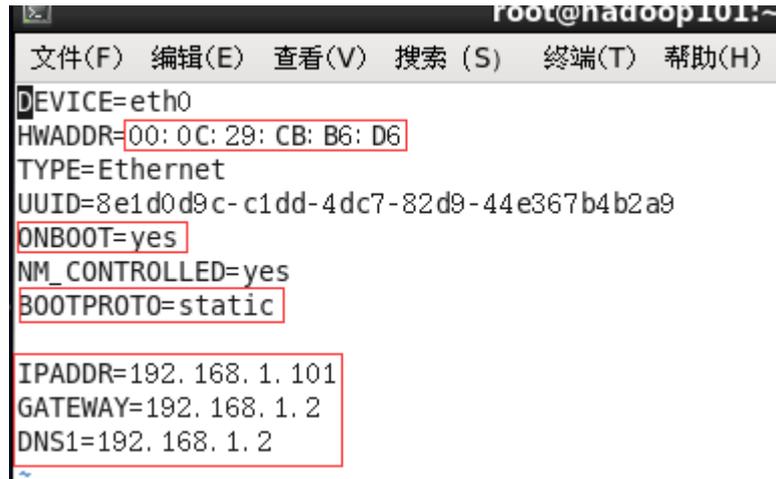
#网关

GATEWAY=192.168.1.2

#域名解析器

DNS1=192.168.1.2
    
```

修改后



```

root@hadoop101:~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
DEVICE=eth0
HWADDR=00:0C:29:CB:B6:D6
TYPE=Ethernet
UUID=8e1d0d9c-c1dd-4dc7-82d9-44e367b4b2a9
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static

IPADDR=192.168.1.101
GATEWAY=192.168.1.2
DNS1=192.168.1.2
    
```

: wq 保存退出

2) 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]
    
```

3) 如果报错, reboot, 重启虚拟机

## 5.3 配置主机名

### 5.3.1 hostname 显示和设置系统的主机名称

1) 基本语法:

hostname (功能描述: 查看当前服务器的主机名称)

2) 案例实操:

(1) 查看当前服务器主机名称

```
[root@hadoop100 桌面]# hostname
```

### 5.3.2 修改主机名称

1) 修改 linux 的主机映射文件 (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@hadoop100 桌面]# hostname
```

```
hadoop100
```

(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@hadoop100 桌面]# vi /etc/sysconfig/network
```

文件中内容

```
NETWORKING=yes
```

```
NETWORKING_IPV6=no
```

```
HOSTNAME= hadoop100
```

**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 hadoop100。

(4) 保存退出。

(5) 打开/etc/hosts

```
[root@hadoop100 桌面]# vim /etc/hosts
```

添加如下内容

```
192.168.1.100 hadoop100
```

```
192.168.1.101 hadoop101
```

```
192.168.1.102 hadoop102
```

```
192.168.1.103 hadoop103
```

```
192.168.1.104 hadoop104
```

```
192.168.1.105 hadoop105
```

```
192.168.1.106 hadoop106
```

```
192.168.1.107 hadoop107
```

```
192.168.1.108 hadoop108
```

(6) 并重启设备，重启后，查看主机名，已经修改成功

2) 修改 window7 的主机映射文件（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

```
192.168.1.100 hadoop100
192.168.1.101 hadoop101
192.168.1.102 hadoop102
192.168.1.103 hadoop103
192.168.1.104 hadoop104
192.168.1.105 hadoop105
192.168.1.106 hadoop106
192.168.1.107 hadoop107
192.168.1.108 hadoop108
```

3) 修改 window10 的主机映射文件 (hosts 文件)

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

(2) 拷贝 hosts 文件到桌面

(3) 打开桌面 hosts 文件并添加如下内容

```
192.168.1.100 hadoop100
192.168.1.101 hadoop101
192.168.1.102 hadoop102
192.168.1.103 hadoop103
192.168.1.104 hadoop104
192.168.1.105 hadoop105
192.168.1.106 hadoop106
192.168.1.107 hadoop107
192.168.1.108 hadoop108
```

(4) 将桌面 hosts 文件覆盖 C:\Windows\System32\drivers\etc 路径 hosts 文件

## 5.4 关闭防火墙

### 5.4.1 service 后台服务管理

1) 基本语法:

service 服务名 start (功能描述: 开启服务)

service 服务名 stop (功能描述: 关闭服务)

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载, 可百度访问: [尚硅谷官网](#)

- service 服务名 restart (功能描述: 重新启动服务)
- service 服务名 status (功能描述: 查看服务状态)

## 2) 经验技巧

查看服务的方法: /etc/init.d/服务名

```
[root@hadoop100 init.d]# pwd
```

```
/etc/init.d
```

```
[root@hadoop100 init.d]# ls -al
```

## 3) 案例实操

- (1) 查看网络服务的状态

```
[root@hadoop100 桌面]#service network status
```

- (2) 停止网络服务

```
[root@hadoop100 桌面]#service network stop
```

- (3) 启动网络服务

```
[root@hadoop100 桌面]#service network start
```

- (4) 重启网络服务

```
[root@hadoop100 桌面]#service network restart
```

- (5) 查看系统中所有的后台服务

```
[root@hadoop100 桌面]#service --status-all
```

## 5.4.2 chkconfig 设置后台服务的自启配置

### 1) 基本语法:

chkconfig (功能描述: 查看所有服务器自启配置)

chkconfig 服务名 off (功能描述: 关掉指定服务的自动启动)

chkconfig 服务名 on (功能描述: 开启指定服务的自动启动)

chkconfig 服务名 --list (功能描述: 查看服务开机启动状态)

### 2) 案例实操

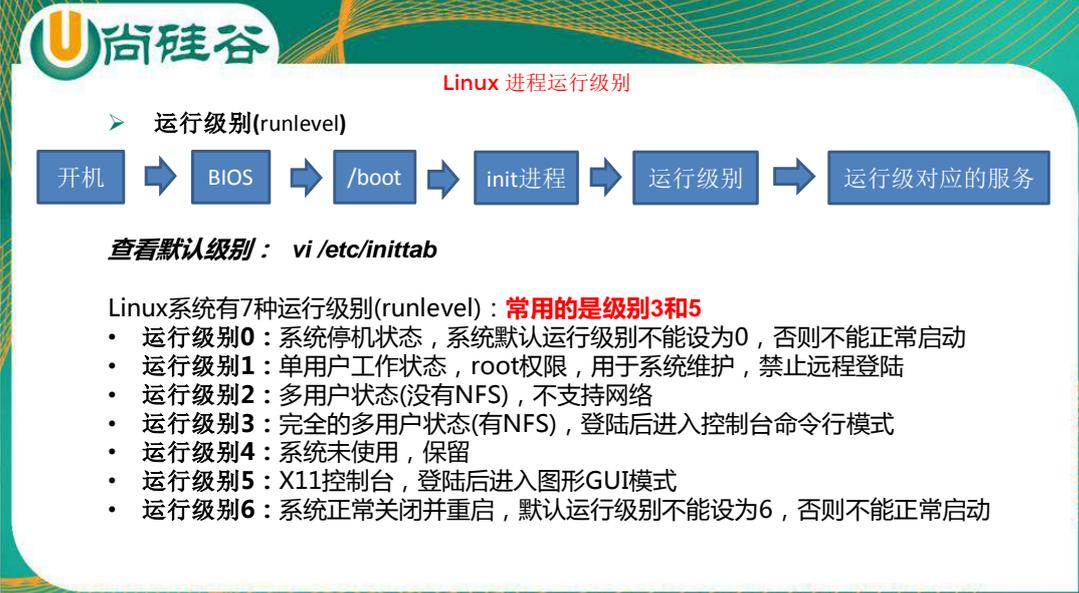
- (1) 关闭iptables服务的自动启动

```
[root@hadoop100 桌面]#chkconfig iptables off
```

- (2) 开启iptables服务的自动启动

```
[root@hadoop100 桌面]#chkconfig iptables on
```

### 5.4.3 进程运行级别



**Linux 进程运行级别**

➤ 运行级别(runlevel)

```
graph LR; A[开机] --> B[BIOS]; B --> C[/boot]; C --> D[init进程]; D --> E[运行级别]; E --> F[运行级对应的服务]
```

查看默认级别：`vi /etc/inittab`

Linux系统有7种运行级别(runlevel)：**常用的是级别3和5**

- 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)，不支持网络
- 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

### 5.4.4 关闭防火墙

#### 1) 临时关闭防火墙:

##### (1) 查看防火墙状态

```
[root@hadoop100 桌面]# service iptables status
```

##### (2) 临时关闭防火墙

```
[root@hadoop100 桌面]# service iptables stop
```

#### 2) 开机启动时关闭防火墙

##### (1) 查看防火墙开机启动状态

```
[root@hadoop100 桌面]# chkconfig iptables --list
```

##### (2) 设置开机时关闭防火墙

```
[root@hadoop100 桌面]# chkconfig iptables off
```

### 5.5 关机重启命令

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) halt (功能描述: 关闭系统, 等同于 shutdown -h now 和 poweroff)
- (3) reboot (功能描述: 就是重启, 等同于 shutdown -r now)
- (4) shutdown [选项] 时间

选项	功能
-h	-h=halt 关机
-r	-r=reboot 重启

参数	功能
now	立刻关机
时间	等待多久后关机 (时间单位是分钟)。

## 2) 经验技巧:

Linux 系统中为了提高磁盘的读写效率, 对磁盘采取了“预读迟写”操作方式。当用户保存文件时, Linux 核心并不一定立即将保存数据写入物理磁盘中, 而是将数据保存在缓冲区中, 等缓冲区满时再写入磁盘, 这种方式可以极大的提高磁盘写入数据的效率。但是, 也带来了安全隐患, 如果数据还未写入磁盘时, 系统掉电或者其他严重问题出现, 则将导致数据丢失。使用 sync 指令可以立即将缓冲区的数据写入磁盘。

## 3) 案例实操:

- (1) 将数据由内存同步到硬盘中

```
[root@hadoop100 桌面]#sync
```

- (2) 重启

```
[root@hadoop100 桌面]# reboot
```

- (3) 关机

```
[root@hadoop100 桌面]#halt
```

- (4) 计算机将在 1 分钟后关机, 并且会显示在登录用户的当前屏幕中

```
[root@hadoop100 桌面]#shutdown -h 1 'This server will shutdown after 1 mins'
```

