# Linux简介

## Linux学习大纲

1.Linux操作系统管理（Redhat、CentOS）

2.Linux常见服务器管理（Apache、tomacat、oracle、sendmail、nginx、nagios、zabbix）

3.Linux系统安全、集群、存储、调优

4.Linux桌面虚拟化、企业级虚拟化及Openstack私有云平台搭建。

## UNINX的诞生

1969年UNIX操作系统诞生，它由美国贝尔实验室的Ken Thompson（肯\*汤普森）、Dennis  
Ritchie（丹尼斯\*里奇）发明。

## C语言的诞生

* 1972年期间C语言诞生。
* 1973年，UNIX用C语言改写完成。UNIX第一代是用汇编语言编写的。
* 最为著名的有加州大学伯克莱（伯克利）分校的BSD系统。
* 从1990AT&T认识到了UNIX价值，因此他起诉包括伯克利在内的很多厂商。IBM、HP、SUN。
* 伯克利不得不推出不包含任何AT&T源代码的4.4 free BSD Lite.
* 1991年，这时候**Linux**系统正式发布。

**理查德斯托尔曼**

自由软件运动的精神领袖、GNU计划以及自由软件基金会 ( Free Software Foundation ）的创立者、著名黑客。

•骇客

GNU组织

* GNU组织：<http://www.gnu.org>
* GNU计划是由RiChard Stallman公开发起。它的目标是创建一套完全自由的操作系统。
* GNU是 "GNU’s Not Unix” 的缩写。
* "free"指的是自由（freedom），而不是价格。

**林纳斯·本纳第克特·托瓦兹（Linus Benedict Torvalds）**

开发了Linux内核

**Linux内核网站：**<https://www.kernel.org/>

## 为什么学习Linux

Linux开源系统是全球互联网公司首选的解决方案。90%公司都会使用Linux.

## Linux各发行版本统计数据和排名

<https://distrowatch.com/>

**redhat**

红帽企业系统(Red Hat Enterprise Linux,RHEL.)

全球最大的开源技术厂商，提供性能与稳定性极强的Linux套件系统并拥有完善的全球技术支持。

**Centos**

社区企业操作系统（Centos）

最初是将红帽子企业系统“重新编译/发布”给用户免费使用而广泛使用。

当前已经正式加入红帽公司并继续保持免费，随RHEL更新而更新。

已被RedHat收购，继续免费。

**红帽用户桌面版（Fedora [LInux]）**

最初由红帽公司发起的桌面版系统套件(也具有企业版)