- (5) 立马关机 (等同于 halt)

```
[root@hadoop100 桌面]# shutdown -h now
```

- (6) 系统立马重启 (等同于 reboot)

```
[root@hadoop100 桌面]# shutdown -r now
```

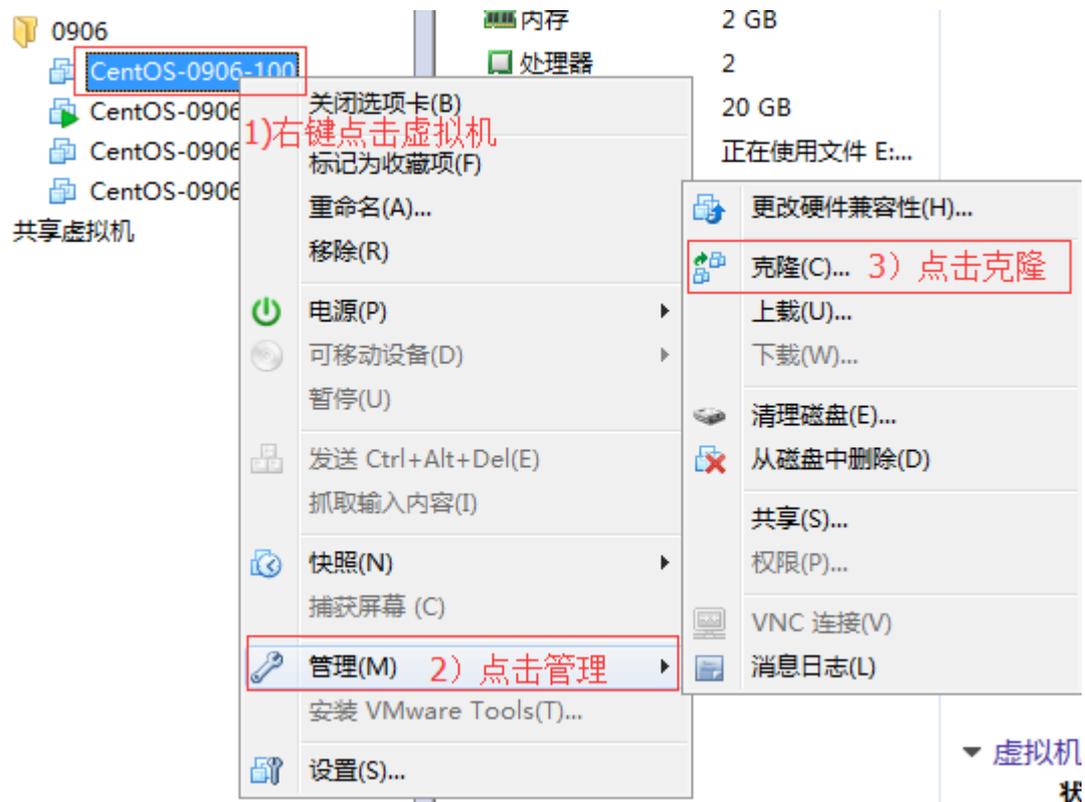
## 5.6 找回 root 密码



尚硅谷大数据技术  
之找回root密码.doc

## 5.7 克隆虚拟机

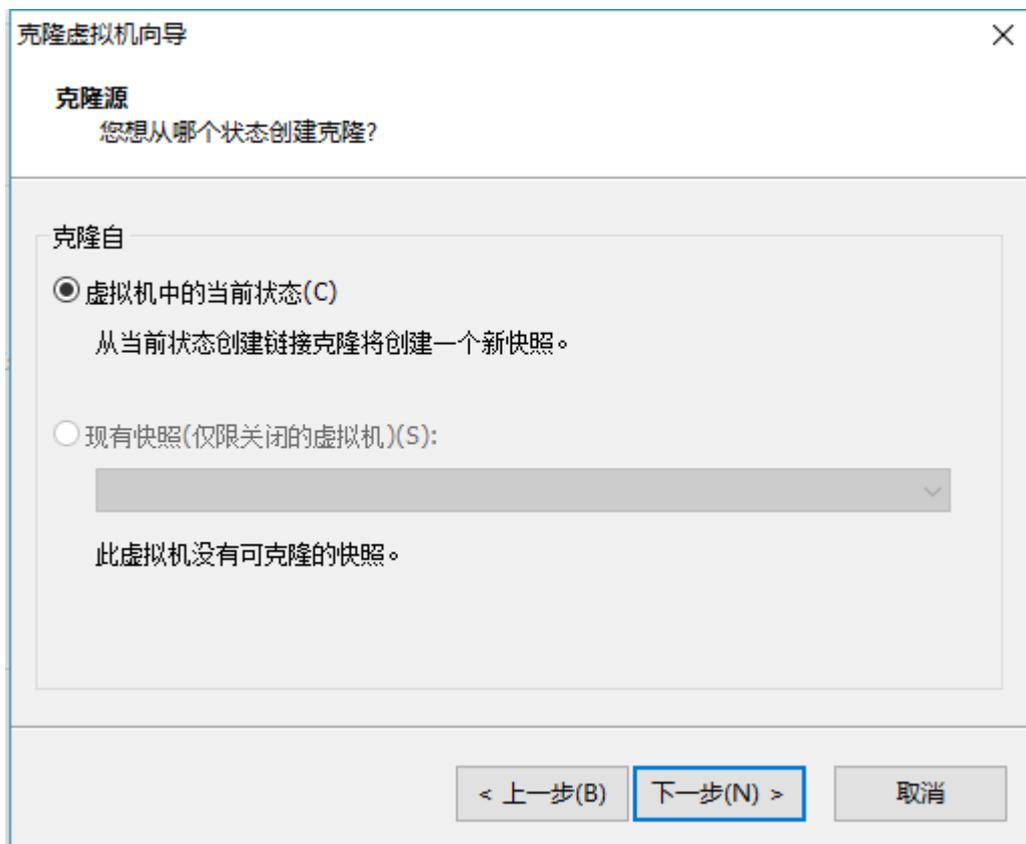
- 1) 关闭要被克隆的虚拟机
- 2) 找到克隆选项



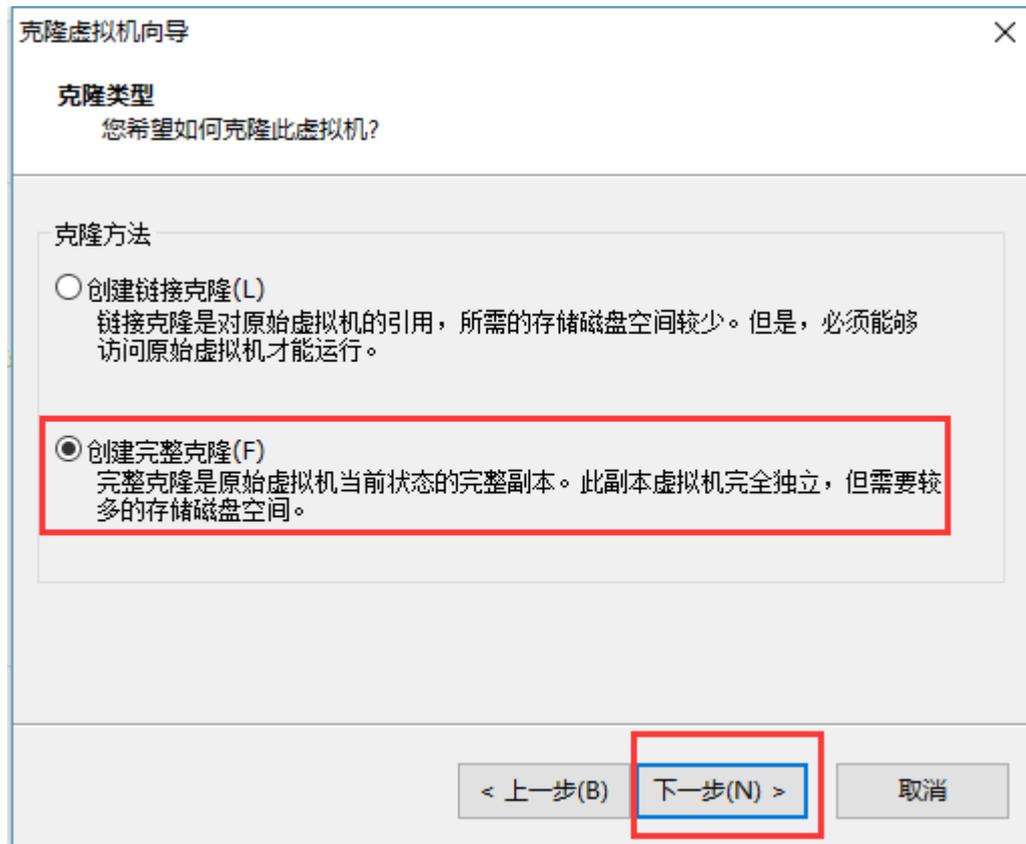
- 3) 欢迎页面



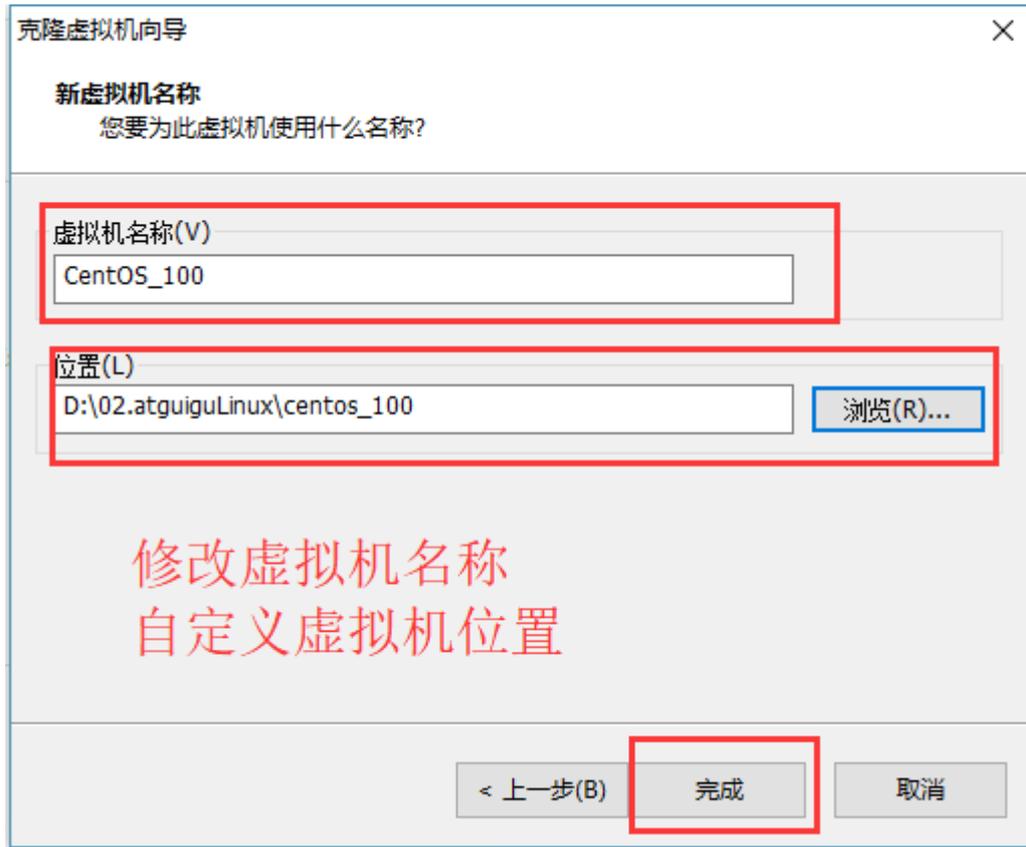
#### 4) 克隆虚拟机



## 5) 设置创建完整克隆



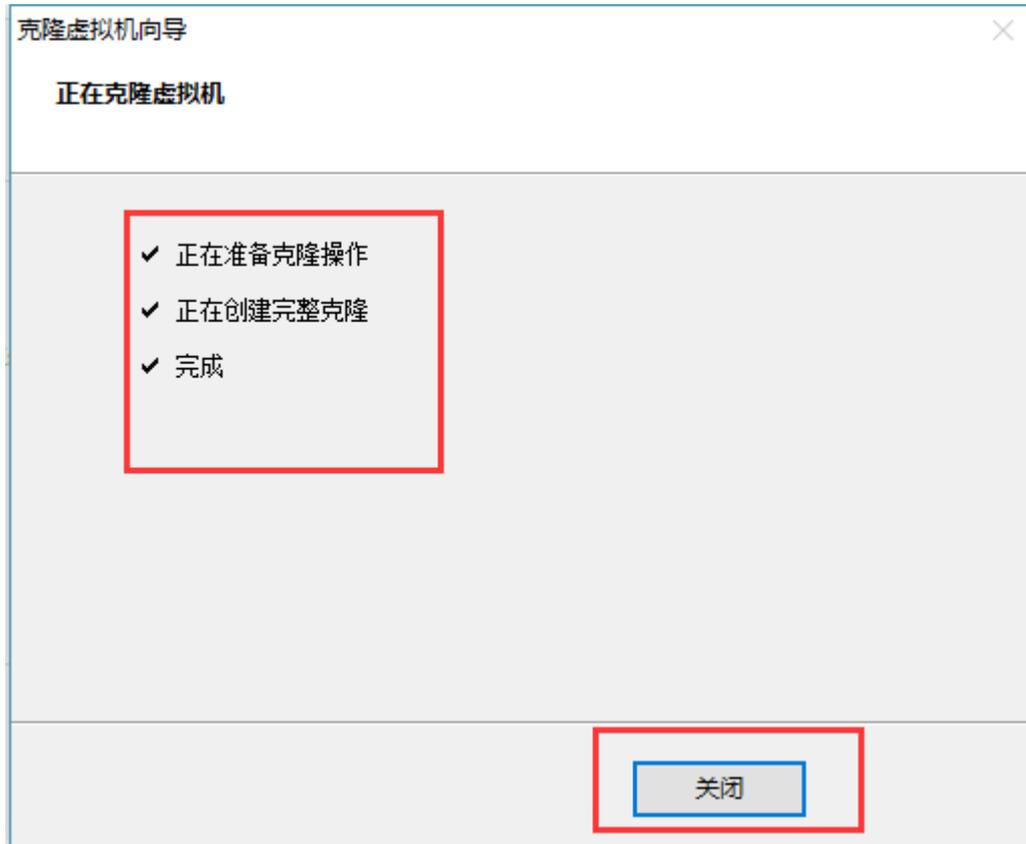
## 6) 设置克隆的虚拟机名称和存储位置



7) 等待正在克隆



8) 点击关闭，完成克隆



9) 修改克隆后虚拟机的 ip

```
[root@hadoop101 ~]#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}=="00:0c:29:0c: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

10) 修改 IP 地址

```
[root@hadoop101 ~]#vim /etc/sysconfig/network-scripts/ifcfg-eth0
```

(1) 把复制的物理 ip 地址更新

```
HWADDR=00:0C:2x:6x:0x:xx #MAC 地址
```

(2) 修改成你想要的 ip

```
IPADDR=192.168.1.101 #IP 地址
```

11) 修改主机名称

详见 5.3。

12) 重新启动服务器

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：[尚硅谷官网](#)

## 第 6 章 远程登录

### 6.1 安装 SecureCRT(英文版)

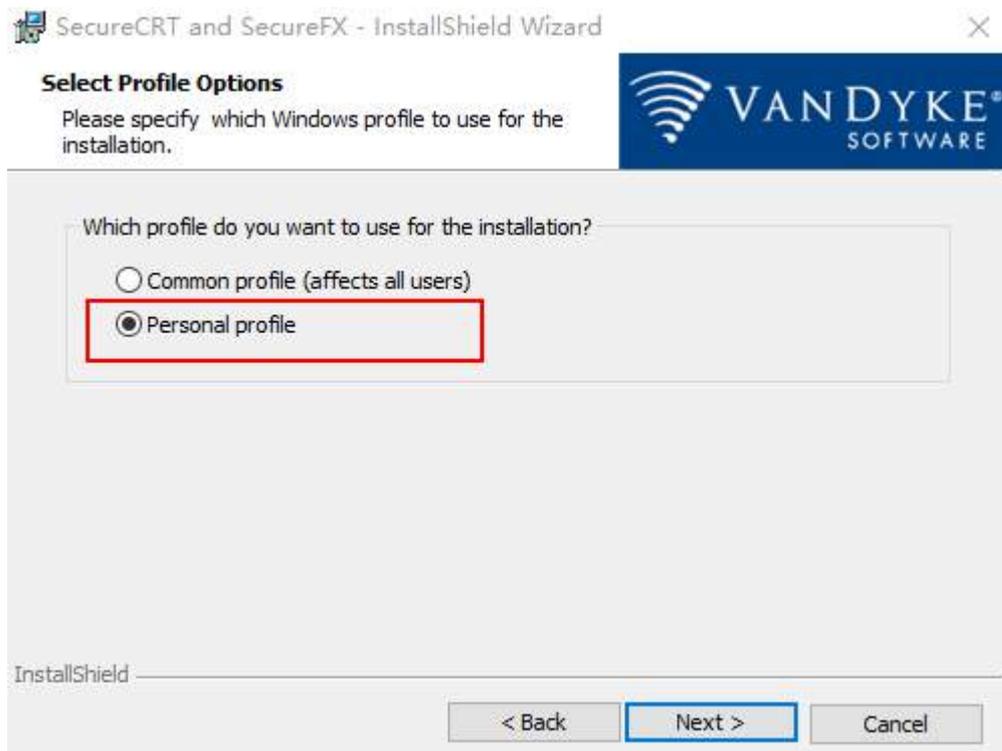
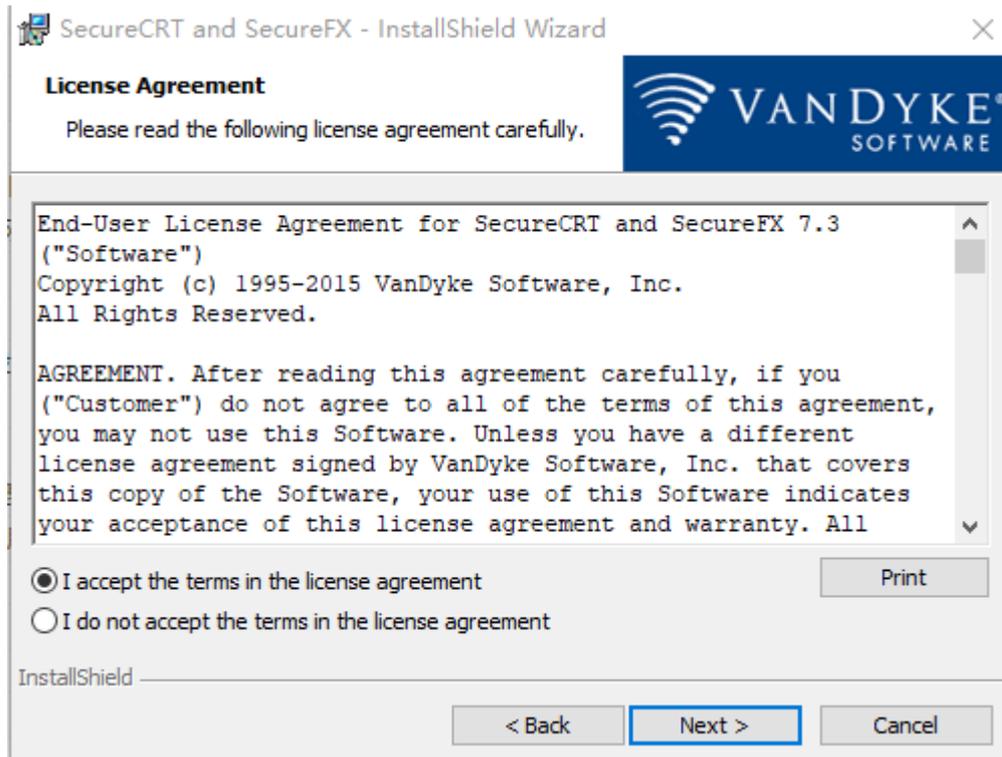
Linux 远程登录及相关工具介绍

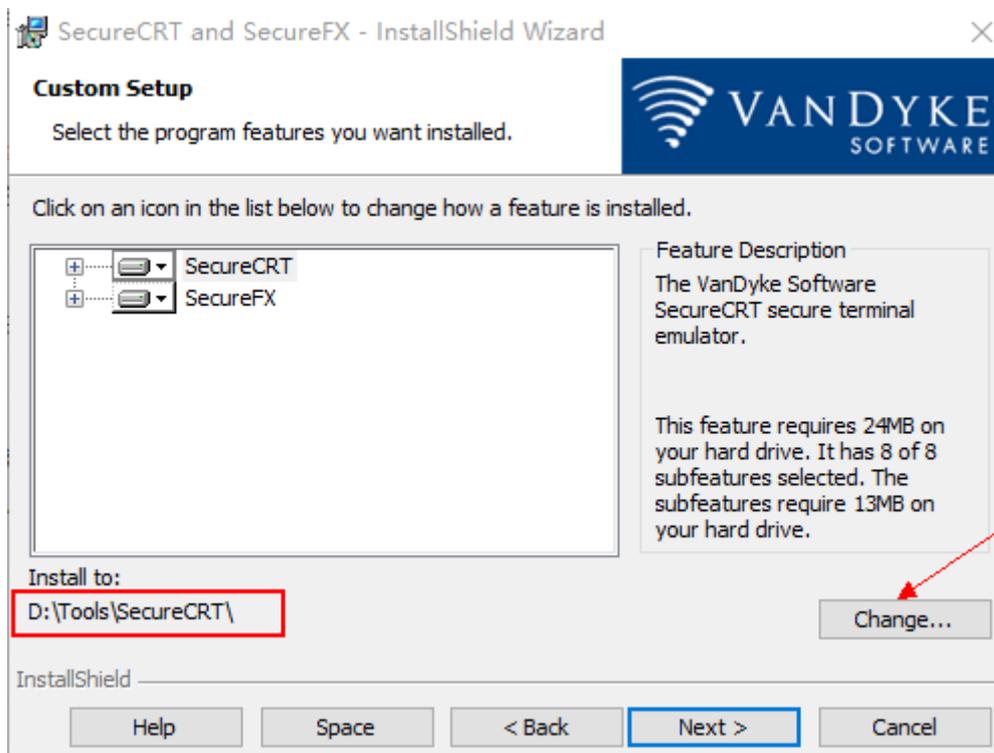
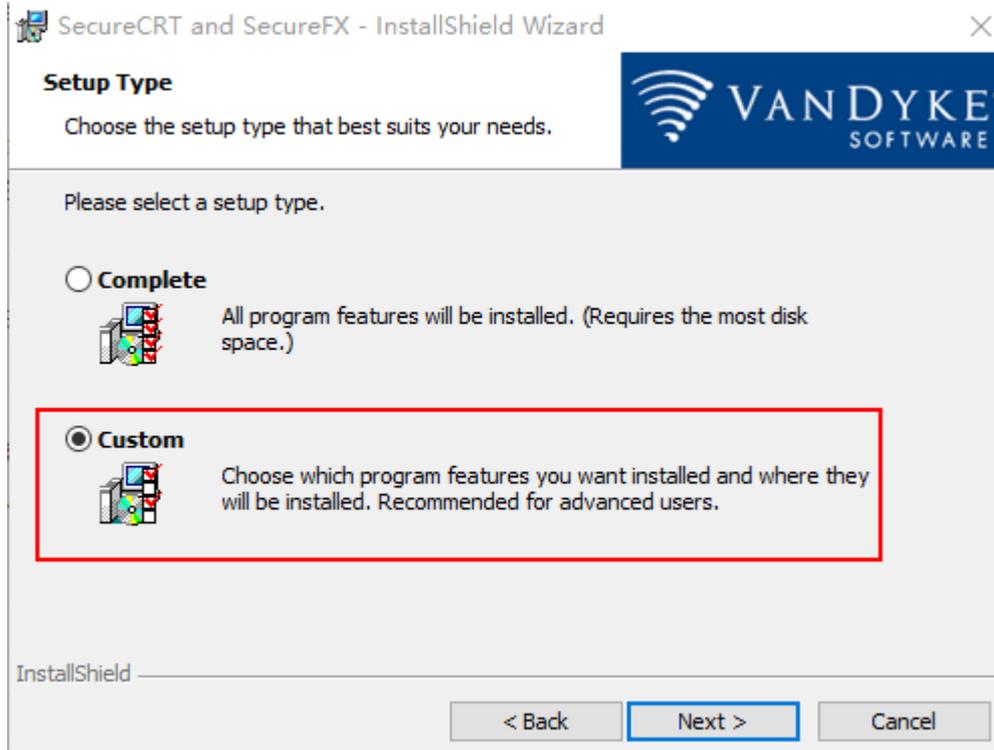
Linux 一般作为服务器使用，而服务器一般放在机房，你不可能在机房操作你的 Linux 服务器。这时我们就需要远程登录到 Linux 服务器来管理维护系统。

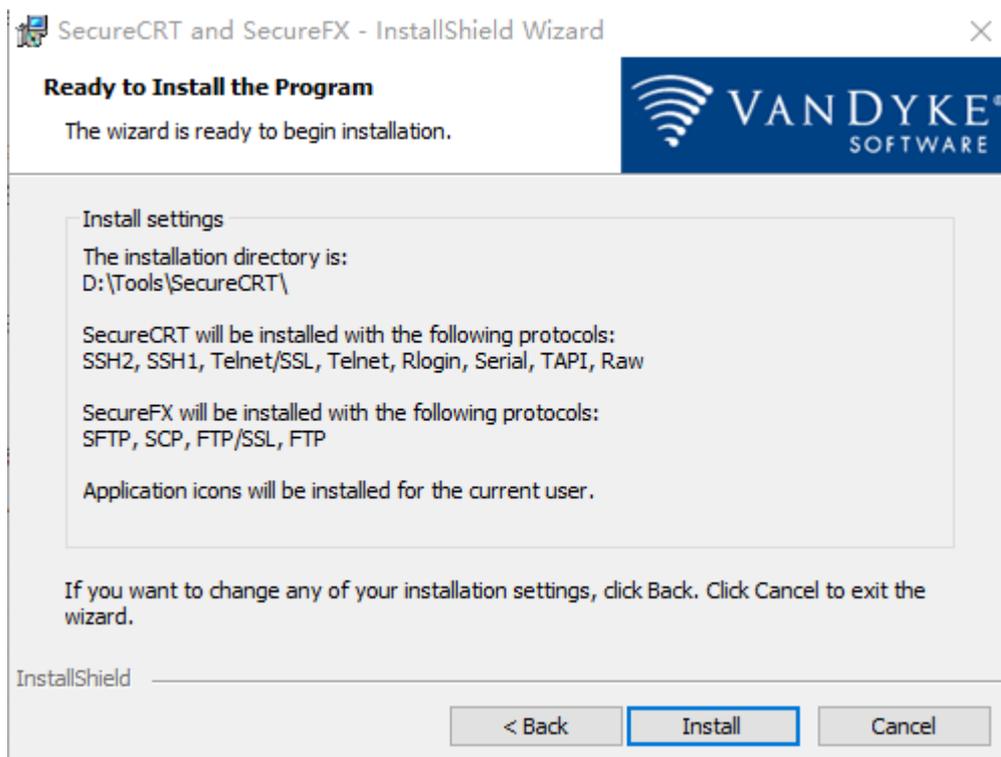
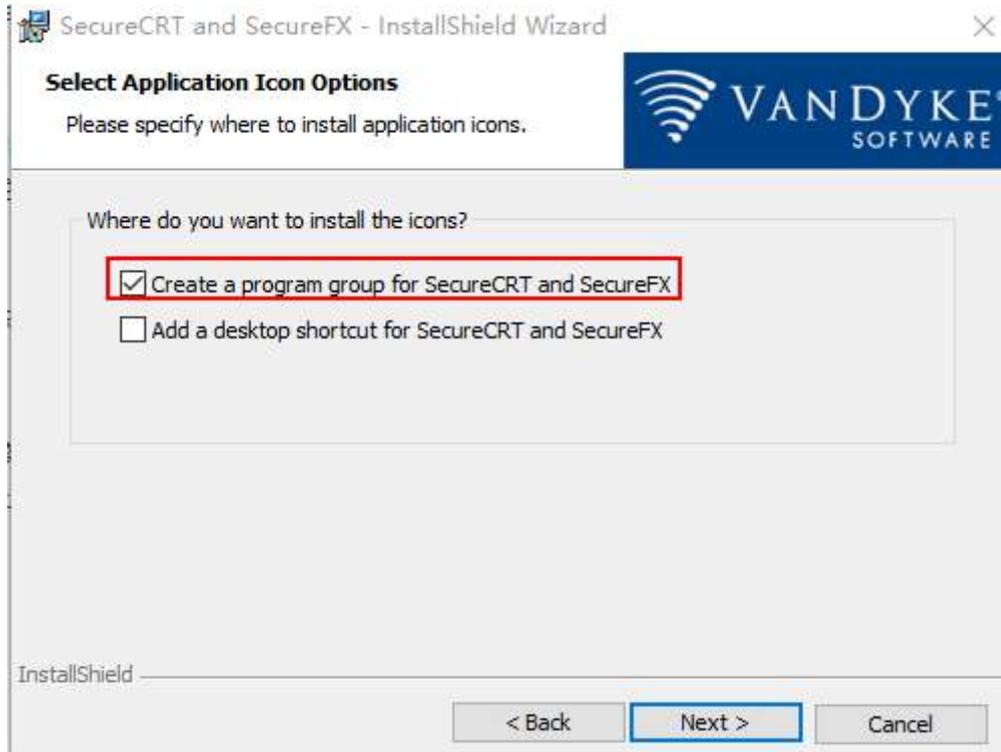
Linux 系统中是通过 SSH 服务实现的远程登录功能，默认 ssh 服务端口号为 22。Window 系统上 Linux 远程登录客户端有 SecureCRT, Putty, SSH Secure Shell, XShell 等

#### 1) 安装步骤



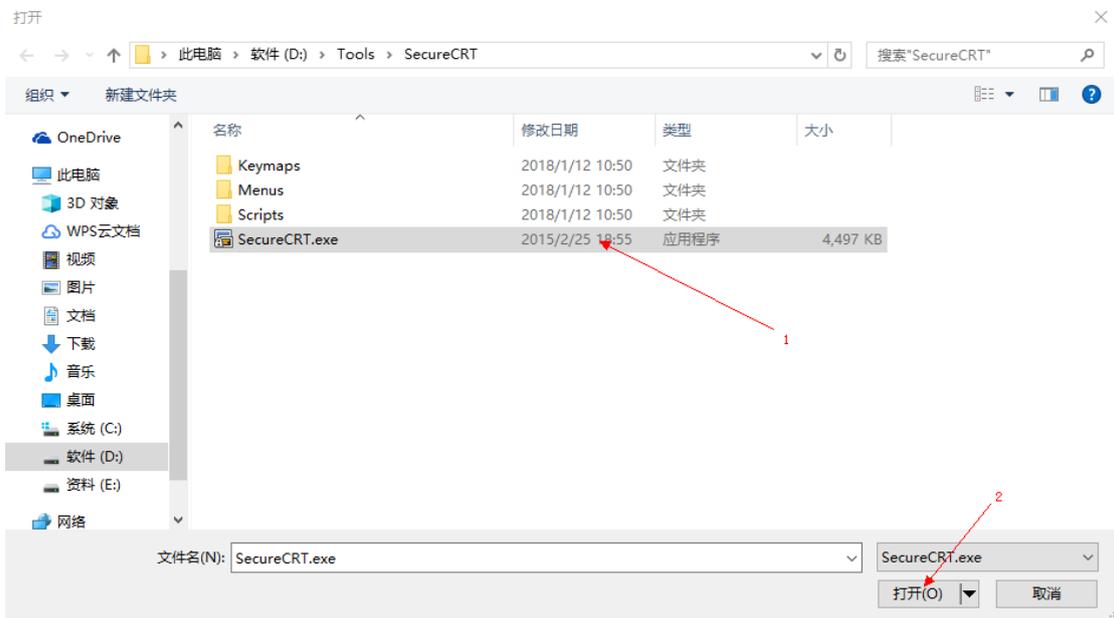
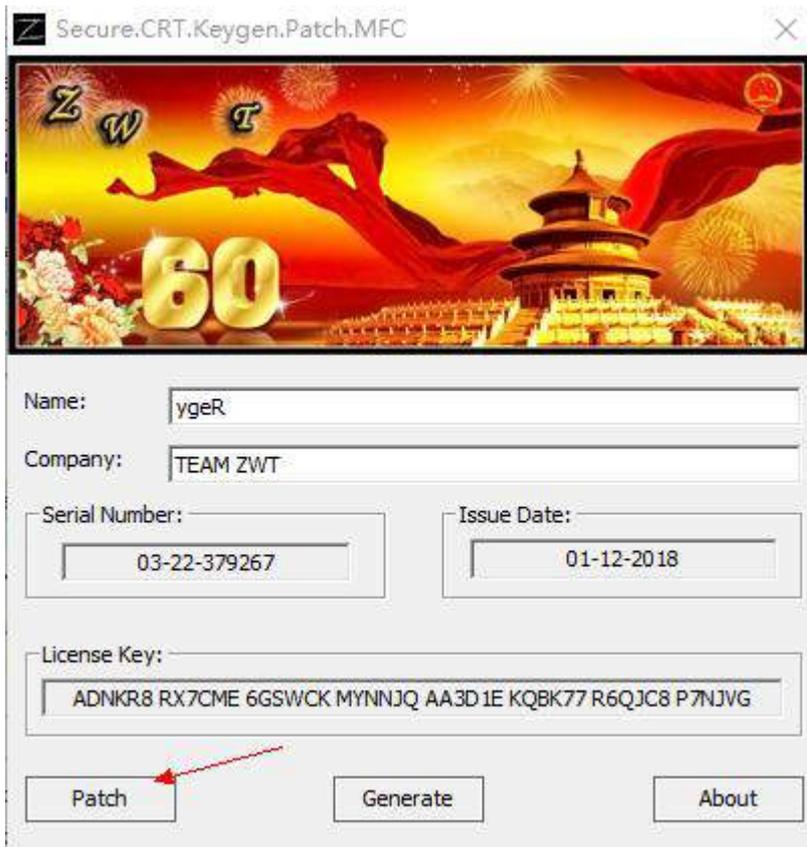


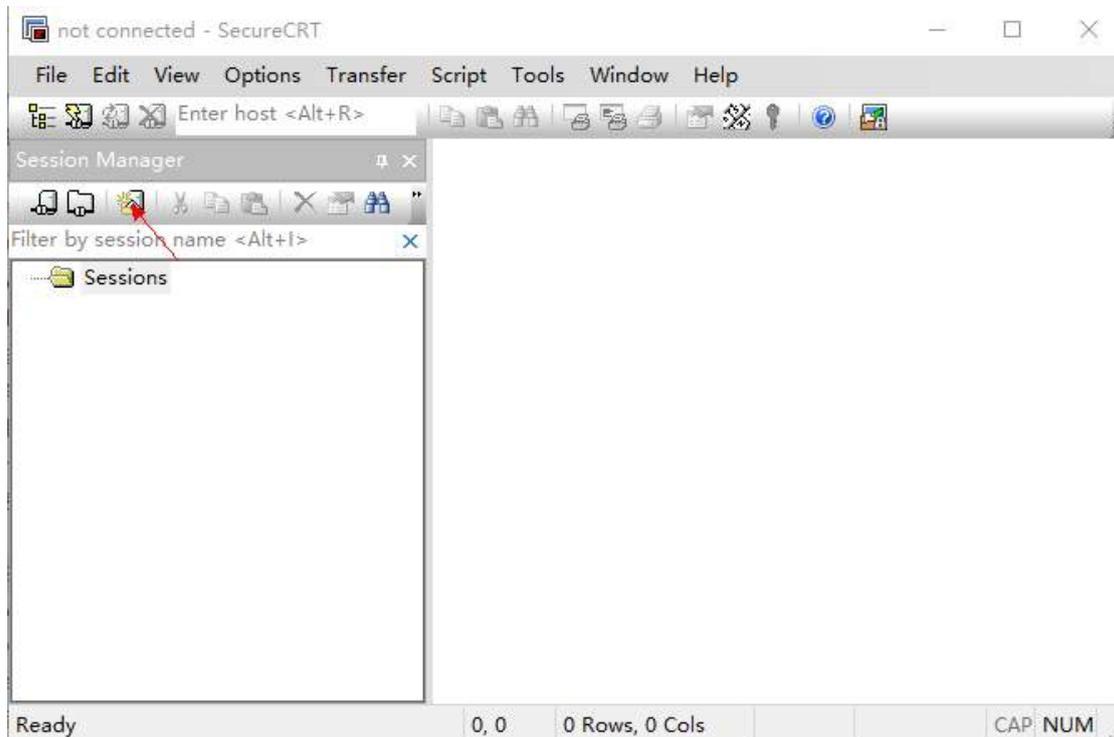
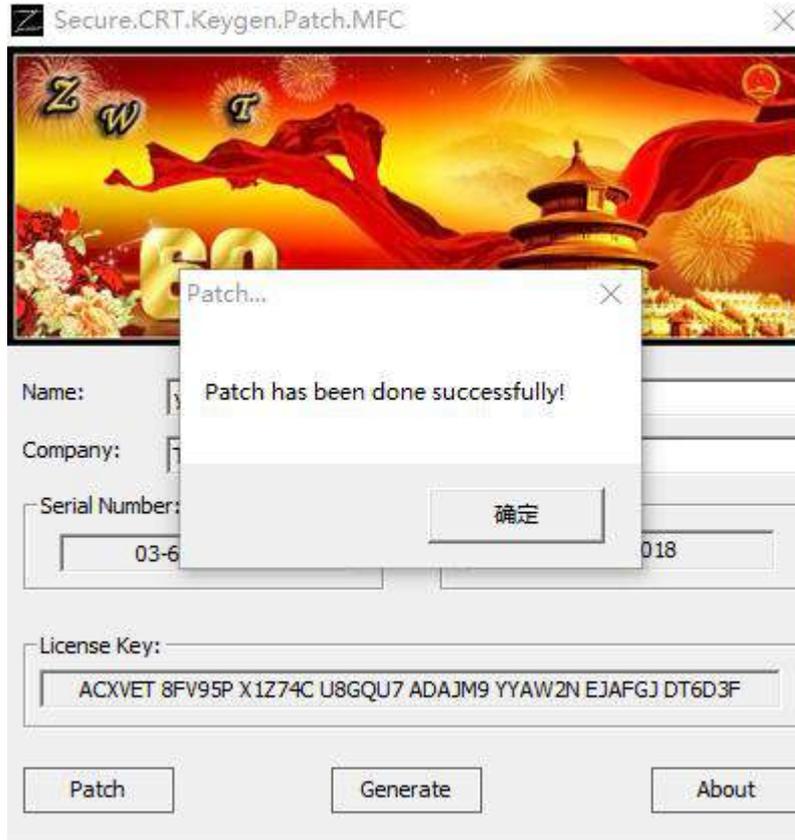


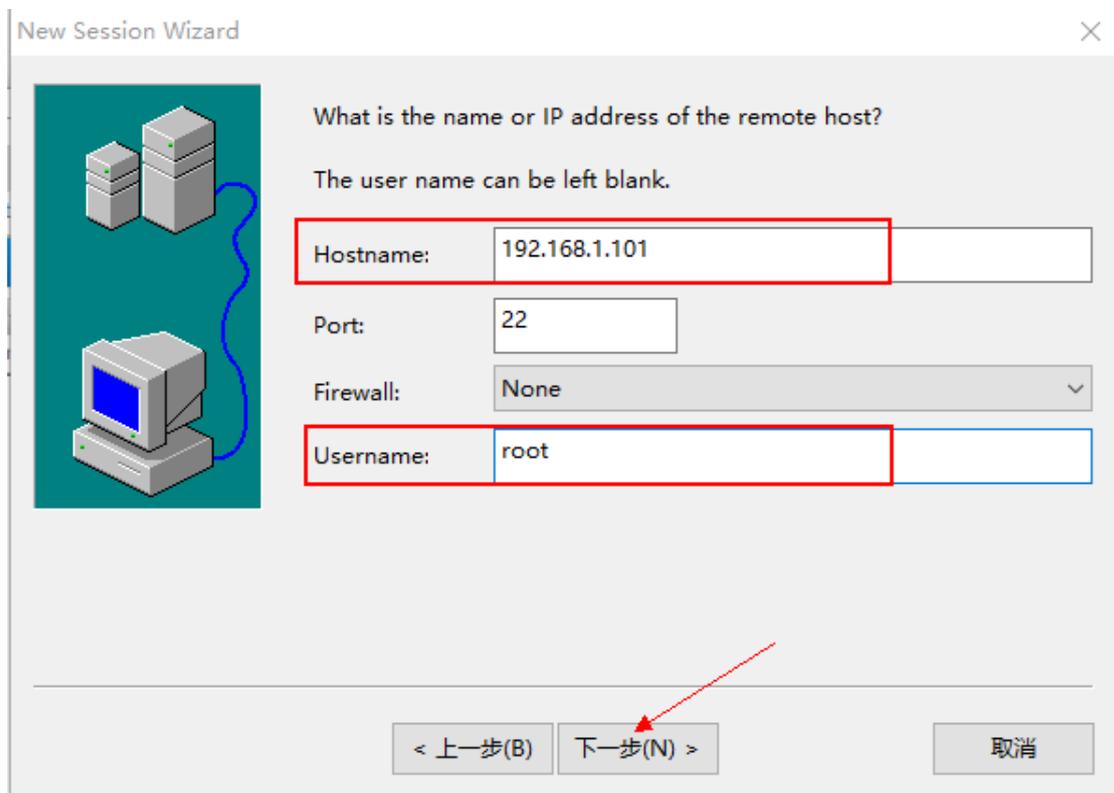
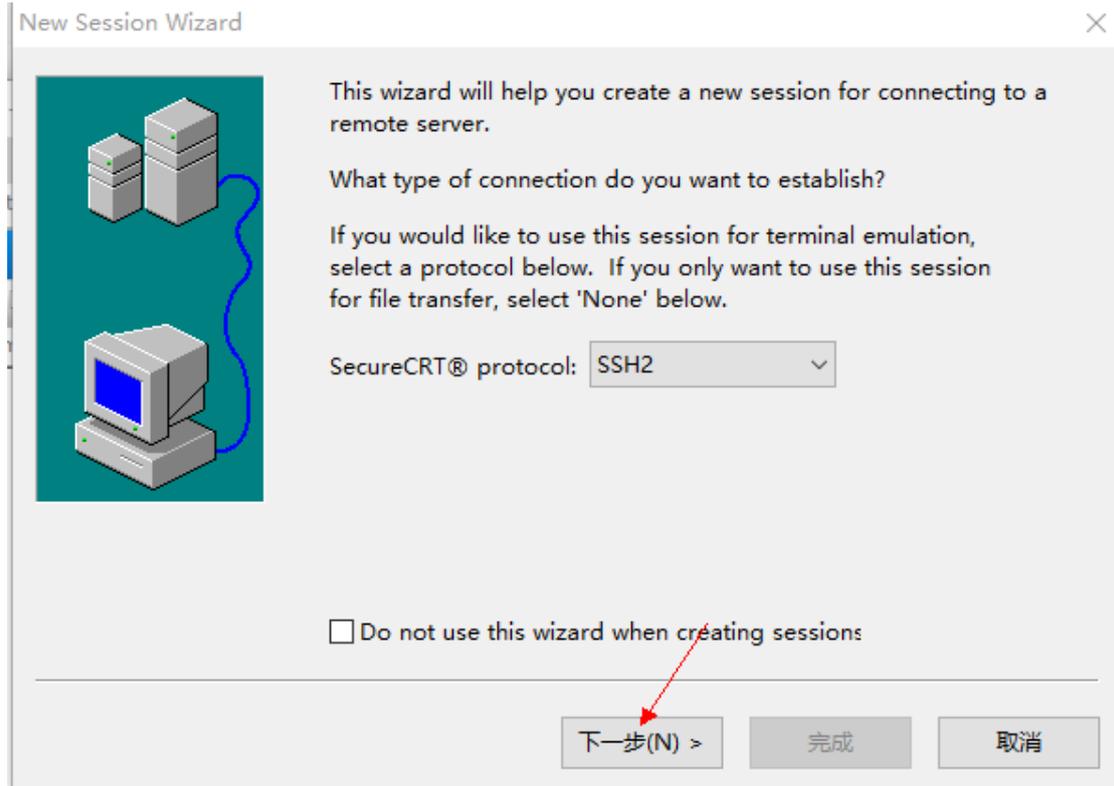


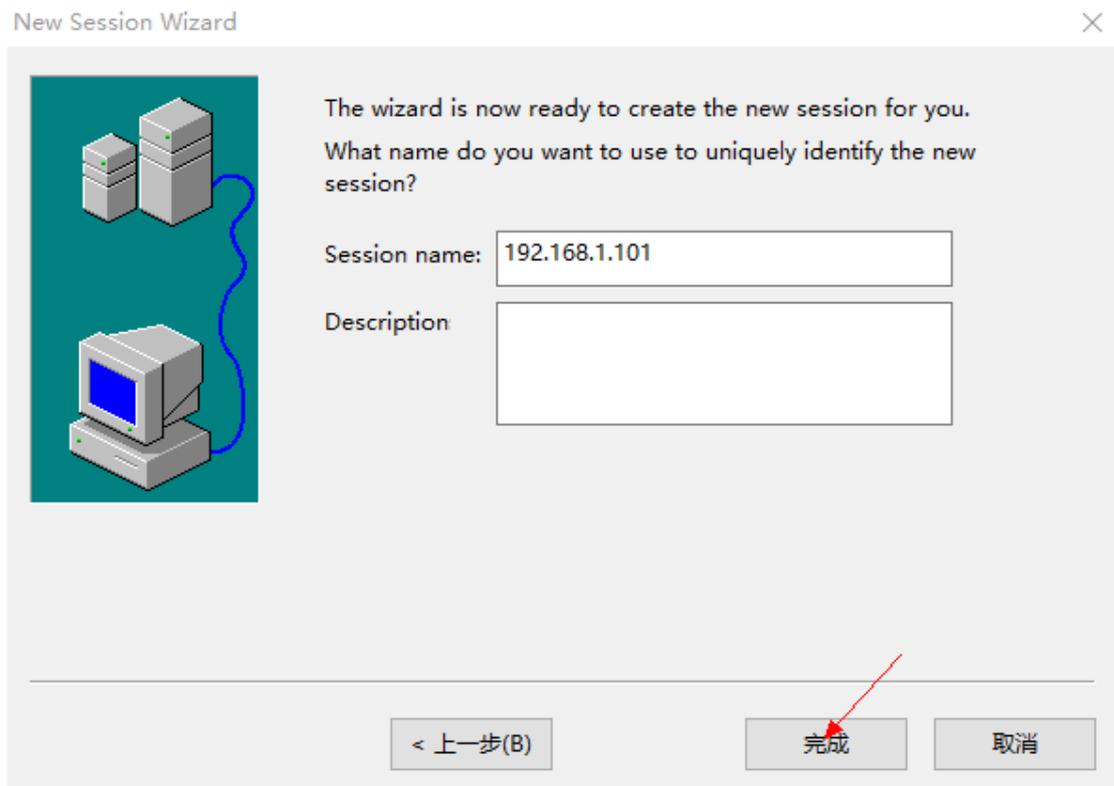
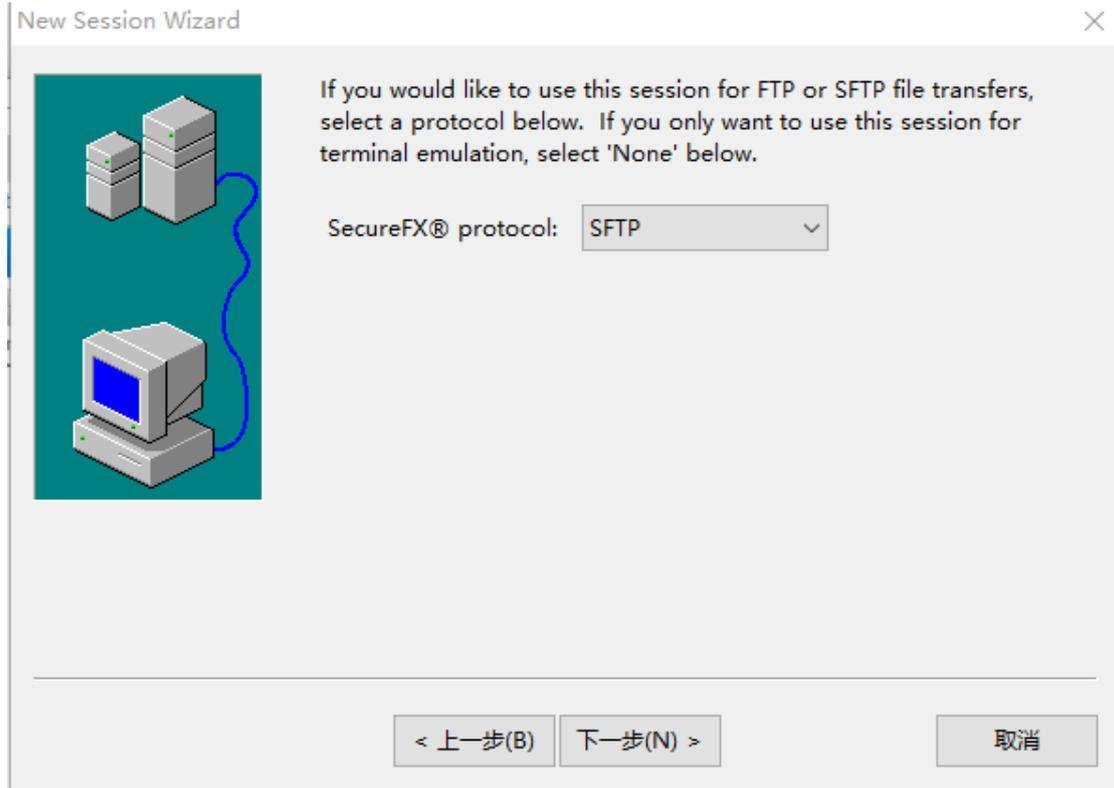