用户可以免费体验到最新的技术和工具，而功能成熟后加入到RHEL中。

**三者之间安装软件包的方式一样，都是rpm和yum.**

## RedHat认证体系

工程师

架构师

安全专家

# 使用VM安装对应的操作系统

基本步骤：

1. 安装VMware
2. 在VMware中创建虚拟机
3. 在VMware中开启创建的虚拟机（安装对应操作系统）

## 安装虚拟机VMware

1.下载并安装VMware

<https://www.zdfans.com/html/5928.html>

2.下载对应操作系统的镜像（用于安装对应的操作系统）

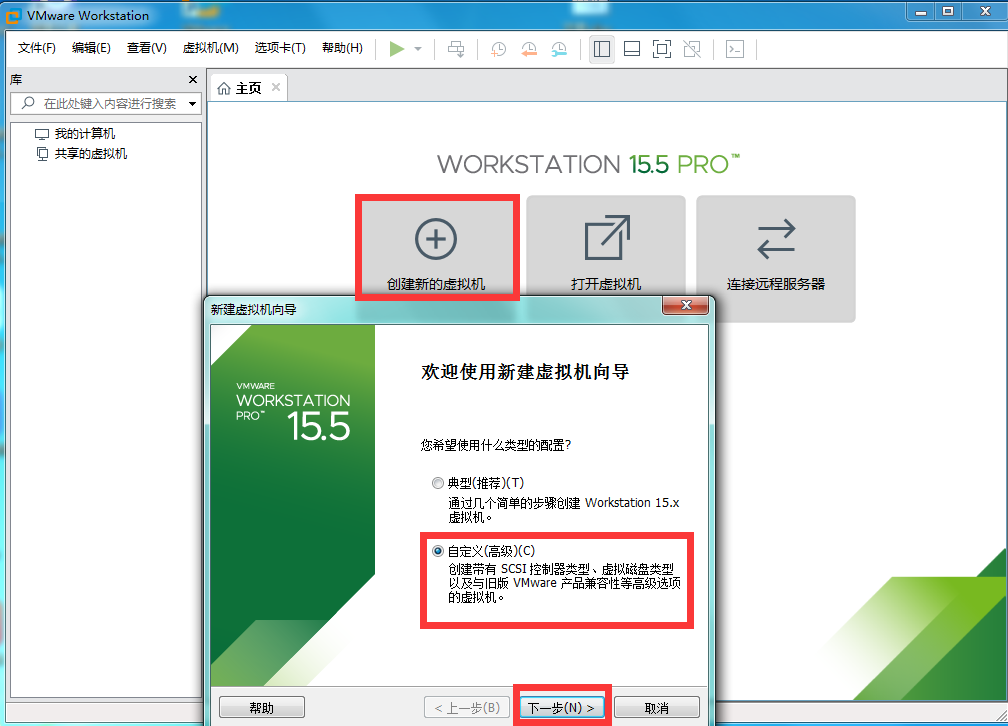
## 在VMware中创建虚拟机

准备工作：

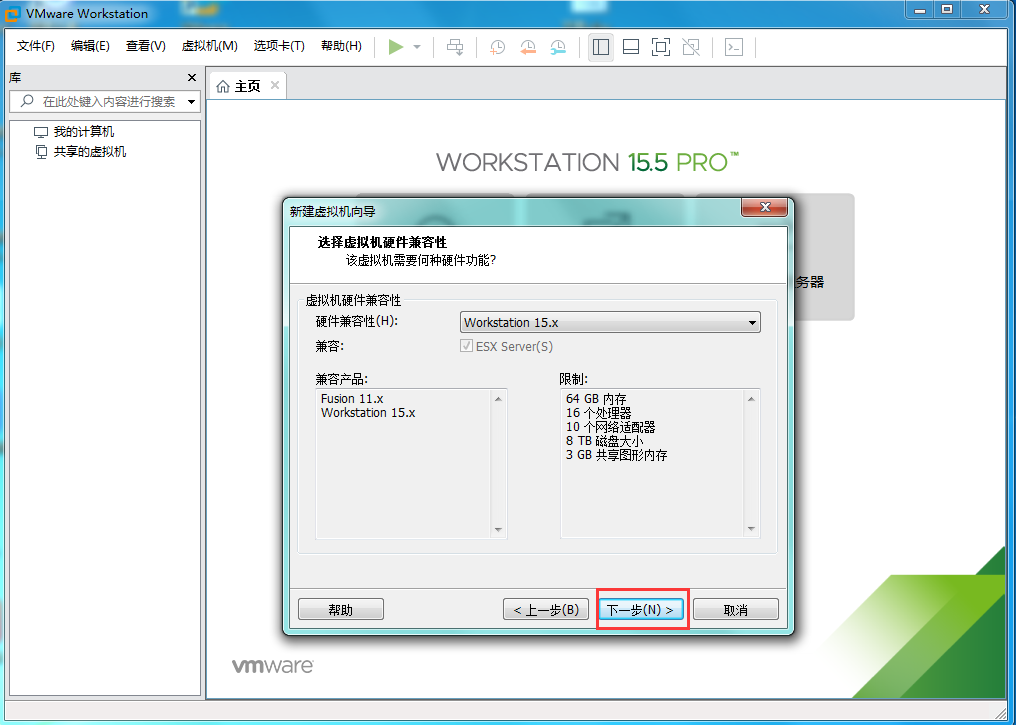
1. 下载对应的操作系统的镜像

安装步骤：

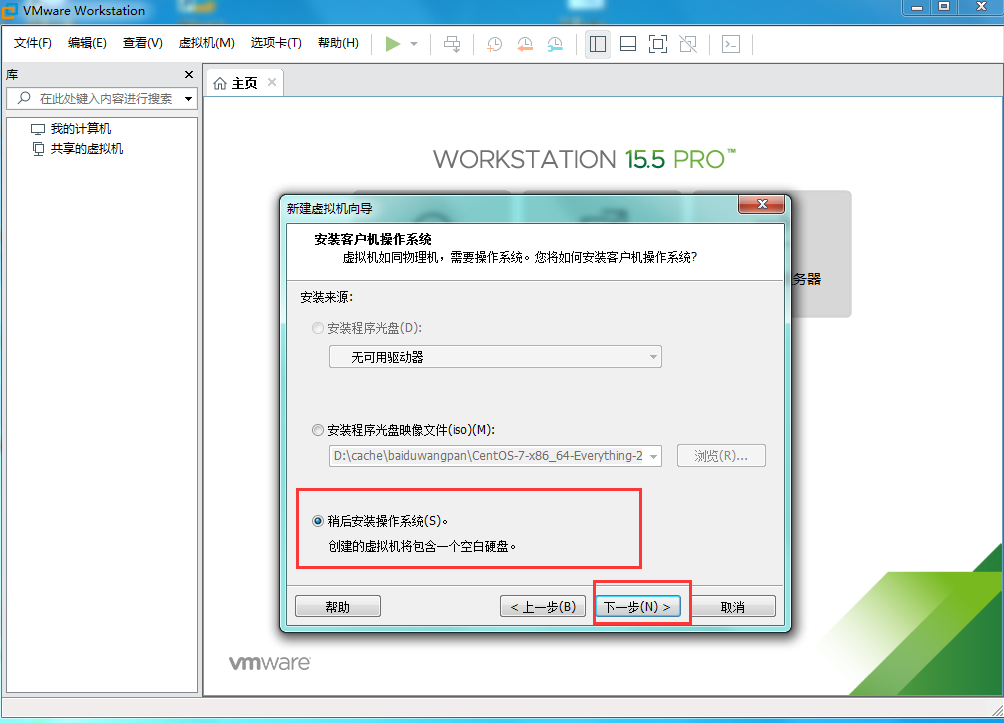
1. 创建虚拟机