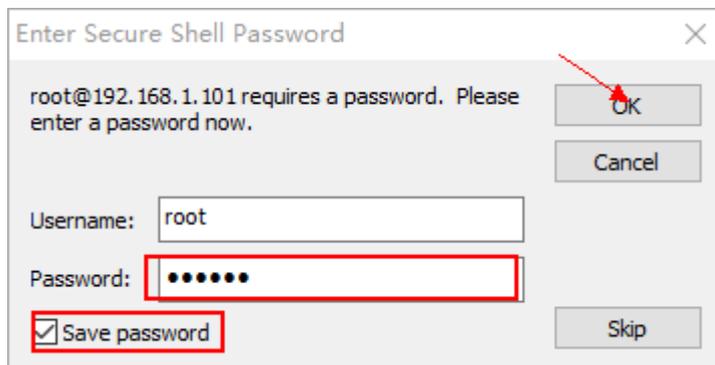
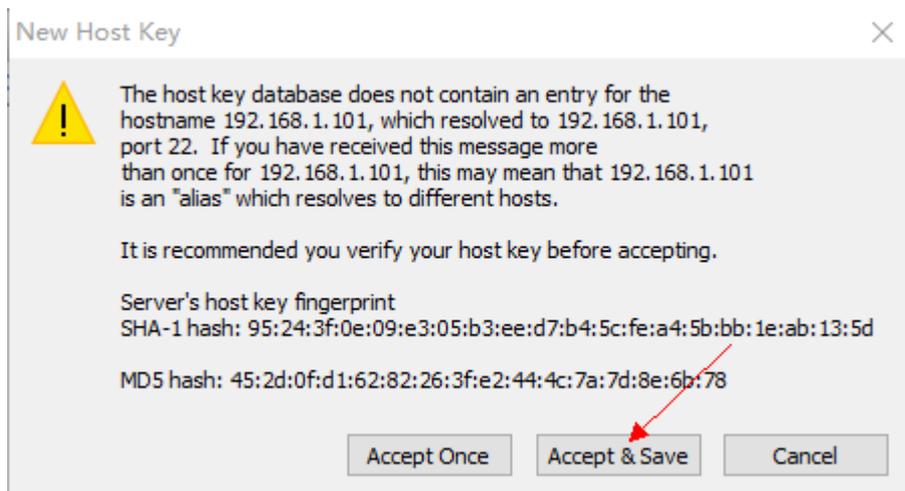
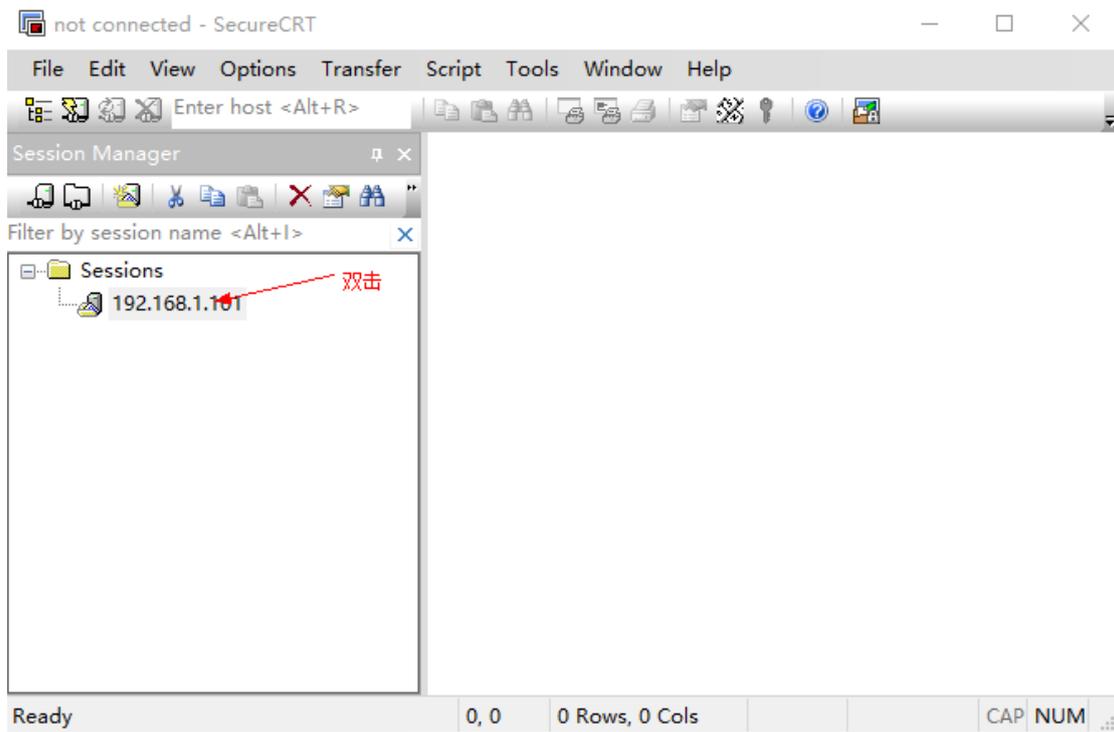
	cryptocme.dll	2013/12/16 10:47	应用程序扩展	288 KB
	cryptocme.sig	2013/12/16 10:47	SIG 文件	3 KB
	cryptocme_test_on_use.sig	2013/12/16 10:47	SIG 文件	3 KB
	FTP.dll	2015/2/25 18:55	应用程序扩展	607 KB
	FTPSSL.dll	2015/2/25 18:55	应用程序扩展	619 KB
	keygen.exe	2017/12/16 14:56	应用程序	433 KB
	LicenseHelper.exe	2015/2/25 18:55	应用程序	859 KB
	LicenseUI73U.dll	2015/2/25 18:55	应用程序扩展	868 KB
	mfc120enu.dll	2014/1/16 9:24	应用程序扩展	64 KB
	mfc120u.dll	2014/1/16 9:24	应用程序扩展	5,503 KB
	msvcp100.dll	2011/8/15 8:47	应用程序扩展	594 KB
	msvcp120.dll	2014/1/16 9:24	应用程序扩展	645 KB

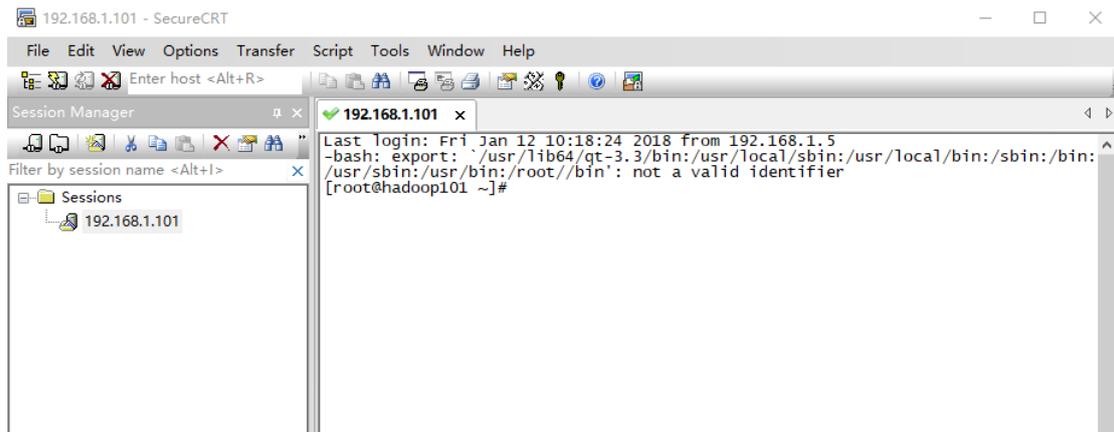




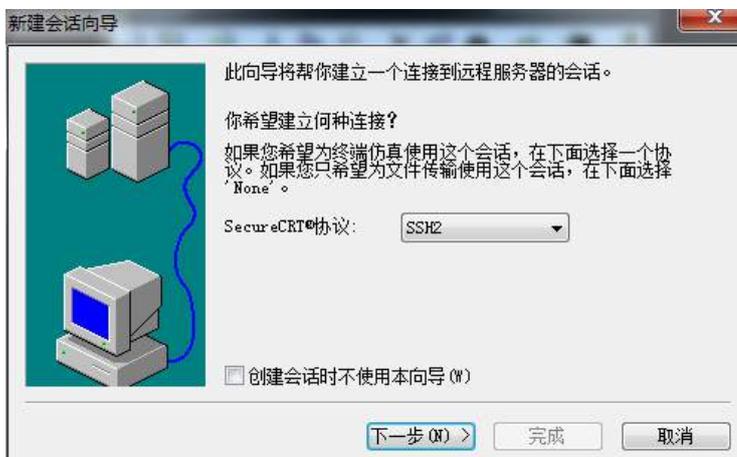
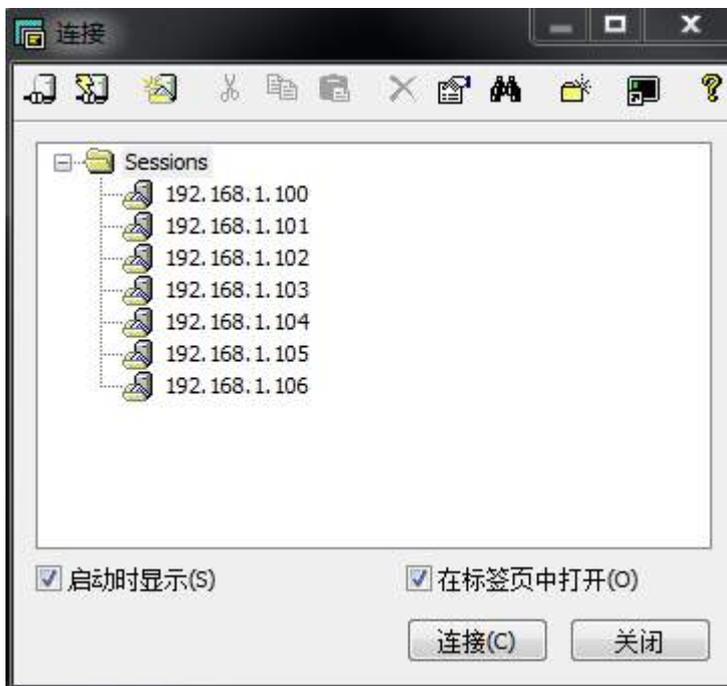


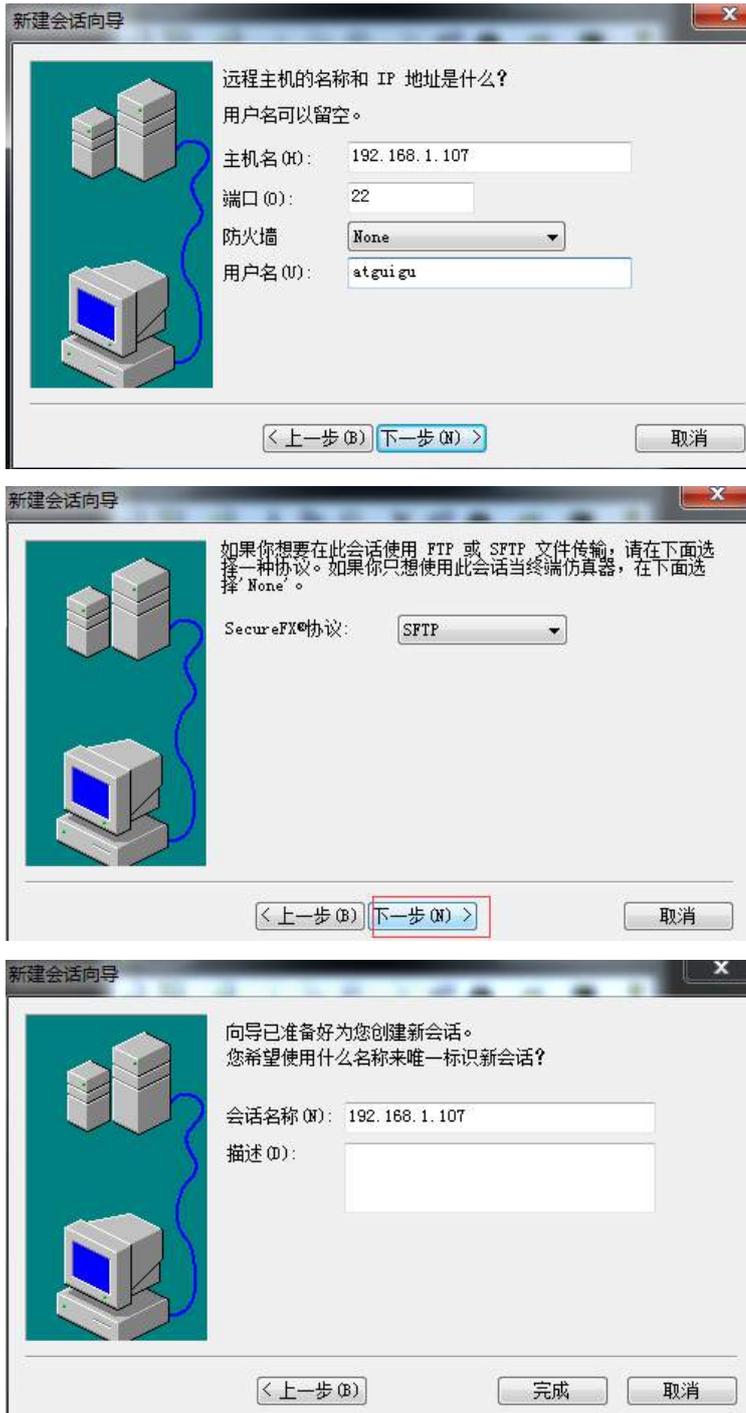






## 6.2 中文版 SecureCRT 的使用





### 6.3 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"】

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：[尚硅谷官网](#)

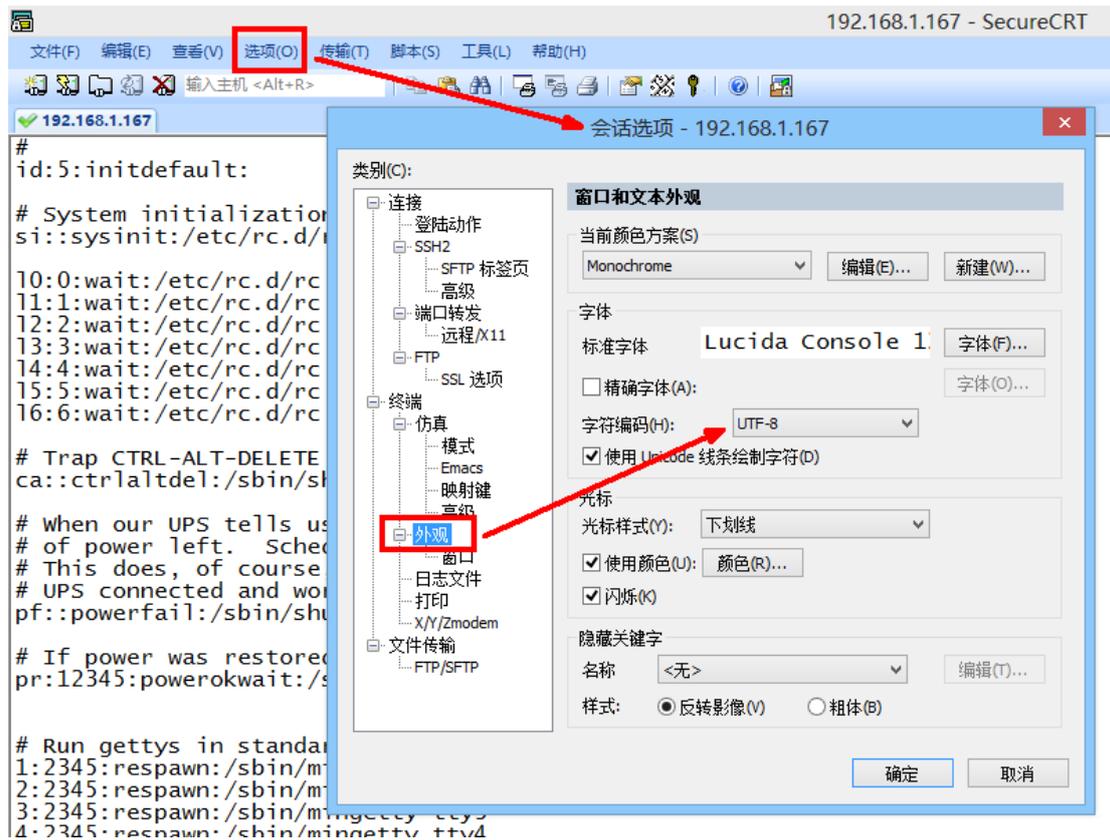
保存文件。断开 SSH，重新登录。就正常了

下面是修改后的查看

```
[root@hadoop101 ~]# cat /etc/sysconfig/i18n
```

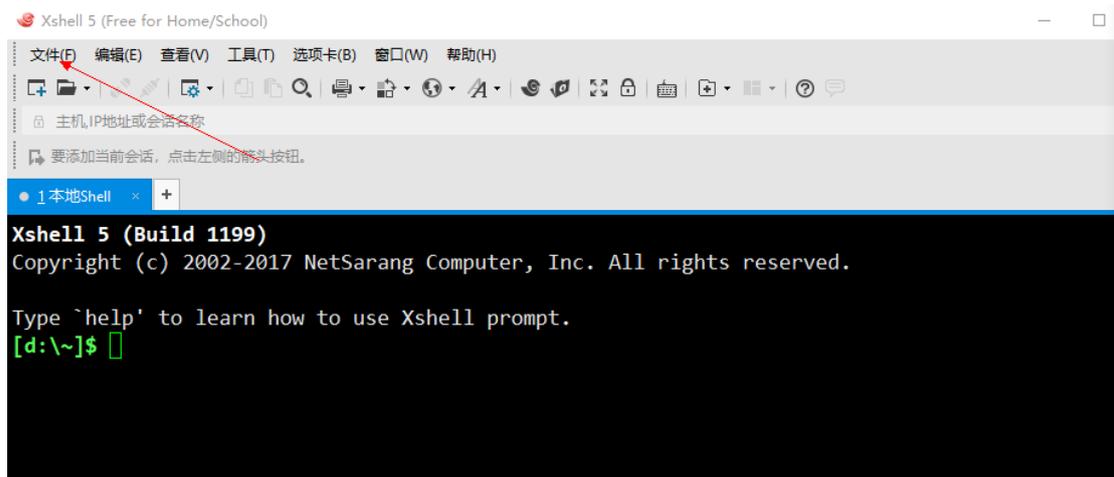
```
LANG="zh_CN.UTF-8"
```

### 3) 调整设置 CRT 解决



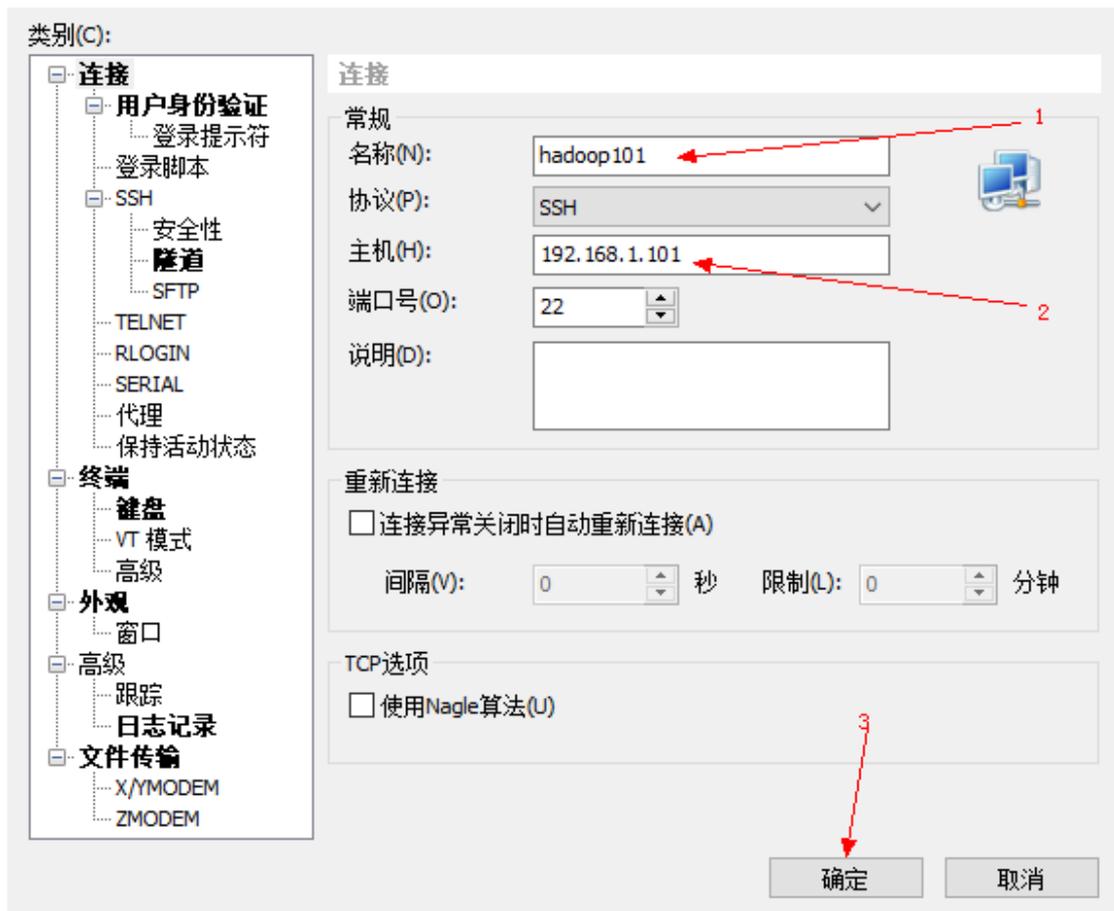
## 6.4 安装 Xshell

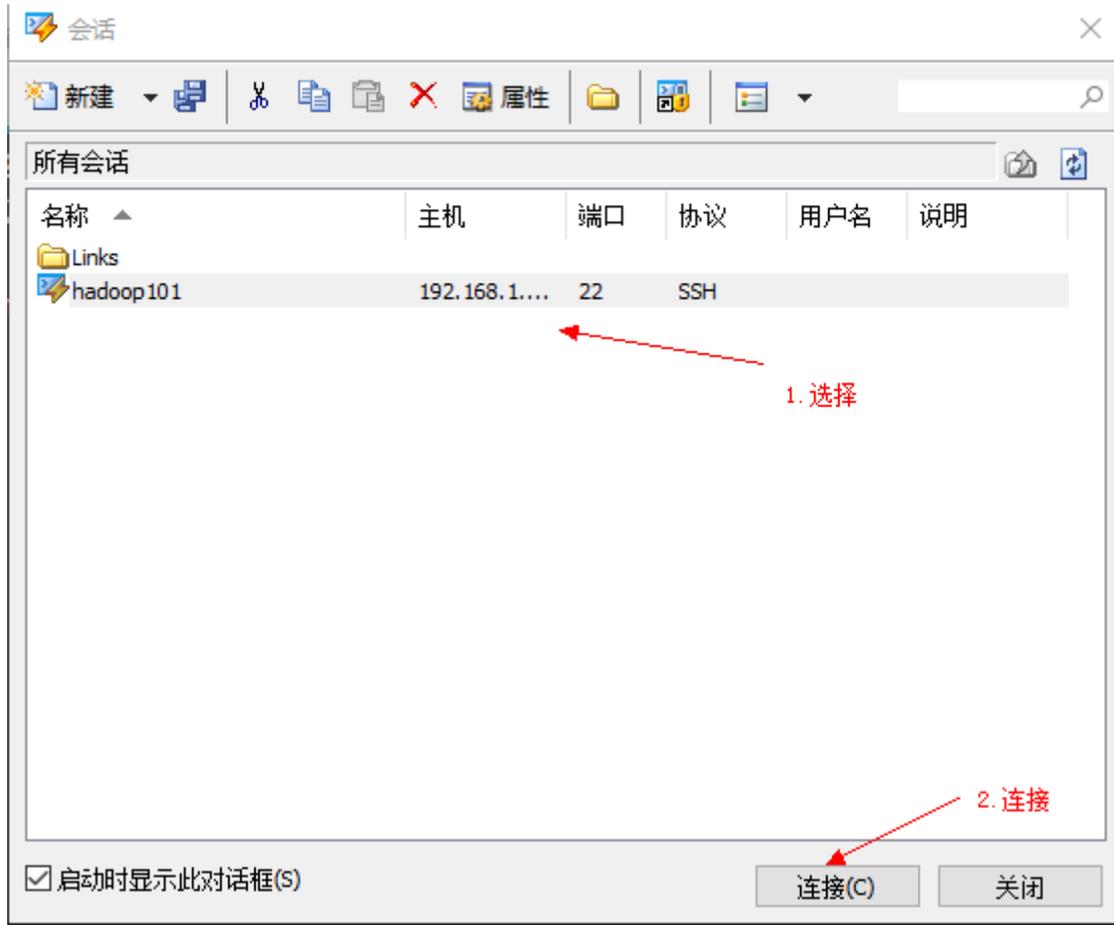
名称	修改日期	类型	大小
SecureCRSecureFXPortable	2018/1/3 13:52	文件夹	
SecureCRT and SecureFX v7.3.2 x86 x...	2018/1/3 13:52	文件夹	
cdkey.txt	2017/12/16 14:56	文本文档	1 KB
CentOS-6.8-x86_64-bin-DVD1.iso	2017/12/16 15:19	光盘映像文件	3,824,640...
FileZilla.zip	2017/12/16 15:19	360压缩 ZIP 文件	7,079 KB
Notepad++.zip	2017/12/16 15:19	360压缩 ZIP 文件	7,606 KB
SecureCRT and SecureFX v7.3.2 x86 x...	2017/12/16 15:19	360压缩 ZIP 文件	41,006 KB
Sublime Text Build 3143 x64 Setup.exe	2017/12/16 13:23	应用程序	8,744 KB
VMware-workstation-full-10.0.4-2249...	2017/12/16 15:23	应用程序	502,922 KB
XshellXftpPortable.zip	2017/12/18 10:07	360压缩 ZIP 文件	15,718 KB



打开文件->新建

新建会话属性





SSH用户身份验证

远程主机: 192.168.1.101:22 (hadoop101)  
登录名: root  
服务器类型: SSH2, OpenSSH\_5.3

请在下面选择恰当的身份验证方法并提供登录所需的信息。

Password(P)  
密码(W):

Public Key(U)  
用户密钥(K):   
密码(H):

Keyboard Interactive(I)  
使用键盘输入用户身份验证。

记住密码(R)

确定 取消