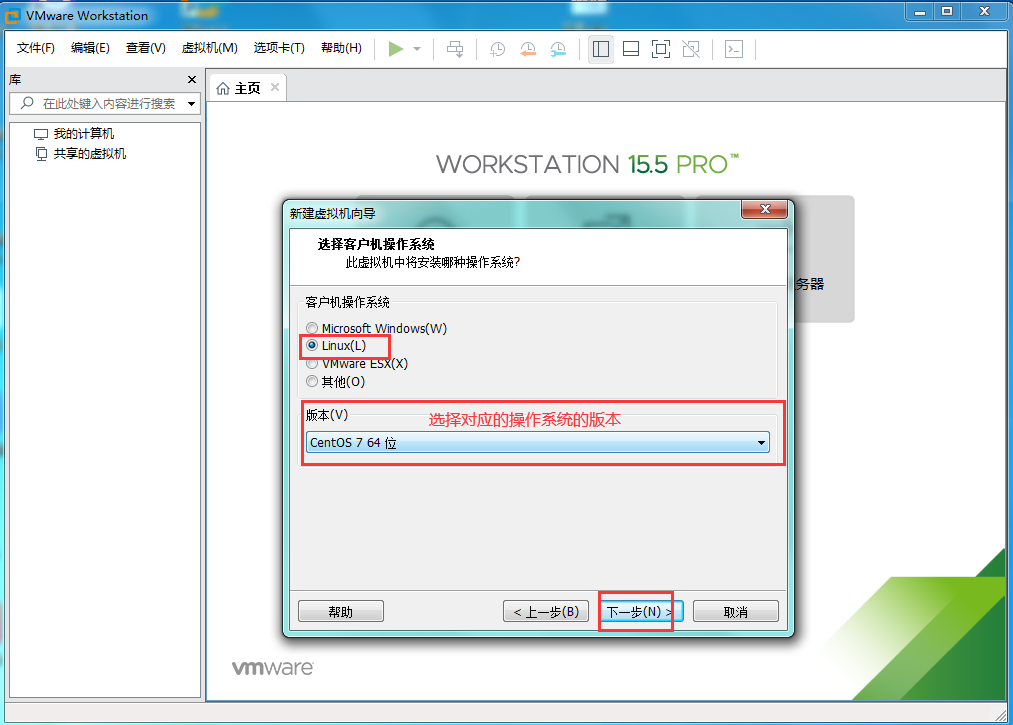
1. 选择虚拟机兼容性（保持默认）



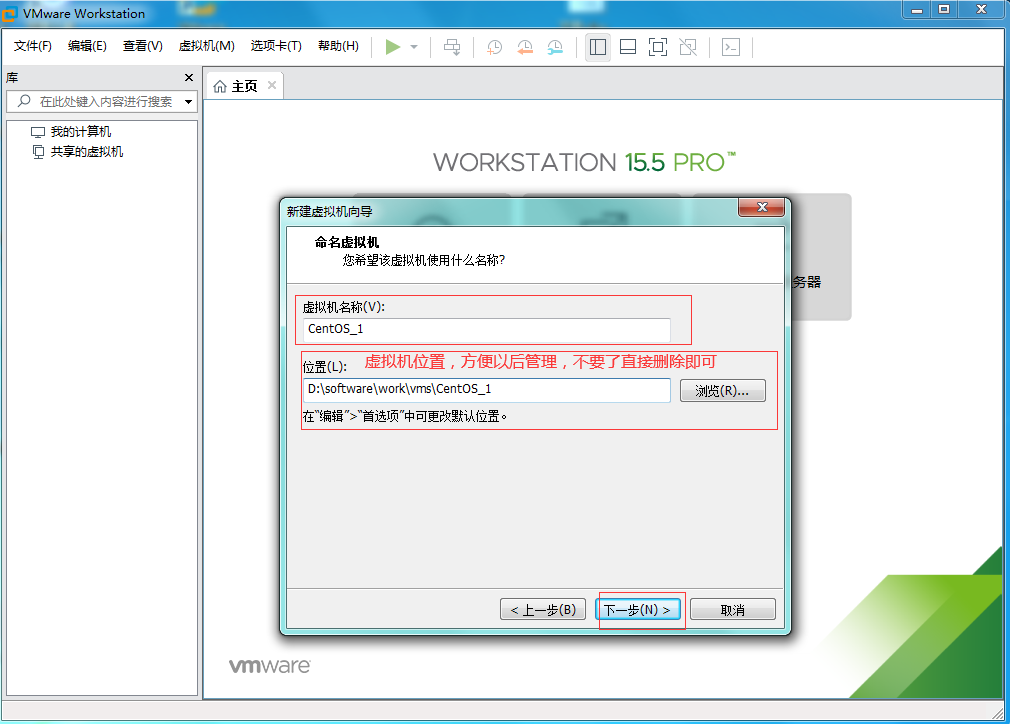
3选择安装来源



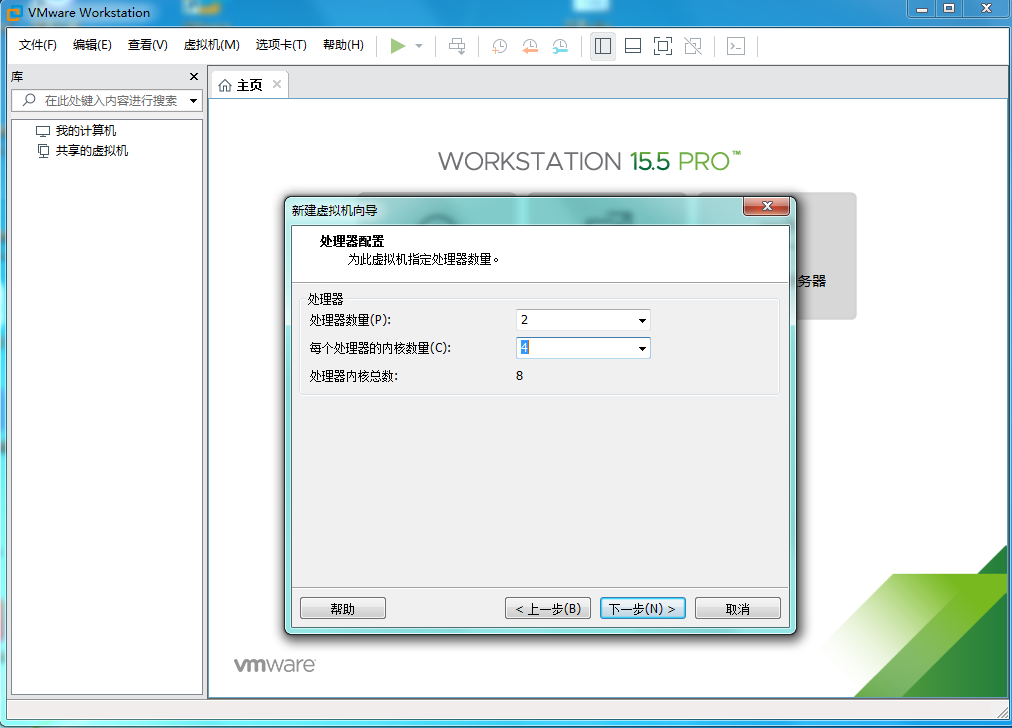
4.选择客户机操作系统



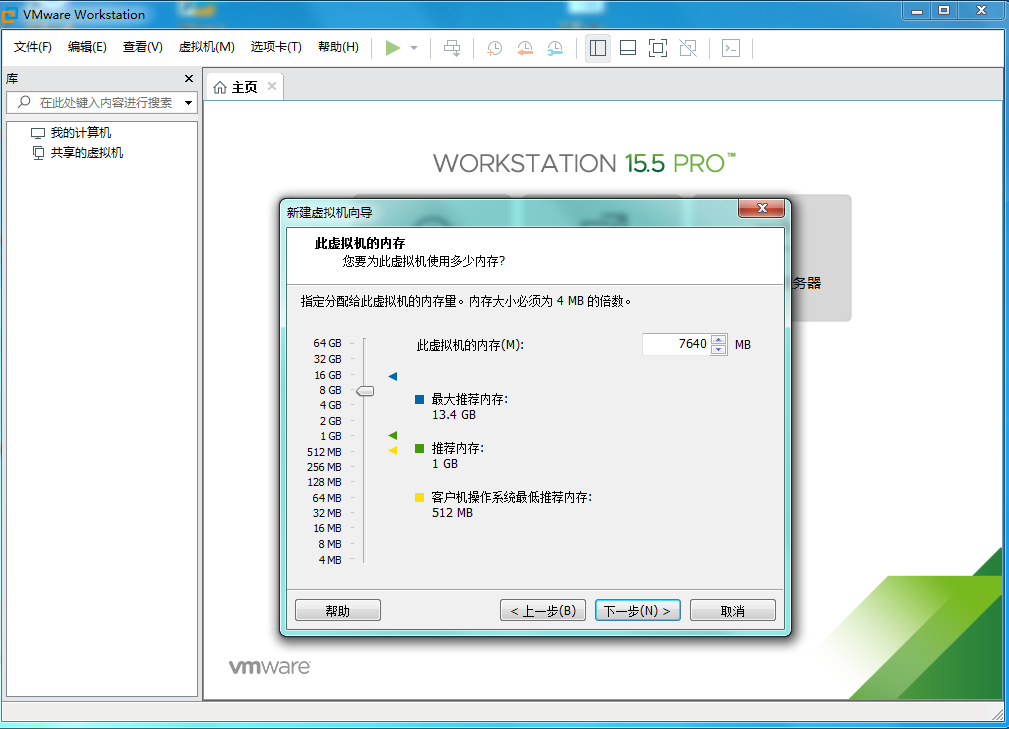
5命名虚拟机



6.处理器配置



7.设置虚拟机的内存



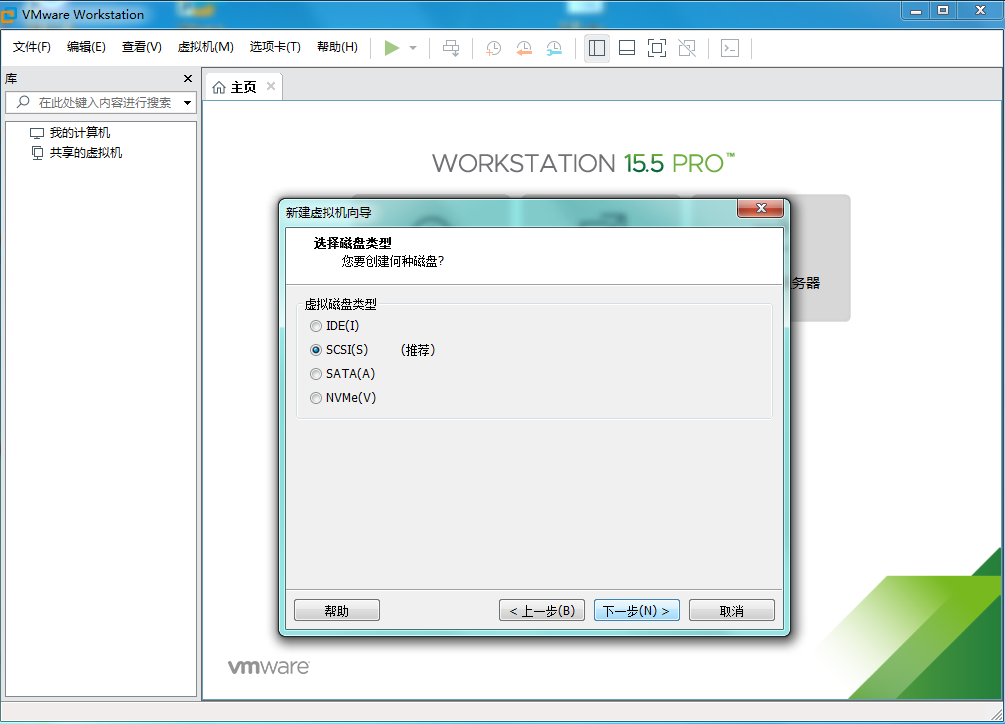
8.设置网络类型



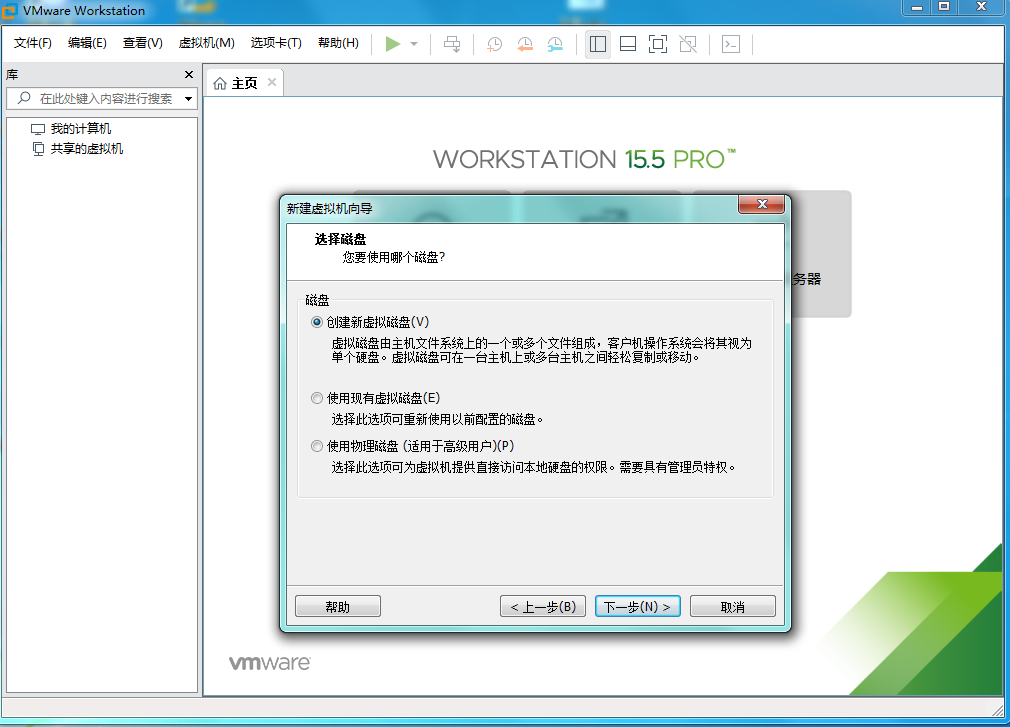
9.选择I/O控制类型（保持默认-下一步）



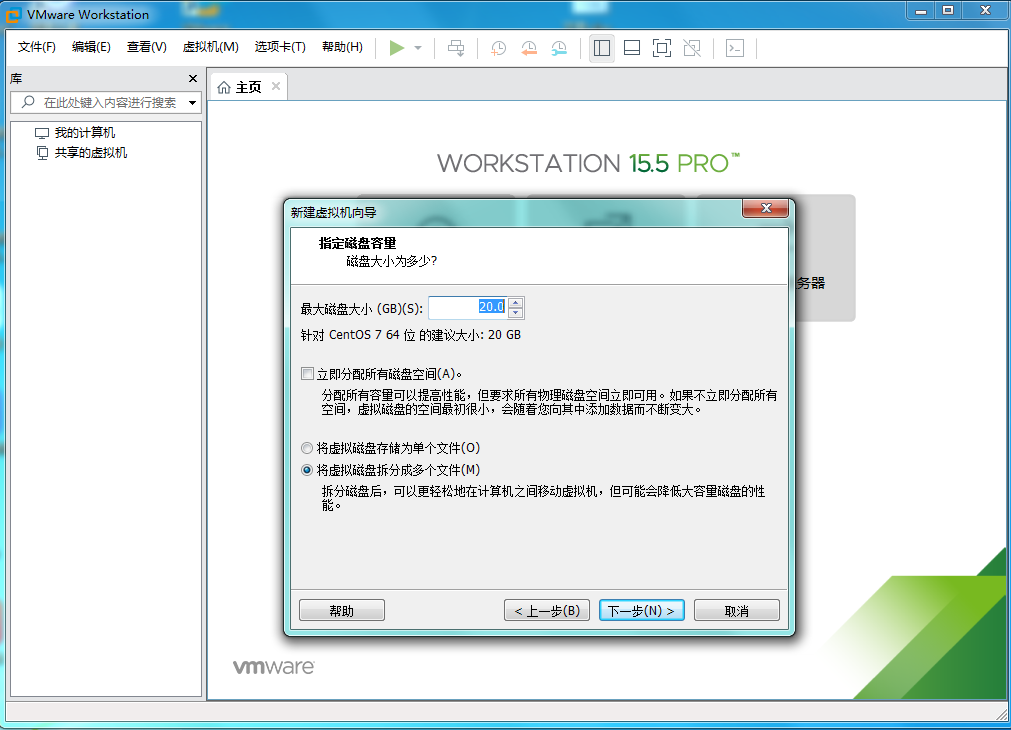
10.选择磁盘类型（保持默认-下一步）



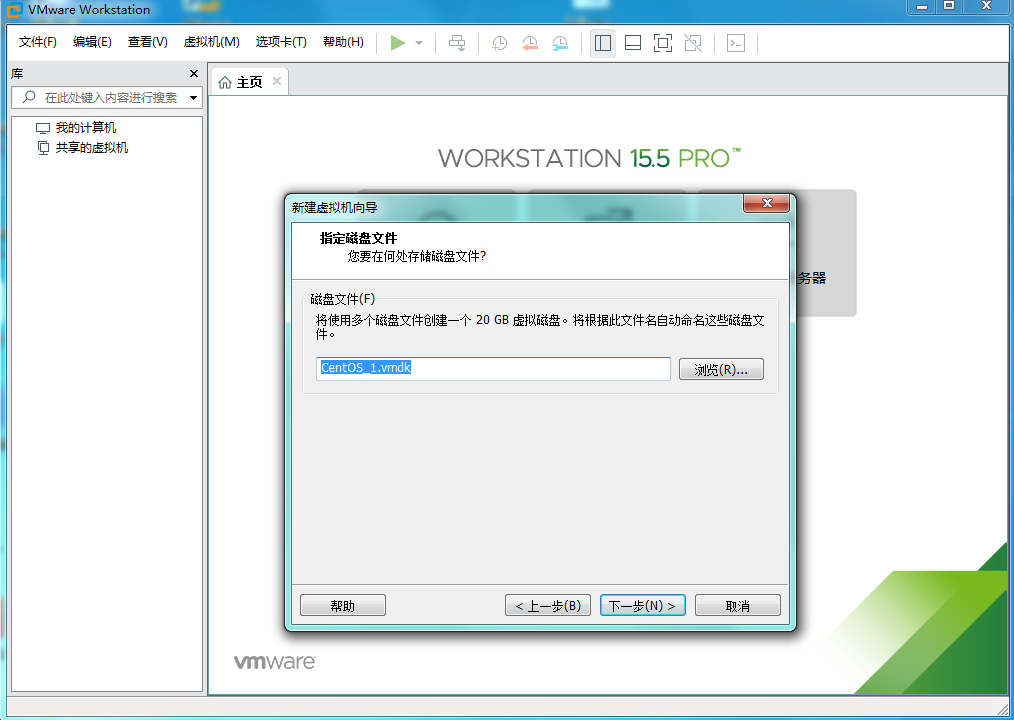