```
1 root@hadoop101:~  
Connection closed.  
Disconnected from remote host(hadoop101) at 22:05:24.  
Type `help' to learn how to use Xshell prompt.  
[d:\~]$  
Connecting to 192.168.1.101:22...  
Connection established.  
To escape to local shell, press 'Ctrl+Alt+]'.  
Last login: Thu Jan 11 22:04:58 2018 from vivo_x6plus_d  
-bash: export: `~/usr/lib64/qt-3.3/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root//bin': not a valid identifier  
[root@hadoop101 ~]#
```

## 第 7 章 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)

2) 显示说明

信息	功能
<b>NAME</b>	命令的名称和单行描述
<b>SYNOPSIS</b>	怎样使用命令
<b>DESCRIPTION</b>	命令功能的深入讨论
<b>EXAMPLES</b>	怎样使用命令的例子
<b>SEE ALSO</b>	相关主题 (通常是手册页)

3) 案例实操

(1) 查看 `ls` 命令的帮助信息

```
[root@hadoop101 ~]# 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) 案例实操:

(1) 查看 `cd` 命令的帮助信息

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

#### 7.1.3 常用快捷键

常用快捷键	功能
<code>ctrl + c</code>	停止进程
<code>ctrl+l</code>	清屏; 彻底清屏是: <code>reset</code>
<code>ctrl + q</code>	退出
善于用 <code>tab</code> 键	提示(更重要的是可以防止敲错)

上下键	查找执行过的命令
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

pwd:print working directory 打印工作目录

#### 1) 基本语法:

pwd (功能描述: 显示当前工作目录的绝对路径)

#### 2) 案例实操

(1) 显示当前工作目录的绝对路径

```
[root@hadoop101 ~]# pwd
/root
```

### 7.2.2 ls 列出目录的内容

ls:list 列出目录内容

#### 1) 基本语法:

ls [选项] [目录或是文件]

#### 2) 选项说明:

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

#### 3) 显示说明:

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

#### 4) 案例实操

(1) 查看当前目录的所有内容信息

```
[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
```

更多 [Java -大数据](#) [-前端](#) [-python](#) 人工智能资料下载, 可百度访问: [尚硅谷官网](#)

### 7.2.3 cd 切换目录

cd:Change Directory 切换路径

1) 基本语法:

```
cd [参数]
```

2) 参数说明

参数	功能
cd 绝对路径	切换路径
cd 相对路径	切换路径
cd ~或者 cd	回到自己的家目录
cd -	回到上一次所在目录
cd ..	回到当前目录的上一级目录
cd -P	跳转到实际物理路径，而非快捷方式路径

3) 案例实操

(1) 使用绝对路径切换到root目录

```
[root@hadoop101 ~]# cd /root/
```

(2) 使用相对路径切换到“公共的”目录

```
[root@hadoop101 ~]# cd 公共的/
```

(3) 表示回到自己的家目录，亦即是 /root 这个目录

```
[root@hadoop101 公共的]# cd ~
```

(4) cd- 回到上一次所在目录

```
[root@hadoop101 ~]# cd -
```

(5) 表示回到当前目录的上一级目录，亦即是 “/root/公共的”的上一级目录的意思;

```
[root@hadoop101 公共的]# cd ..
```

### 7.2.4 mkdir 创建一个新的目录

mkdir:Make directory 建立目录

1) 基本语法:

```
mkdir [选项] 要创建的目录
```

2) 选项说明:

选项	功能
-p	创建多层目录

### 3) 案例实操

(1) 创建一个目录

```
[root@hadoop101 ~]# mkdir xiyou
```

```
[root@hadoop101 ~]# mkdir xiyou/mingjie
```

(2) 创建一个多级目录

```
[root@hadoop101 ~]# mkdir -p xiyou/dssz/meihouwang
```

## 7.2.5 rmdir 删除一个空的目录

**rmdir**: Remove directory 移动目录

### 1) 基本语法:

**rmdir** 要删除的空目录

### 2) 案例实操

(1) 删除一个空的文件夹

```
[root@hadoop101 ~]# rmdir xiyou/dssz/meihouwang
```

## 7.2.6 touch 创建空文件

### 1) 基本语法:

**touch** 文件名称

### 2) 案例实操

```
[root@hadoop101 ~]# touch xiyou/dssz/sunwukong.txt
```

## 7.2.7 cp 复制文件或目录

### 1) 基本语法:

**cp** [选项] source dest (功能描述: 复制source文件到dest)

### 2) 选项说明

选项	功能
-r	递归复制整个文件夹

### 3) 参数说明

参数	功能
----	----

source	源文件
dest	目标文件

#### 4) 经验技巧

强制覆盖不提示的方法: \cp

#### 5) 案例实操

##### (1) 复制文件

```
[root@hadoop101 ~]# cp xiyou/dssz/suwukong.txt xiyou/mingjie/
```

##### (2) 递归复制整个文件夹

```
[root@hadoop101 ~]# cp -r xiyou/dssz/ ./
```

## 7.2.8 rm 移除文件或目录

### 1) 基本语法

rm [选项] deleteFile (功能描述: 递归删除目录中所有内容)

### 2) 选项说明

选项	功能
-r	递归删除目录中所有内容
-f	强制执行删除操作, 而不提示用于进行确认。
-v	显示指令的详细执行过程

### 3) 案例实操

#### (1) 删除目录中的内容

```
[root@hadoop101 ~]# rm xiyou/mingjie/sunwukong.txt
```

#### (1) 递归删除目录中所有内容

```
[root@hadoop101 ~]# rm -rf dssz/
```

## 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) 案例实操:

#### (1) 重命名

```
[root@hadoop101 ~]# mv xiyou/dss/suwukong.txt xiyou/dss/houge.txt
```

(2) 移动文件

```
[root@hadoop101 ~]# mv xiyou/dss/houge.txt ./
```

## 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
```

2) 选项说明

选项	功能描述
<b>-n</b>	显示所有行的行号，包括空行。

3) 经验技巧:

一般查看比较小的文件，一屏幕能显示全的。

4) 案例实操

(1) 查看文件内容并显示行号

```
[atguigu@hadoop101 ~]$ cat -n houge.txt
```

## 7.2.11 more 文件内容分屏查看器

`more` 指令是一个基于 VI 编辑器的文本过滤器，它以全屏幕的方式按页显示文本文件的内容。`more` 指令中内置了若干快捷键，详见操作说明。

1) 基本语法:

```
more 要查看的文件
```

2) 操作说明

操作	功能说明
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 <code>more</code> ，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) 案例实操

(1) 采用more查看文件

```
[root@hadoop101 ~]# more smartd.conf
```

## 7.2.12 less 分屏显示文件内容

less 指令用来分屏查看文件内容，它的功能与 more 指令类似，但是比 more 指令更加强大，支持各种显示终端。less 指令在显示文件内容时，并不是一次将整个文件加载之后才显示，而是根据显示需要加载内容，对于显示大型文件具有较高的效率。

### 1) 基本语法:

less 要查看的文件

### 2) 操作说明

操作	功能说明
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; n: 向下查找; N: 向上查找;
?字符串	向上搜寻『字符串』的功能; n: 向上查找; N: 向下查找;
q	离开 less 这个程序;

### 3) 经验技巧

用SecureCRT时[pagedown]和[pageup]可能会出现无法识别的问题。

### 4) 案例实操

(1) 采用less查看文件

```
[root@hadoop101 ~]# less smartd.conf
```

## 7.2.13 echo

### 1) echo 输出内容到控制台

(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
\n	换行符
\t	制表符，也就是 Tab 键

## (2) 案例实操

```
[atguigu@hadoop101 ~]$ echo "hello\tworld"
```

```
hello\tworld
```

```
[atguigu@hadoop101 ~]$ echo -e "hello\tworld"
```

```
hello    world
```

## 7.2.14 head 显示文件头部内容

head 用于显示文件的开头部分内容，默认情况下 head 指令显示文件的前 10 行内容。

### 1) 基本语法

head 文件 (功能描述: 查看文件头10行内容)

head -n 5 文件 (功能描述: 查看文件头5行内容, 5可以是任意行数)

### 2) 选项说明

选项	功能
-n<行数>	指定显示头部内容的行数

### 3) 案例实操

#### (1) 查看文件的头2行

```
[root@hadoop101 ~]# head -n 2 smartd.conf
```

## 7.2.15 tail 输出文件尾部内容

tail 用于输出文件中尾部的内容，默认情况下 tail 指令显示文件的前 10 行内容。

### 1) 基本语法

(1) tail 文件 (功能描述: 查看文件头10行内容)

(2) tail -n 5 文件 (功能描述: 查看文件头5行内容, 5可以是任意行数)

(3) tail -f 文件 (功能描述: 实时追踪该文档的所有更新)

### 2) 选项说明

选项	功能
-n<行数>	输出文件尾部n行内容
-f	显示文件最新追加的内容，监视文件变化

### 3) 案例实操

- (1) 查看文件头1行内容

```
[root@hadoop101 ~]# tail -n 1 smartd.conf
```

- (2) 实时追踪该档的所有更新

```
[root@hadoop101 ~]# tail -f hougex.txt
```

## 7.2.16 > 输出重定向和 >> 追加

### 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))

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

- (3) `cat 文件 1 > 文件 2` (功能描述: 将文件 1 的内容覆盖到文件 2)

- (4) `echo "内容" >> 文件`

### 2) 案例实操

- (1) 将 ls 查看信息写入到文件中

```
[root@hadoop101 ~]# ls -l>hougex.txt
```

- (2) 将 ls 查看信息追加到文件中

```
[root@hadoop101 ~]# ls -l>>hougex.txt
```

- (3) 采用 echo 将 hello 单词追加到文件中

```
[root@hadoop101 ~]# echo hello>>hougex.txt
```

## 7.2.17 ln 软链接

软链接也成为符号链接，类似于 windows 里的快捷方式，有自己的数据块，主要存放了链接其他文件的路径。

### 1) 基本语法:

`ln -s [原文件或目录] [软链接名]` (功能描述: 给原文件创建一个软链接)

### 2) 经验技巧

删除软链接: `rm -rf 软链接名`, 而不是 `rm -rf 软链接名/`

查询：通过 ll 就可以查看，列表属性第 1 位是 l，尾部会有位置指向。

### 3) 案例实操：

(1) 创建软连接

```
[root@hadoop101 ~]# mv hougge.txt xiyou/dssz/
```

```
[root@hadoop101 ~]# ln -s xiyou/dssz/hougge.txt ./houzi
```

```
[root@hadoop101 ~]# ll
```

```
lrwxrwxrwx. 1 root    root      20 6 月  17 12:56 houzi -> xiyou/dssz/hougge.txt
```

(2) 删除软连接

```
[root@hadoop101 ~]# rm -rf houzi
```

(3) 进入软连接实际物理路径

```
[root@hadoop101 ~]# ln -s xiyou/dssz/ ./dssz
```

```
[root@hadoop101 ~]# cd -P dssz/
```

## 7.2.18 history 查看已经执行过历史命令

### 1) 基本语法：

history (功能描述：查看已经执行过历史命令)

### 2) 案例实操

(1) 查看已经执行过的历史命令

```
[root@hadoop101 test1]# history
```

## 7.3 时间日期类

### 1) 基本语法

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

### 2) 选项说明

选项	功能
-d<时间字符串>	显示指定的“时间字符串”表示的时间，而非当前时间
-s<日期时间>	设置系统日期时间

### 3) 参数说明

参数	功能
<+日期时间格式>	指定显示时使用的日期时间格式

### 7.3.1 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 %H:%M:%S" (功能描述: 显示年月日时分秒)

#### 2) 案例实操

- (1) 显示当前时间信息

```
[root@hadoop101 ~]# date
```

```
2017 年 06 月 19 日 星期一 20:53:30 CST
```

- (2) 显示当前时间年月日

```
[root@hadoop101 ~]# date +%Y%m%d
```

```
20170619
```

- (3) 显示当前时间年月日时分秒

```
[root@hadoop101 ~]# date "+%Y-%m-%d %H:%M:%S"
```

```
2017-06-19 20:54:58
```

### 7.3.2 date 显示非当前时间

#### 1) 基本语法:

- (1) date -d '1 days ago' (功能描述: 显示前一天时间)
- (2) date -d '-1 days ago' (功能描述: 显示明天时间)

#### 2) 案例实操:

- (1) 显示前一天

```
[root@hadoop101 ~]# date -d '1 days ago'
```

```
2017 年 06 月 18 日 星期日 21:07:22 CST
```

- (2) 显示明天时间

```
[root@hadoop101 ~]# date -d '-1 days ago'
```

```
2017 年 06 月 20 日 星期日 21:07:22 CST
```

### 7.3.3 date 设置系统时间

1) 基本语法:

`date -s 字符串时间`

2) 案例实操

(1) 设置系统当前时间

```
[root@hadoop101 ~]# date -s "2017-06-19 20:52:18"
```

### 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)

2) 选项说明

选项	功能
具体某一年	显示这一年的日历

3) 案例实操:

(1) 查看当前月的日历

```
[root@hadoop101 ~]# cal
```

(2) 查看 2017 年的日历

```
[root@hadoop101 ~]# cal 2017
```

## 7.4 用户管理命令

### 7.4.1 useradd 添加新用户

1) 基本语法:

`useradd 用户名` (功能描述: 添加新用户)

`useradd -g 组名 用户名` (功能描述: 添加新用户到某个组)

2) 案例实操:

(1) 添加一个用户

```
[root@hadoop101 ~]# useradd tangseng
```

```
[root@hadoop101 ~]# ll /home/
```

## 7.4.2 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) 案例实操

(1) 设置用户的密码

```
[root@hadoop101 ~]# passwd tangseng
```

## 7.4.3 id 查看用户是否存在

### 1) 基本语法:

`id 用户名`

### 2) 案例实操:

(1) 查看用户是否存在

```
[root@hadoop101 ~]# id tangseng
```

## 7.4.4 cat /etc/passwd 查看创建了哪些用户

### 1) 基本语法:

```
[root@hadoop101 ~]# cat /etc/passwd
```

## 7.4.5 su 切换用户

su: swith user 切换用户

### 1) 基本语法:

`su 用户名称` (功能描述: 切换用户, 只能获得用户的执行权限, 不能获得环境变量)

`su - 用户名称` (功能描述: 切换到用户并获得该用户的环境变量及执行权限)

### 2) 案例实操

(1) 切换用户

```
[root@hadoop101 ~]# su tangseng
```

```
[root@hadoop101 ~]# echo $PATH
```

```
/usr/lib64/qt-3.3/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
```

```
[root@hadoop101 ~]# exit
```

```
[root@hadoop101 ~]# su - tangseng
```

```
[root@hadoop101 ~]# echo $PATH
```

```
/usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/tangseng/
```

bin

## 7.4.5 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) 选项说明

选项	功能
-r	删除用户的同时, 删除与用户相关的所有文件。

### 3) 案例实操:

- (1) 删除用户但保存用户主目录

```
[root@hadoop101 ~]#userdel tangseng
```

```
[root@hadoop101 ~]#ll /home/
```

- (2) 删除用户和用户主目录, 都删除

```
[root@hadoop101 ~]#useradd zhubajie
```

```
[root@hadoop101 ~]#ll /home/
```

```
[root@hadoop101 ~]#userdel -r zhubajie
```

```
[root@hadoop101 ~]#ll /home/
```

## 7.4.6 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)

### 2) 案例实操

- (1) 显示自身用户名称

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

- (2) 显示登录用户的用户名

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

## 7.4.7 sudo 设置普通用户具有 root 权限

1) 添加 atguigu 用户，并对其设置密码。

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

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

2) 修改配置文件

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

修改 /etc/sudoers 文件，找到下面一行(91 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
```

或者配置成采用 sudo 命令时，不需要输入密码

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

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

3) 案例实操

(1) 用普通用户在 /opt 目录下创建一个文件夹

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

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

## 7.4.9 usermod 修改用户

1) 基本语法:

```
usermod -g 用户组 用户名
```

2) 选项说明

选项	功能
-g	修改用户的初始登录组，给定的组必须存在。默认组id是1。

3) 案例实操:

(1) 将用户加入到用户组

```
[root@hadoop101 opt]#usermod -g root zhubajie
```

## 7.5 用户组管理命令

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。

### 7.5.1 groupadd 新增组

#### 1) 基本语法

`groupadd 组名`

#### 2) 案例实操:

(1) 添加一个 `xitianqujing` 组

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

### 7.5.2 groupdel 删除组

#### 1) 基本语法:

`groupdel 组名`

#### 2) 案例实操

(1) 删除 `xitianqujing` 组

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

### 7.5.3 groupmod 修改组

#### 1) 基本语法:

`groupmod -n 新组名 老组名`

#### 2) 选项说明

选项	功能描述
<code>-n&lt;新组名&gt;</code>	指定工作组的新组名

#### 3) 案例实操

(1) 修改 `atguigu` 组名称为 `atguigu1`

```
[root@hadoop101 ~]#groupadd xitianqujing
```

```
[root@hadoop101 ~]# groupmod -n xitian xitianqujing
```

## 7.5.4 cat /etc/group 查看创建了哪些组

1) 基本操作

```
[root@hadoop101 atguigu]# cat /etc/group
```

## 7.6 文件权限类

### 7.6.1 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：

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
<b>d</b>	<b>rwX</b>	<b>r-X</b>	<b>r-X</b>
目录文件	读 写 执行	读 写 执行	读 写 执行

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

l 链接文档(link file);

(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

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。

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：[尚硅谷官网](#)

[ 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) 案例实操

```
[root@hadoop101 ~]# ll
```

```
总用量 104
```

```
-rw-----. 1 root root 1248 1月 8 17:36 anaconda-ks.cfg
```

```
drwxr-xr-x. 2 root root 4096 1月 12 14:02 dssz
```

```
lrwxrwxrwx. 1 root root 20 1月 12 14:32 houzi ->
```

```
xiyou/dssz/houge.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]#
```

(1) 如果查看到是文件: 链接数指的是硬链接个数。创建硬链接方法

```
ln [原文件] [目标文件]
```

```
[root@hadoop101 ~]# ln xiyou/dssz/houge.txt ./hg.txt
```

(2) 如果查看的是文件夹: 链接数指的是子文件夹个数。

```
[root@hadoop101 ~]# ls -al xiyou/
```

```
总用量 16
```

```
drwxr-xr-x. 4 root root 4096 1月 12 14:00 .
```

```
dr-xr-x---. 29 root root 4096 1月 12 14:32 ..
```

```
drwxr-xr-x. 2 root root 4096 1月 12 14:30 dssz
```

```
drwxr-xr-x. 2 root root 4096 1月 12 14:04 mingjie
```

## 7.6.2 chmod 改变权限

1) 基本语法:

文件类型 0	属主 u 1 2 3	属组 g 4 5 6	其他用户 o 7 8 9
d 目录文件	rwX 读 写 执行	r-X 读 写 执行	r-X 读 写 执行

第一种方式变更权限

```
chmod [{u go a}]{+ -=}{rwx} 文件或目录
```

第二种方式变更权限

```
chmod [mode=421] [文件或目录]
```

## 2) 经验技巧

u:所有者 g:所有组 o:其他人 a:所有人(u、g、o 的总和)

r=4 w=2 x=1          rwx=4+2+1=7

## 3) 案例实操

(1) 修改文件使其所属主用户具有执行权限

```
[root@hadoop101 ~]# cp xiyou/dssz/houge.txt ./
```

```
[root@hadoop101 ~]# chmod u+x houge.txt
```

(2) 修改文件使其所属组用户具有执行权限

```
[root@hadoop101 ~]# chmod g+x houge.txt
```

(3) 修改文件所属主用户执行权限,并使其他用户具有执行权限

```
[root@hadoop101 ~]# chmod u-x,o+x houge.txt
```

(4) 采用数字的方式, 设置文件所有者、所属组、其他用户都具有可读可写可执行权限。

```
[root@hadoop101 ~]# chmod 777 houge.txt
```

(5) 修改整个文件夹里面的所有文件的所有者、所属组、其他用户都具有可读可写可执行权限。

```
[root@hadoop101 ~]# chmod -R 777 xiyou/
```

## 7.6.3 chown 改变所有者

### 1) 基本语法:

```
chown [选项] [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) 选项说明

选项	功能
-R	递归操作

3) 案例实操

(1) 修改文件所有者

```
[root@hadoop101 ~]# chown atguigu hougex.txt
[root@hadoop101 ~]# ls -al
-rwxrwxrwx. 1 atguigu root 551 5月 23 13:02 hougex.txt
```

(2) 递归改变文件所有者和所有组

```
[root@hadoop101 xiyou]# ll
drwxrwxrwx. 2 root root 4096 9月 3 21:20 xiyou
[root@hadoop101 xiyou]# chown -R atguigu:atguigu xiyou/
[root@hadoop101 xiyou]# ll
drwxrwxrwx. 2 atguigu atguigu 4096 9月 3 21:20 xiyou
```

## 7.6.4 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) 案例实操

(1) 修改文件的所属组

```
[root@hadoop101 ~]# chgrp root hougex.txt
[root@hadoop101 ~]# ls -al
-rwxrwxrwx. 1 atguigu root 551 5月 23 13:02 hougex.txt
```

## 7.7 搜索查找类

### 7.7.1 find 查找文件或者目录

`find` 指令将从指定目录向下递归地遍历其各个子目录, 将满足条件的文件显示在终端。

1) 基本语法:

`find [搜索范围] [选项]`

## 2) 选项说明

选项	功能
-name<查询方式>	按照指定的文件名查找模式查找文件
-user<用户名>	查找属于指定用户名所有文件
-size<文件大小>	按照指定的文件大小查找文件。

## 3) 案例实操

(1) 按文件名：根据名称查找/目录下的filename.txt文件。

```
[root@hadoop101 ~]# find xiyou/ -name *.txt
```

(2) 按拥有者：查找/opt目录下，用户名称为-user的文件

```
[root@hadoop101 ~]# find xiyou/ -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@hadoop101 ~]# find /home -size +204800
```

## 7.7.2 locate 快速定位文件路径

locate 指令利用事先建立的系统中所有文件名称及路径的 locate 数据库实现快速定位给定的文件。Locate 指令无需遍历整个文件系统，查询速度较快。为了保证查询结果的准确度，管理员必须定期更新 locate 时刻。

## 1) 基本语法

locate 搜索文件

## 2) 经验技巧

由于 locate 指令基于数据库进行查询，所以第一次运行前，必须使用 updatedb 指令创建 locate 数据库。

## 3) 案例实操

(1) 查询文件夹

```
[root@hadoop101 ~]# updatedb
```

```
[root@hadoop101 ~]# locate tmp
```

## 7.7.3 grep 过滤查找及“|”管道符

0) 管道符，“|”，表示将前一个命令的处理结果输出传递给后面的命令处理

## 1) 基本语法

grep 选项 查找内容 源文件

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：[尚硅谷官网](#)

## 2) 选项说明

选项	功能
-n	显示匹配行及行号。

## 3) 案例实操

(1) 查找某文件在第几行

```
[root@hadoop101 ~]# ls | grep -n test
```

## 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) 经验技巧:

(1) 只能压缩文件不能压缩目录

(2) 不保留原来的文件

## 3) 案例实操

(1) gzip压缩

```
[root@hadoop101 ~]# ls
test.java
[root@hadoop101 ~]# gzip houg.txt
[root@hadoop101 ~]# ls
houge.txt.gz
```

(2) gunzip解压缩文件

```
[root@hadoop101 ~]# gunzip houg.txt.gz
[root@hadoop101 ~]# ls
houge.txt
```

### 7.8.2 zip/unzip 压缩

## 1) 基本语法:

zip [选项] XXX.zip 将要压缩的内容 (功能描述: 压缩文件和目录的命令)

unzip [选项] XXX.zip

(功能描述: 解压缩文件)

## 2) 选项说明

zip 选项	功能
-r	压缩目录

unzip选项	功能
-d<目录>	指定解压后文件的存放目录

## 3) 经验技巧

zip 压缩命令在window/linux都通用, 可以压缩目录且保留源文件。

## 4) 案例实操:

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

```
[root@hadoop101 opt]# touch bailongma.txt
[root@hadoop101 ~]# zip houma.zip houg.txt bailongma.txt
adding: houg.txt (stored 0%)
adding: bailongma.txt (stored 0%)
[root@hadoop101 opt]# ls
houg.txt bailongma.txt houma.zip
```

(2) 解压 mypackage.zip

```
[root@hadoop101 ~]# unzip houma.zip
Archive: houma.zip
extracting: houg.txt
extracting: bailongma.txt
[root@hadoop101 ~]# ls
houg.txt bailongma.txt houma.zip
```

(3) 解压mypackage.zip到指定目录-d

```
[root@hadoop101 ~]# unzip houma.zip -d /opt
[root@hadoop101 ~]# ls /opt/
```

## 7.8.3 tar 打包

### 1) 基本语法:

更多 [Java](#) -[大数据](#) -[前端](#) -[python](#) 人工智能资料下载, 可百度访问: [尚硅谷官网](#)

`tar [选项] 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)

## 2) 选项说明

选项	功能
-c	产生.tar 打包文件
-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

## 3) 案例实操

### (1) 压缩多个文件

```
[root@hadoop101 opt]# tar -zcvf houma.tar.gz houg.txt bailongma.txt
houg.txt
bailongma.txt
[root@hadoop101 opt]# ls
houma.tar.gz houg.txt bailongma.txt
```

### (2) 压缩目录

```
[root@hadoop101 ~]# tar -zcvf xiyou.tar.gz xiyou/
xiyou/
xiyou/mingjie/
xiyou/dssz/
xiyou/dssz/houg.txt
```

### (3) 解压到当前目录

```
[root@hadoop101 ~]# tar -zxvf houma.tar.gz
```

### (4) 解压到指定目录

```
[root@hadoop101 ~]# tar -zxvf xiyou.tar.gz -C /opt
[root@hadoop101 ~]# ll /opt/
```

## 7.9 磁盘分区类

### 7.9.1 df 查看磁盘空间使用情况

df: disk free 空余硬盘

#### 1) 基本语法:

`df` 选项 (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)

#### 2) 选项说明

选项	功能
<code>-h</code>	以人们较易阅读的 GBytes, MBytes, KBytes 等格式自行显示;

#### 3) 案例实操

(1) 查看磁盘使用情况

```
[root@hadoop101 ~]# 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
```

### 7.9.2 fdisk 查看分区

#### 1) 基本语法:

`fdisk -l` (功能描述: 查看磁盘分区详情)

#### 2) 选项说明

选项	功能
<code>-l</code>	显示所有硬盘的分区列表

#### 3) 经验技巧:

该命令必须在 `root` 用户下才能使用

#### 4) 功能说明:

(1) Linux 分区

**Device:** 分区序列

**Boot:** 引导

Start: 从X磁柱开始

End: 到Y磁柱结束

Blocks: 容量

Id: 分区类型ID

System: 分区类型

## (2) Win7分区



## 5) 案例实操

### (1) 查看系统分区情况

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

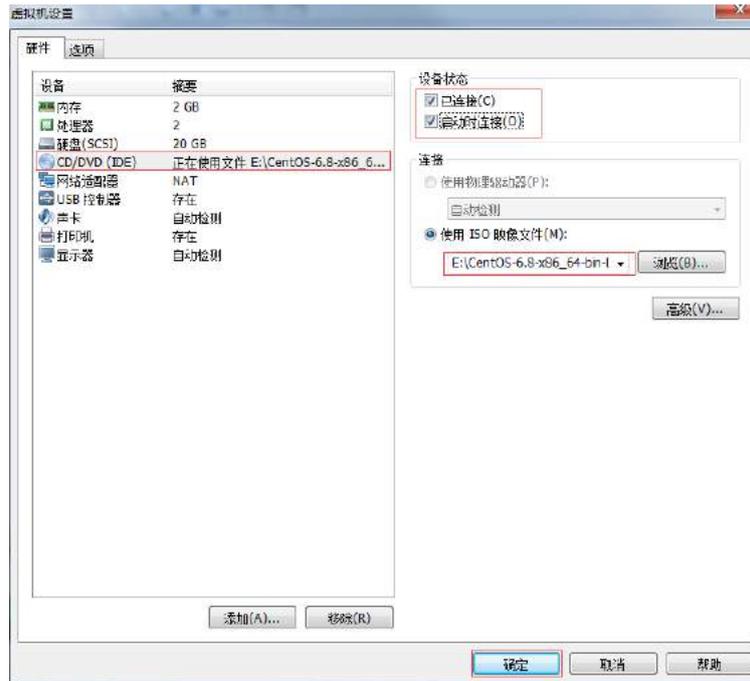
### 7.9.3 mount/umount 挂载/卸载

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，它在用一种叫做“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) 基本语法:

`mount [-t vfstype] [-o options] device dir` (功能描述: 挂载设备)

`umount` 设备文件名或挂载点 (功能描述: 卸载设备)

### 2) 参数说明

参数	功能
-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 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
-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

device	要挂载(mount)的设备
dir	设备在系统上的挂载点(mount point)

## 2) 案例实操

(1) 挂载光盘镜像文件

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

建立挂载点

```
[root@hadoop101 ~]# 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/
```

(2) 卸载光盘镜像文件

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

## 3) 设置开机自动挂载:

```
[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
```

## 7.10 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。

### 7.10.1 ps 查看当前系统进程状态

ps:process status 进程状态

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载, 可百度访问: [尚硅谷官网](#)

## 1) 基本语法:

`ps aux | grep xxx` (功能描述: 查看系统中所有进程)

`ps -ef | grep xxx` (功能描述: 可以查看子父进程之间的关系)

## 2) 选项说明

选项	功能
-a	选择所有进程
-u	显示所有用户的所有进程
-x	显示没有终端的进程

## 3) 功能说明

(1) `ps aux` 显示信息说明

**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

(2) `ps -ef` 显示信息说明

**UID:** 用户 ID

**PID:** 进程 ID

**PPID:** 父进程 ID

**C:** CPU 用于计算执行优先级的因子。数值越大, 表明进程是 CPU 密集型运算, 执行优先级会降低; 数值越小, 表明进程是 I/O 密集型运算, 执行优先级会提高

STIME: 进程启动的时间

TTY: 完整的终端名称

TIME: CPU 时间

CMD: 启动进程所用的命令和参数

#### 4) 经验技巧:

如果想查看进程的 CPU 占用率和内存占用率, 可以使用 `aux`;

如果想查看进程的父进程 ID 可以使用 `ef`;

#### 5) 案例实操

```
[root@hadoop101 datas]# ps aux
```

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]

```
[root@hadoop101 datas]# ps -ef
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	10:26	?	00:00:01	/sbin/init
root	2	0	0	10:26	?	00:00:00	[kthreadd]
root	3	2	0	10:26	?	00:00:00	[migration/0]
root	4	2	0	10:26	?	00:00:00	[ksoftirqd/0]
root	5	2	0	10:26	?	00:00:00	[stopper/0]
root	6	2	0	10:26	?	00:00:00	[watchdog/0]
root	7	2	0	10:26	?	00:00:00	[migration/1]
root	8	2	0	10:26	?	00:00:00	[stopper/1]
root	9	2	0	10:26	?	00:00:00	[ksoftirqd/1]

## 7.10.2 kill 终止进程

#### 1) 基本语法:

`kill [选项] 进程号` (功能描述: 通过进程号杀死进程)

`killall 进程名称` (功能描述: 通过进程名称杀死进程, 也支持通配符, 这在系统因负载过大而变得很慢时很有用)

#### 2) 选项说明

选项	功能
-9	表示强迫进程立即停止

#### 3) 案例实操:

(1) 杀死浏览器进程

```
[root@hadoop101 桌面]# kill -9 5102
```

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载, 可百度访问: [尚硅谷官网](#)

(2) 通过进程名称杀死进程

```
[root@hadoop101 桌面]# killall firefox
```

### 7.10.3 pstree 查看进程树

1) 基本语法:

```
pstree [选项]
```

2) 选项说明

选项	功能
-p	显示进程的 PID
-u	显示进程的所属用户

3) 案例实操:

(1) 显示进程 pid

```
[root@hadoop101 datas]# pstree -p
```

(2) 显示进程所属用户

```
[root@hadoop101 datas]# pstree -u
```

### 7.10.4 top 查看系统健康状态

1) 基本命令

```
top [选项]
```

2) 选项说明

选项	功能
-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。

3) 操作说明:

操作	功能
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
---	--------

#### 4) 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

### 5) 案例实操

```
[root@hadoop101 atguigu]# top -d 1
```

```
[root@hadoop101 atguigu]# top -i
```

```
[root@hadoop101 atguigu]# top -p 2575
```

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。

## 7.10.5 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）

### 2) 选项说明

选项	功能
-an	按一定顺序排列输出
-p	表示显示哪个进程在调用

### 3) 案例实操

(1) 通过进程号查看该进程的网络信息

```
[root@hadoop101 hadoop-2.7.2]# netstat -anp | grep 火狐浏览器进程号
```

```
udp 0 0 192.168.1.101:33893 192.168.1.2:53 ESTABLISHED 4043/firefox
```

```
udp 0 0 192.168.1.101:47416 192.168.1.2:53 ESTABLISHED 4043/firefox
```

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：[尚硅谷官网](#)

```

unix 2      [ ACC ]     STREAM      LISTENING   28916  4043/firefox
/tmp/orbit-atguigu/linc-fcb-0-382f8b667a38a

unix 3      [ ]          STREAM      CONNECTED   28919  4043/firefox
/tmp/orbit-atguigu/linc-fcb-0-382f8b667a38a
    
```

## 7.11 crond 系统定时任务

### 7.11.1 crond 服务管理

- 1) 重新启动 crond 服务

```
[root@hadoop101 ~]# service crond restart
```

### 7.11.2 crontab 定时任务设置

- 1) 基本语法

```
crontab [选项]
```

- 2) 选项说明

选项	功能
-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

- 3) 参数说明

```
[root@hadoop101 ~]# 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	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。

### 4) 案例实操:

(1) 每隔 1 分钟，向/root/bailongma.txt 文件中添加一个 11 的数字

```
*/1 * * * * /bin/echo "11" >> /root/bailongma.txt
```

## 第 8 章 软件包管理

### 8.1 RPM

#### 8.1.1 RPM 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

#### 8.1.2 RPM 查询命令 (rpm -qa)

1) 基本语法:

`rpm -qa` (功能描述: 查询所安装的所有 rpm 软件包)

2) 经验技巧:

由于软件包比较多，一般都会采取过滤。`rpm -qa | grep rpm`软件包

3) 案例实操

(1) 查询firefox软件安装情况

```
[root@hadoop101 Packages]# rpm -qa |grep firefox
```

```
firefox-45.0.1-1.el6.centos.x86_64
```

#### 8.1.3 RPM 卸载命令 (rpm -e)

1) 基本语法:

(1) `rpm -e RPM软件包`

(2) `rpm -e --nodeps 软件包`

2) 选项说明

选项	功能
-e	卸载软件包

<code>--nodeps</code>	卸载软件时，不检查依赖。这样的话，那些使用该软件包的软件在此之后可能就不能正常工作了。
-----------------------	---

### 3) 案例实操

(1) 卸载firefox软件

```
[root@hadoop101 Packages]# rpm -e firefox
```

## 8.1.4 RPM 安装命令 (rpm -ivh)

### 1) 基本语法:

`rpm -ivh` RPM 包全名

### 2) 选项说明

选项	功能
<code>-i</code>	<code>-i=install</code> , 安装
<code>-v</code>	<code>-v=verbose</code> , 显示详细信息
<code>-h</code>	<code>-h=hash</code> , 进度条
<code>--nodeps</code>	<code>--nodeps</code> , 不检测依赖进度

### 3) 案例实操

(1) 安装firefox软件