11.选择磁盘（保持默认-下一步）



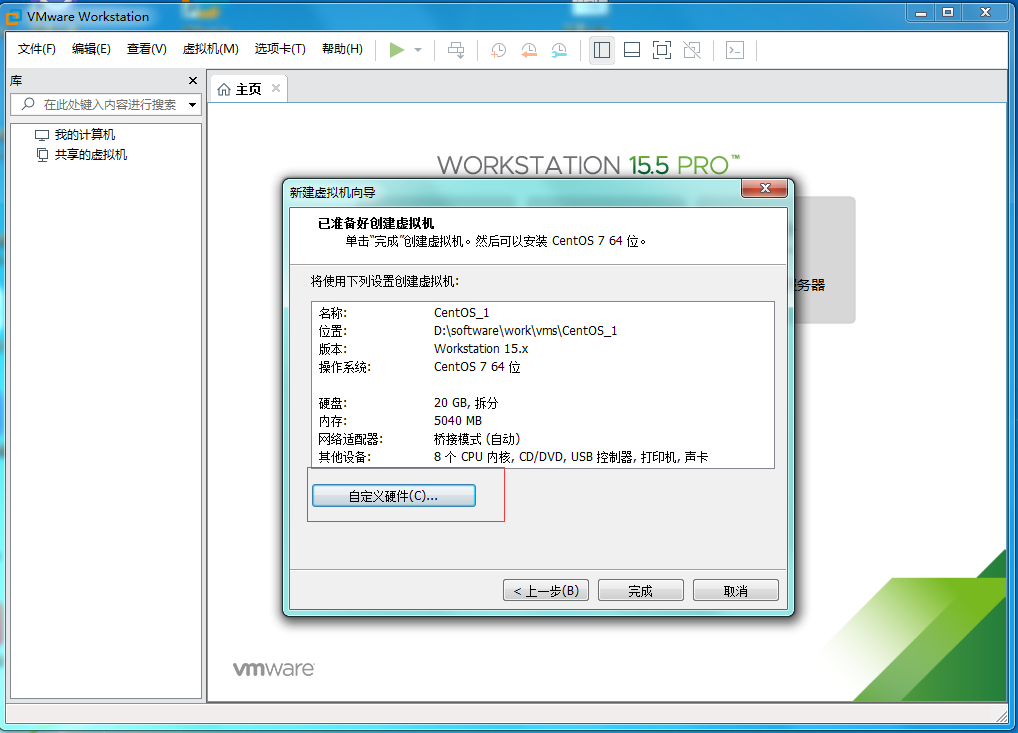
12.指定磁盘容量（保持默认）



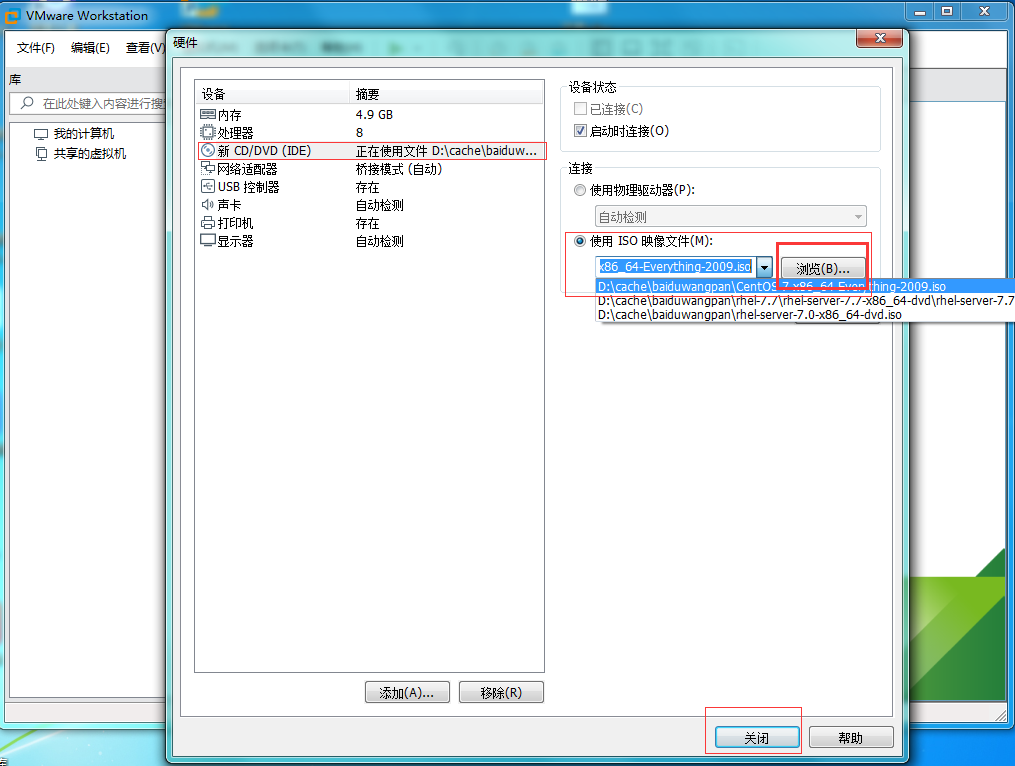
13.指定磁盘文件（保持默认）



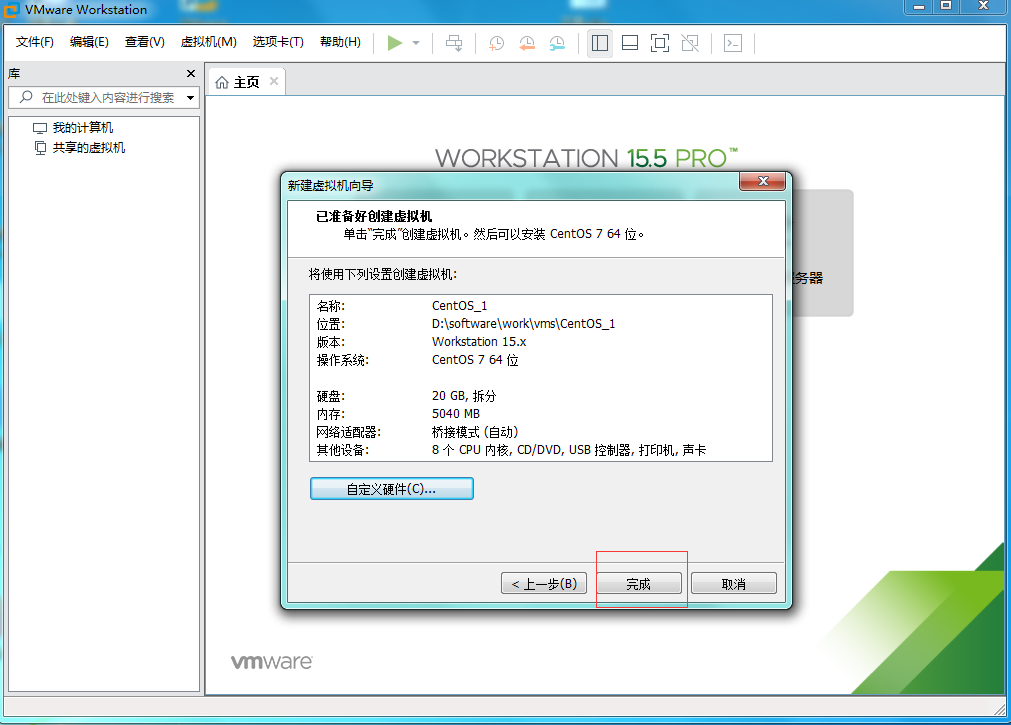
14.自定义硬件



15.引入对应操作系统镜像

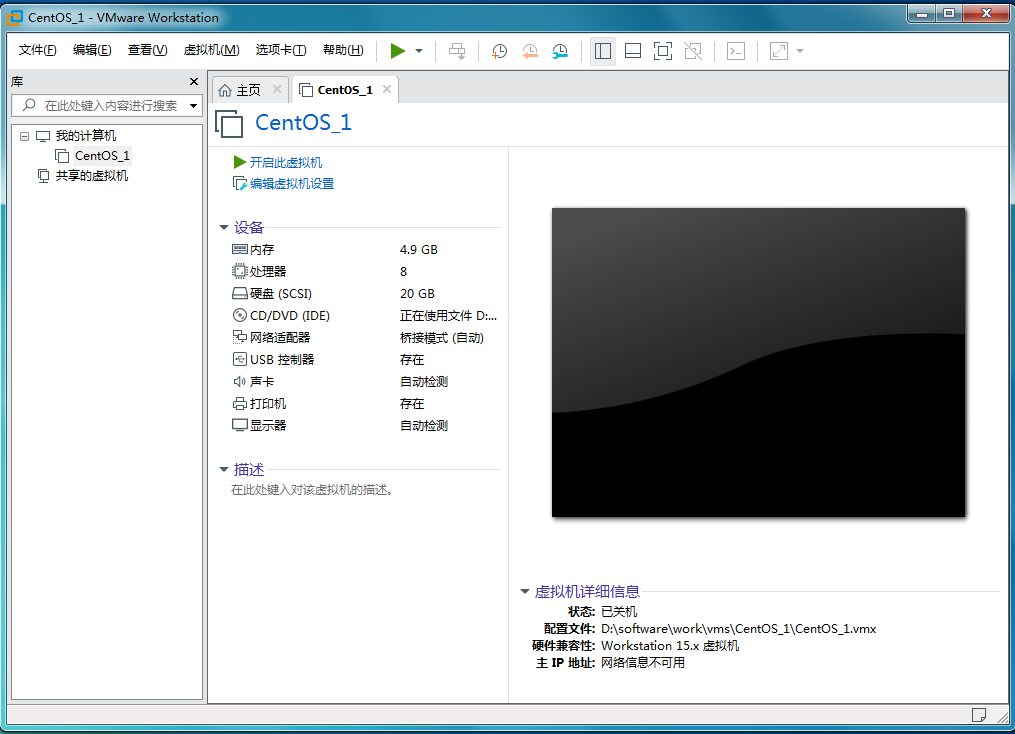


16.点击完成

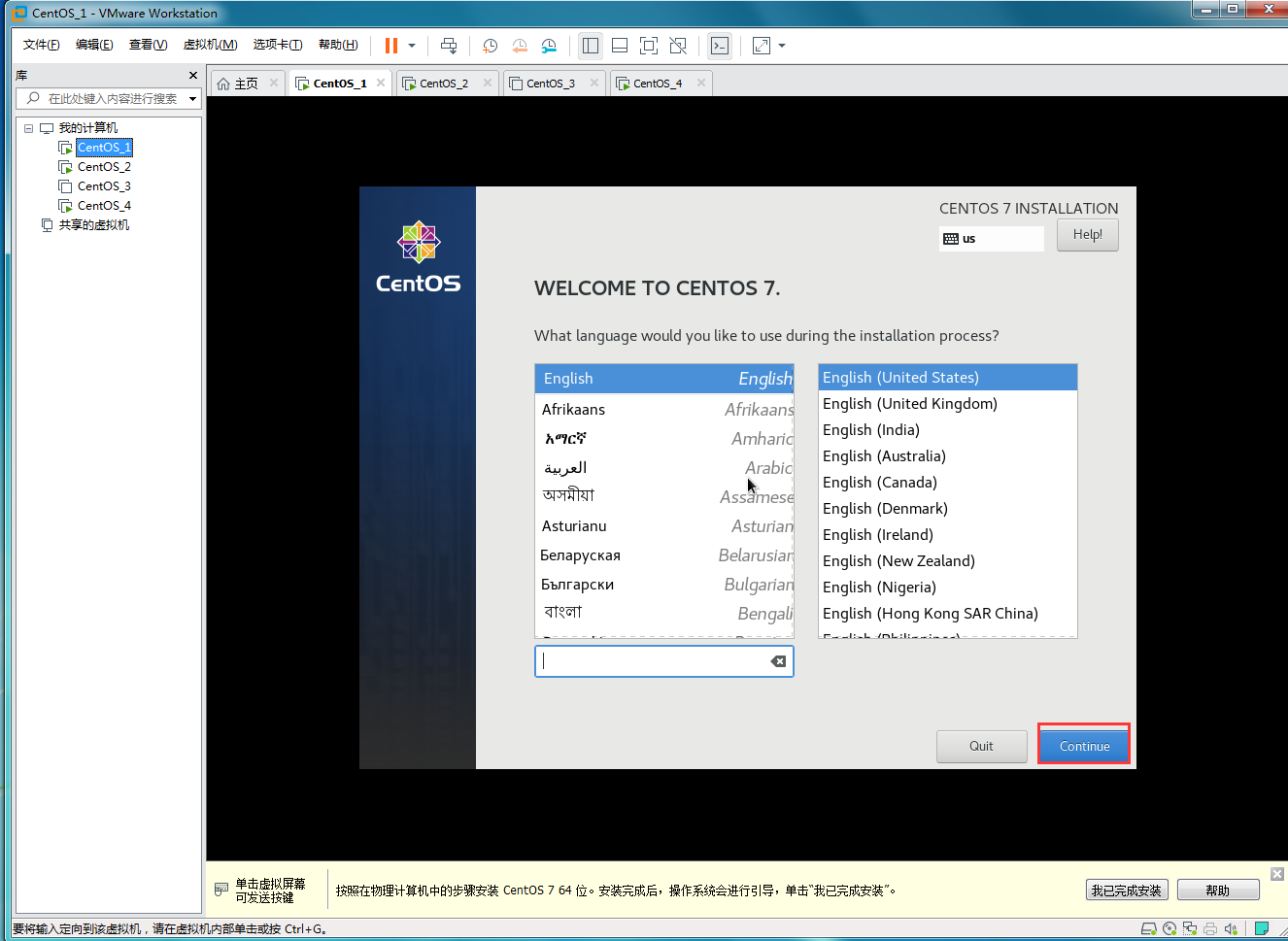


## 在VMware中安装对应操作系统（CentOS）

1. 开启创建的虚拟机

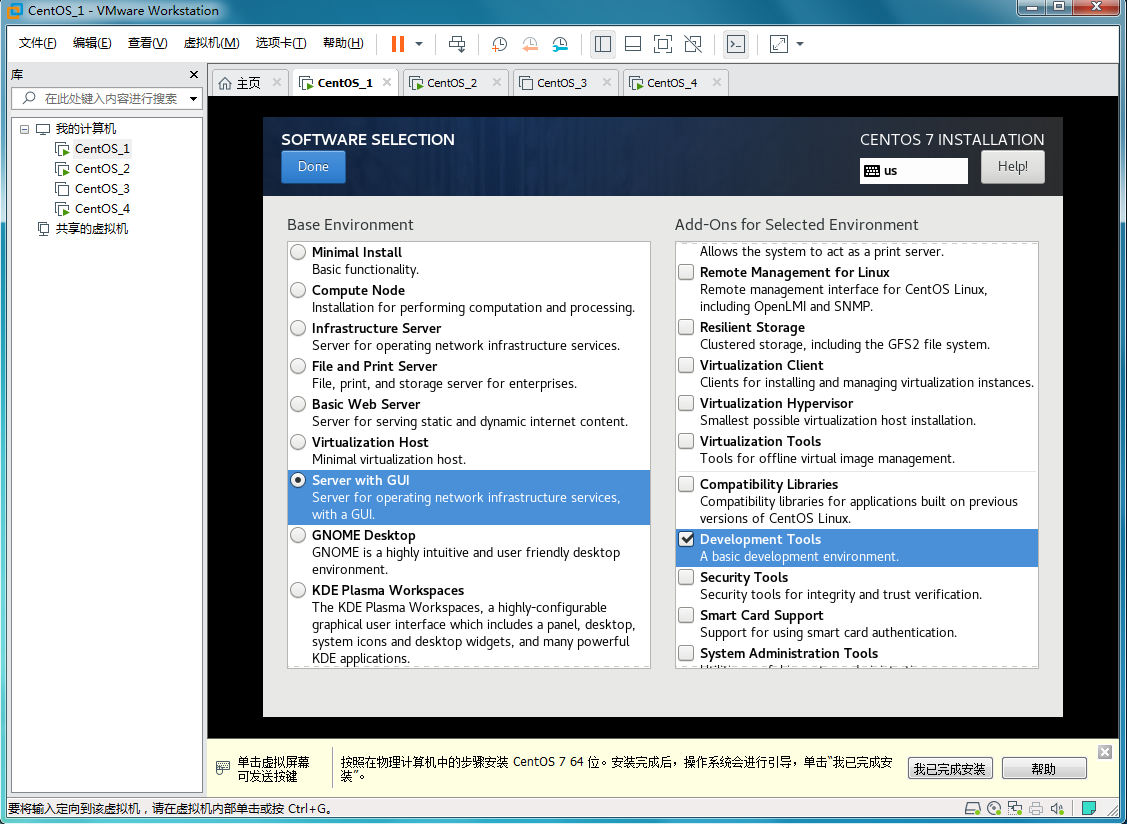


1. 选择语言

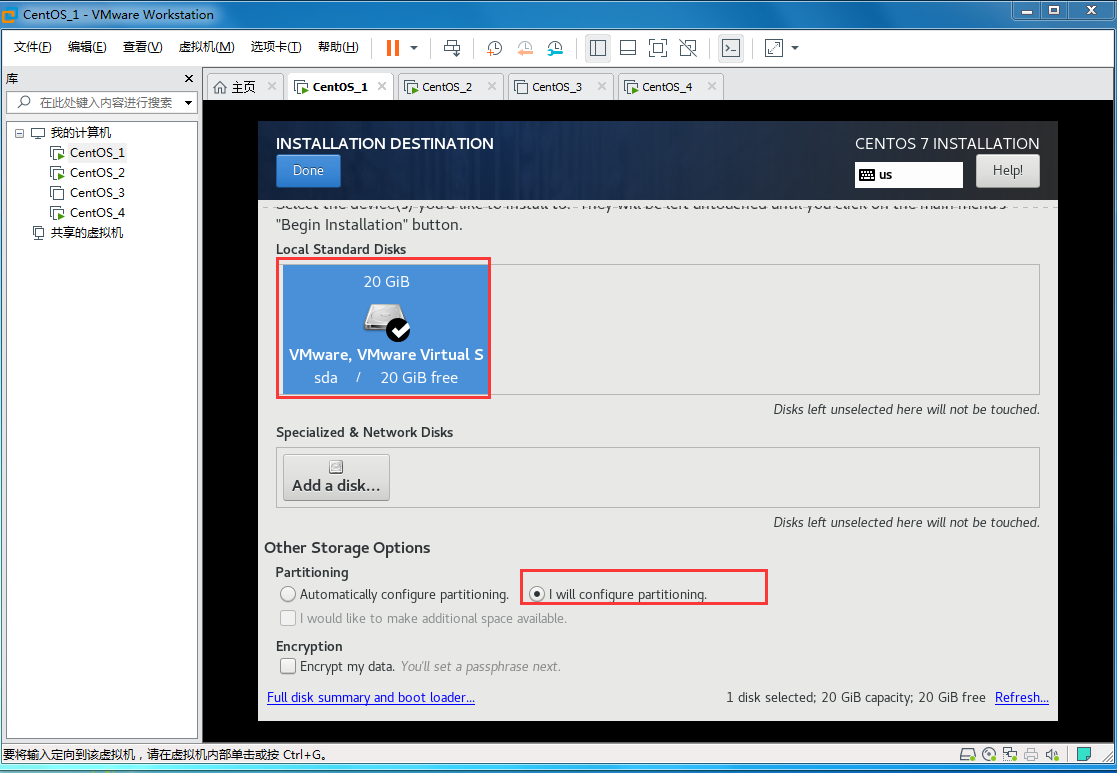




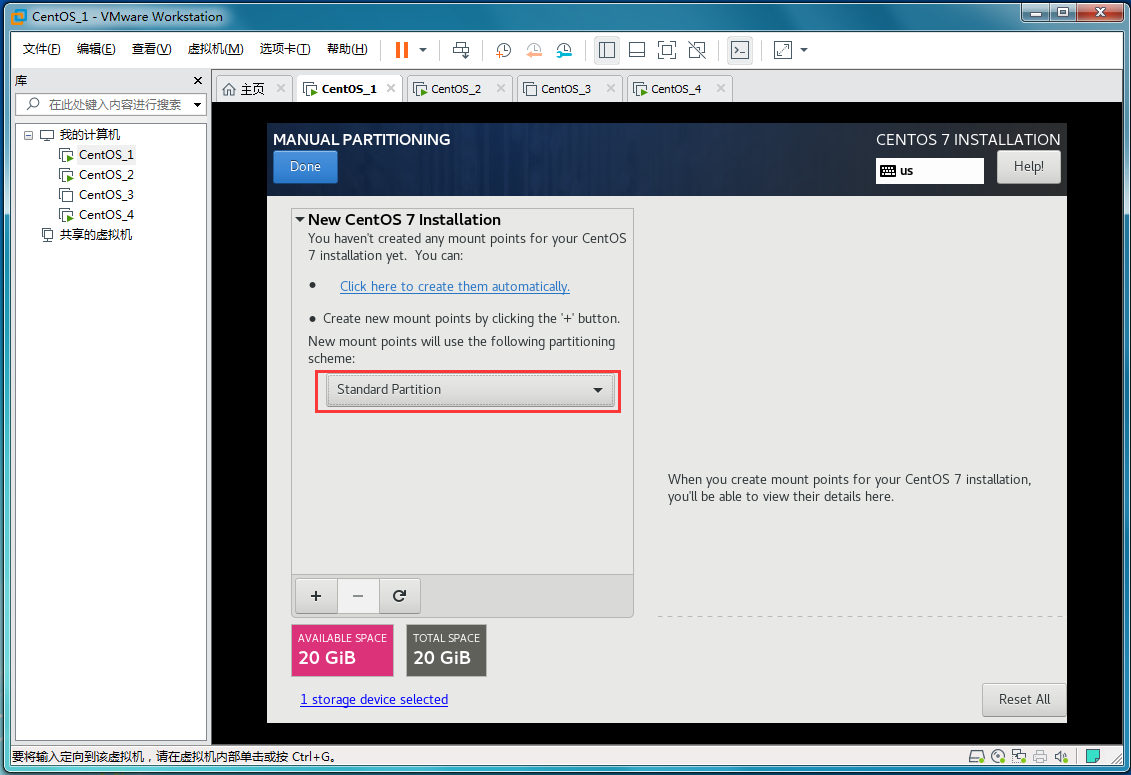