```
[root@hadoop101 Packages]# pwd
```

```
/media/CentOS_6.8_Final/Packages
```

```
[root@hadoop101 Packages]# rpm -ivh firefox-45.0.1-1.el6.centos.x86_64.rpm
```

```
warning: firefox-45.0.1-1.el6.centos.x86_64.rpm: Header V3 RSA/SHA1 Signature, key ID
c105b9de: NOKEY
```

```
Preparing... ##### [100%]
```

```
1:firefox #####
```

```
[100%]
```

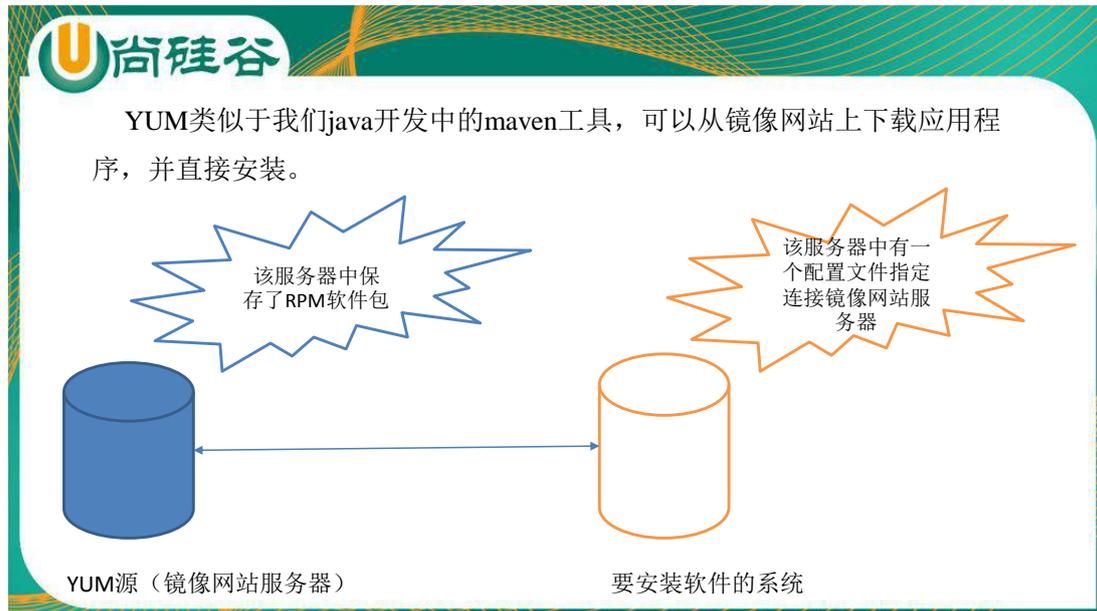
## 8.2 YUM 仓库配置

### 8.2.1 YUM 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

更多 [Java - 大数据 - 前端 - python 人工智能资料下载](#)，可百度访问：[尚硅谷官网](#)

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。



## 8.2.2 YUM 的常用命令

1) 基本语法:

```
yum [选项] [参数]
```

2) 选项说明

选项	功能
-y	对所有提问都回答“yes”

3) 参数说明

参数	功能
install	安装 rpm 软件包
update	更新 rpm 软件包
check-update	检查是否有可用的更新 rpm 软件包
remove	删除指定的 rpm 软件包
list	显示软件包信息
clean	清理 yum 过期的缓存
deplist	显示 yum 软件包的所有依赖关系

4) 案例实操实操

(1) 采用 yum 方式安装 tree 软件

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：[尚硅谷官网](#)

```
[root@hadoop101 ~]#yum -y install tree
```

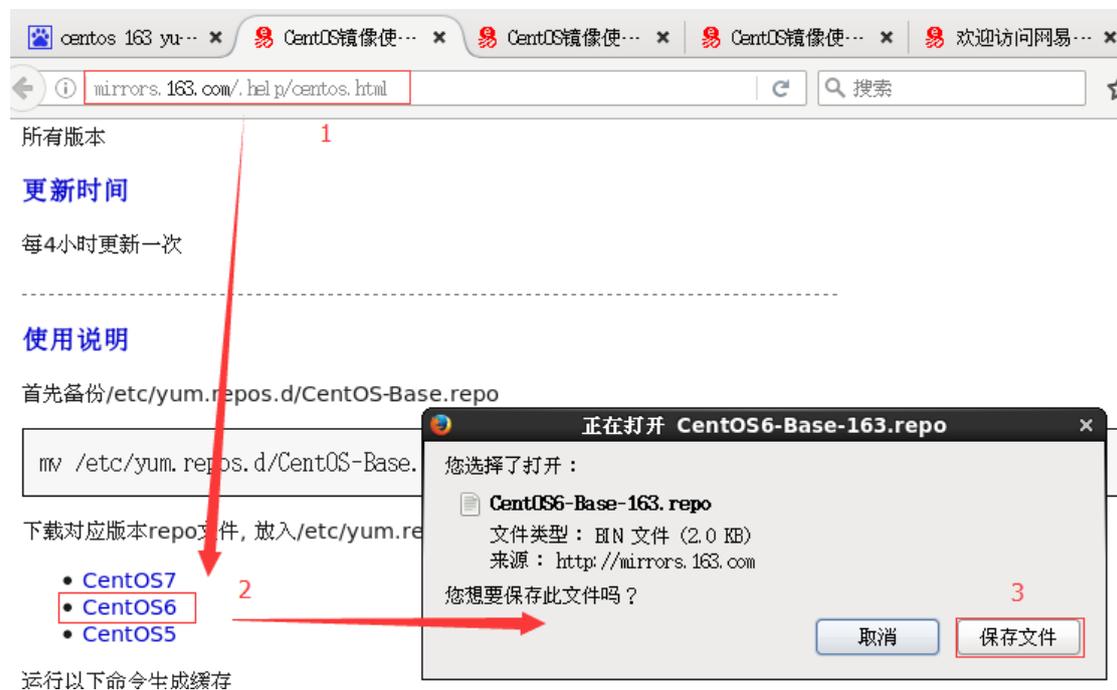
### 8.2.3 修改网络 YUM 源

默认的系统 YUM 源，需要连接国外 apache 网站，网速比较慢，可以修改关联的网络 YUM 源为国内镜像的网站，比如网易 163。

#### 1) 前期文件准备

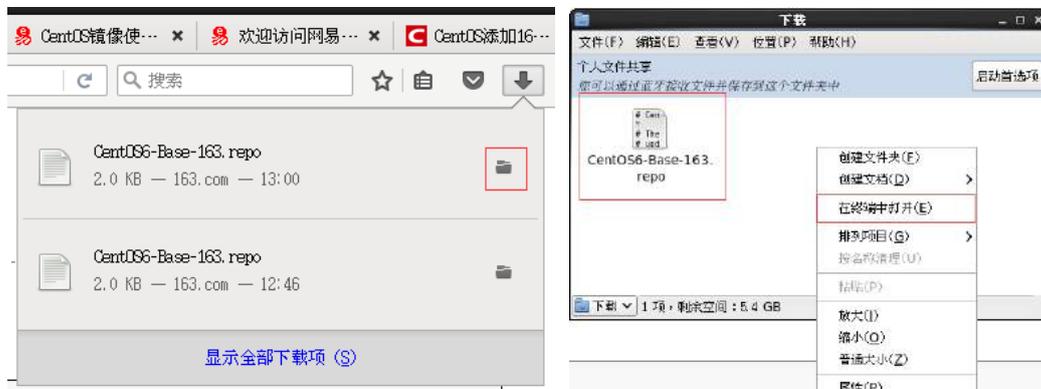
(1) 前提条件 linux 系统必须可以联网

(2) 在 Linux 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



运行以下命令生成缓存

#### (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
```

更多 Java - 大数据 - 前端 - python 人工智能资料下载，可百度访问：[尚硅谷官网](#)

/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
```

```
/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

(3) [root@hadoop101 yum.repos.d]# yum -y install createrepo

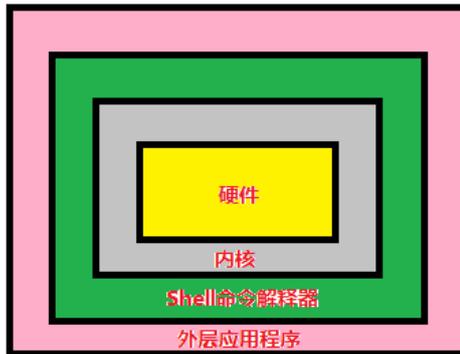
## 4) 测试

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

## 第 9 章 Shell 编程

### 9.1 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。

### 9.2 Shell 脚本的执行方式

#### 2) 脚本格式

- (1) 脚本以#!/bin/bash 开头
- (2) 脚本必须有可执行权限

#### 3) 第一个 Shell 脚本

- (1) 需求：创建一个 Shell 脚本，输出 helloworld
- (2) 实操：

```
[atguigu@hadoop101 datas]$ touch helloworld.sh  
[atguigu@hadoop101 datas]$ vi helloworld.sh
```

在 helloworld.sh 中输入如下内容

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

#### 4) 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@hadoop101 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
```

## 9.3 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。

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

3) 显示当前 shell 中所有变量：set

### 9.3.1 定义变量

1) 基本语法：

(1) 定义变量：变量=值

(2) 撤销变量：unset 变量

(3) 声明静态变量：readonly 变量，注意：不能 unset

2) 变量定义规则 0

(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

3) 案例实操

(1) 定义变量 A

```
A=8
```

(2) 撤销变量A

```
unset A
```

(3) 声明静态的变量B=2，不能unset

```
readonly B=2
```

(4) 可把变量提升为全局环境变量，可供其他shell程序使用

```
export 变量名
```

### 9.3.2 将命令的返回值赋给变量

- 1) `A=`ls -la`` 反引号，运行里面的命令，并把结果返回给变量 A
- 2) `A=$(ls -la)` 等价于反引号

### 9.3.3 设置环境变量

- 1) 基本语法:

- (1) `export 变量名=变量值` (功能描述: 将 shell 变量输出为环境变量)
- (2) `source 配置文件` (功能描述: 让修改后的配置信息立即生效)
- (3) `echo $变量名` (功能描述: 查询环境变量的值)

- 2) 案例实操:

- (1) 在 `/etc/profile` 文件中定义 `JAVA_HOME` 环境变量

```
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

```
export PATH=$PATH:$JAVA_HOME/bin
```

- (2) 查看环境变量 `JAVA_HOME` 的值

```
[atguigu@hadoop101 datas]$ echo $JAVA_HOME
```

```
/opt/module/jdk1.8.0_144
```

### 9.3.4 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) 案例实操

- (1) 输出输入的的参数 1, 参数 2, 所有参数, 参数个数

```
#!/bin/bash
echo "$0 $1 $2"
echo "$*"
echo "$@"
echo "$#"
```

- (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$х 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

### 9.3.5 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) 案例实操

```
#!/bin/bash
echo "$$"
./helloworld.sh &
echo "$!"
echo "$?"
```

## 9.4 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

(3) `expr m - n`

(4) `expr \*,/,%` 乘, 除, 取余

2) 案例实操: 计算  $(2+3) \times 4$  的值

(1) 采用 `$(运算式)` 方式

```
[root@hadoop101 datas]# S=$(2+3)*4
```

```
[root@hadoop101 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
```

## 9.5 条件判断

### 9.5.1 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) 案例实操:

```
[atguigu] 返回 true
```

```
[] 返回 false
```

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

### 9.5.2 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) 案例实操

(1) 23 是否大于等于 22

```
[ 23 -ge 22 ]
```

(2) student.txt 是否具有写权限

```
[ -w student.txt ]
```

(3) /root/install.log 目录中的文件是否存在

```
[ -e /root/install.log ]
```

## 9.6 流程控制

### 9.6.1 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) 案例实操

```
#!/bin/bash

if [ $1 -eq "123" ]
then
    echo "123"
elif [ $1 -eq "456" ]
then
    echo "456"
fi
```

## 9.6.2 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) 案例实操

```
#!/bin/bash

case $1 in
    "1")
        echo "1"
        ;;
    "2")
        echo "2"
```

```
;;
*)
    echo "other"
;;
esac
```

### 9.6.3 for 循环

#### 1) 基本语法 1:

```
for 变量 in 值1 值2 值3...
```

```
do
```

```
    程序
```

```
done
```

#### 2) 案例实操:

##### (1) 打印输入参数

```
#!/bin/bash
#打印数字

for i in "$*"
do
    echo "The num is $i "
done

for j in "$@"
do
    echo "The num is $j"
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) 案例实操

##### (1) 从 1 加到 100

```
#!/bin/bash

s=0
for((i=0;i<=100;i++))
```

```
do
    s=${s+$i}
done
echo "$s"
```

## 9.6.4 while 循环

### 1) 基本语法:

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

### 2) 案例实操

#### (1) 从 1 加到 100

```
#!/bin/bash
s=0
i=1
while [ $i -le 100 ]
do
    s=${s+$i}
    i=$((i+1))
done
echo $s
```

## 9.7 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) 案例实操

读取控制台输入的名称

```
#!/bin/bash
```

```
read -t 7 -p "please 7 miao input your name " NAME
echo $NAME
```

## 9.8 函数

### 9.8.1 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) 案例实操

```
[atguigu@hadoop101 opt]$ basename /opt/test.txt
```

```
test.txt
```

```
[atguigu@hadoop101 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) 案例实操

```
[atguigu@hadoop101 opt]$ dirname /opt/test.txt
```

```
/opt
```

### 9.8.2 自定义函数

#### 1) 基本语法：

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

## 2) 经验技巧:

(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)`

## 3) 案例实操

### (1) 计算输入参数的和

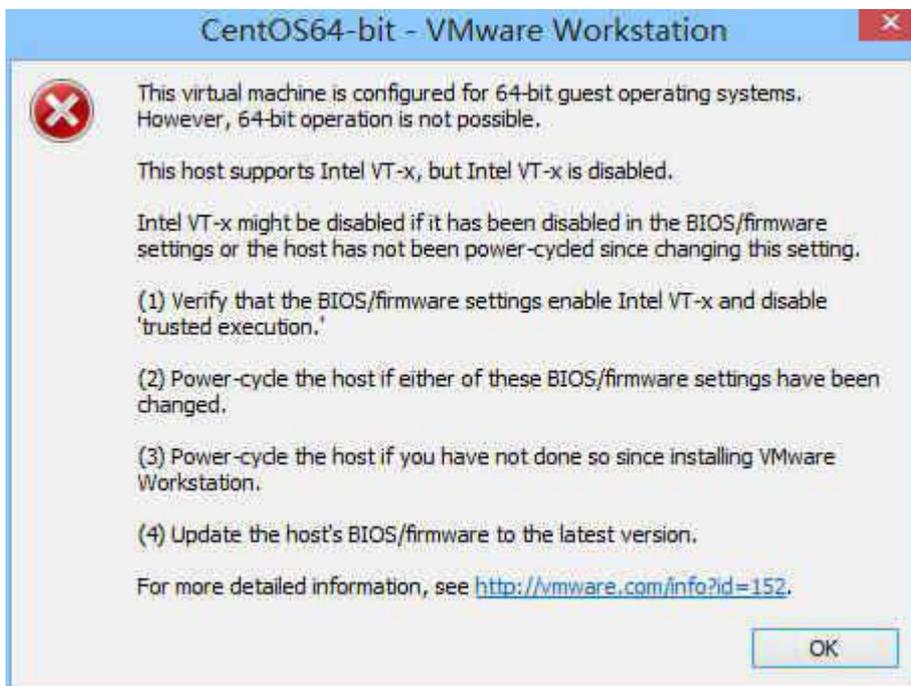
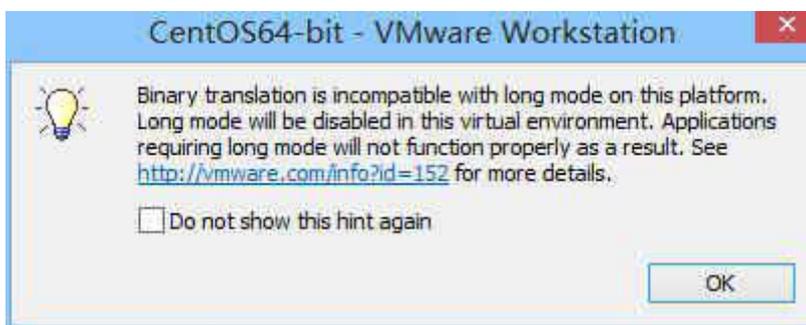
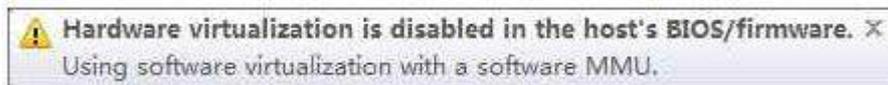
```
#!/bin/bash
function sum()
{
    s=0
    s=$(( $1 + $2 ))
    echo "$s"
}

read -p "Please input the number1: " n1;
read -p "Please input the number2: " n2;
sum $n1 $n2;
```

## 第 10 章 常见错误及解决方案

1) 虚拟化支持异常情况如下几种情况

```
Loading vmlinuz.....
Loading initrd.img.....ready.
This kernel requires an x86-64 CPU, but only detected an i686 CPU.
Unable to boot - please use a kernel appropriate for your CPU.
```



问题原因：宿主机 BIOS 设置中的硬件虚拟化被禁用了

解决办法：需要打开笔记本 BIOS 中的 IVT 对虚拟化的支持