1. 软件选择（选择带GUI的服务器和基本的开发环境）



1. 分区（选择手动分区）



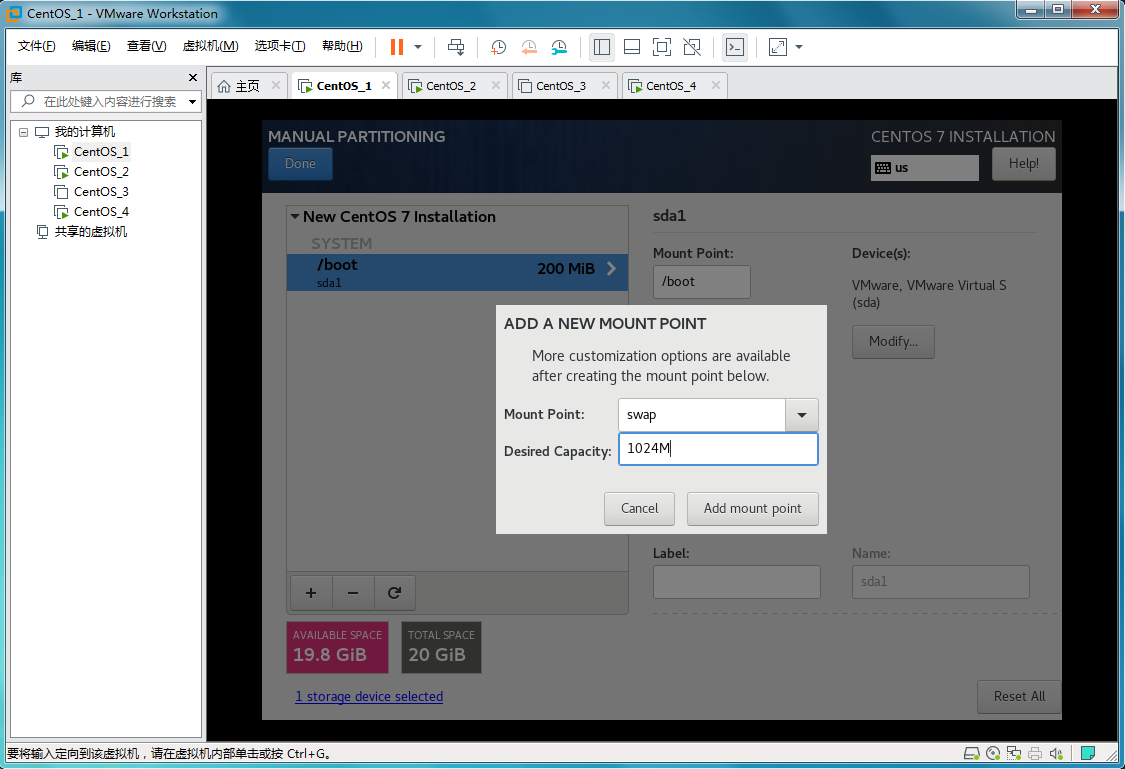
1. 手动分区（选择标准分区）



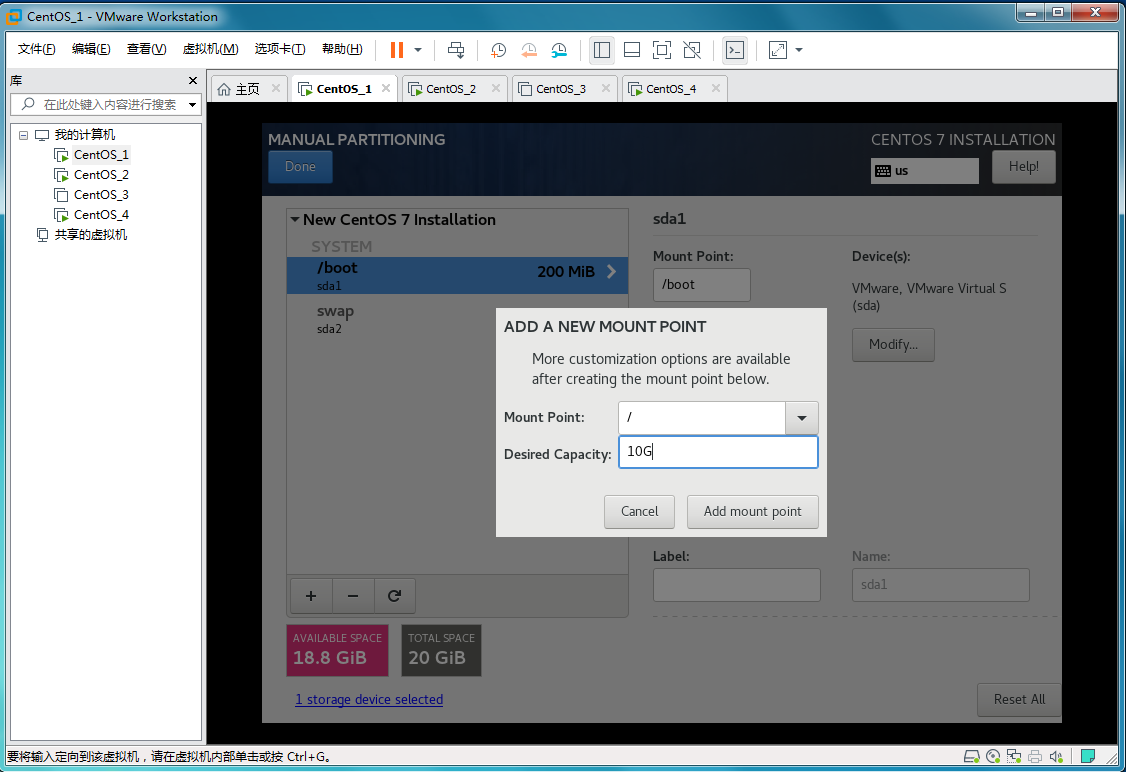
**引导分区**

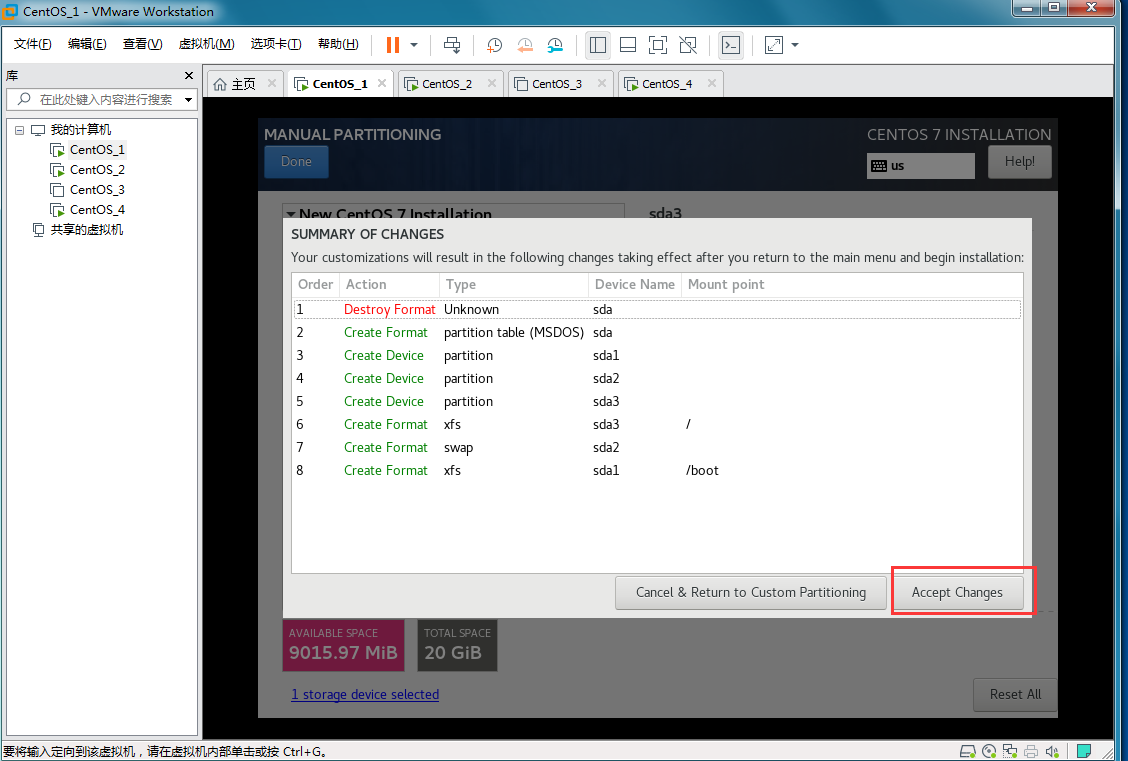


**一种说法是给内存的1.5倍或2倍，具体场景具体分析。**

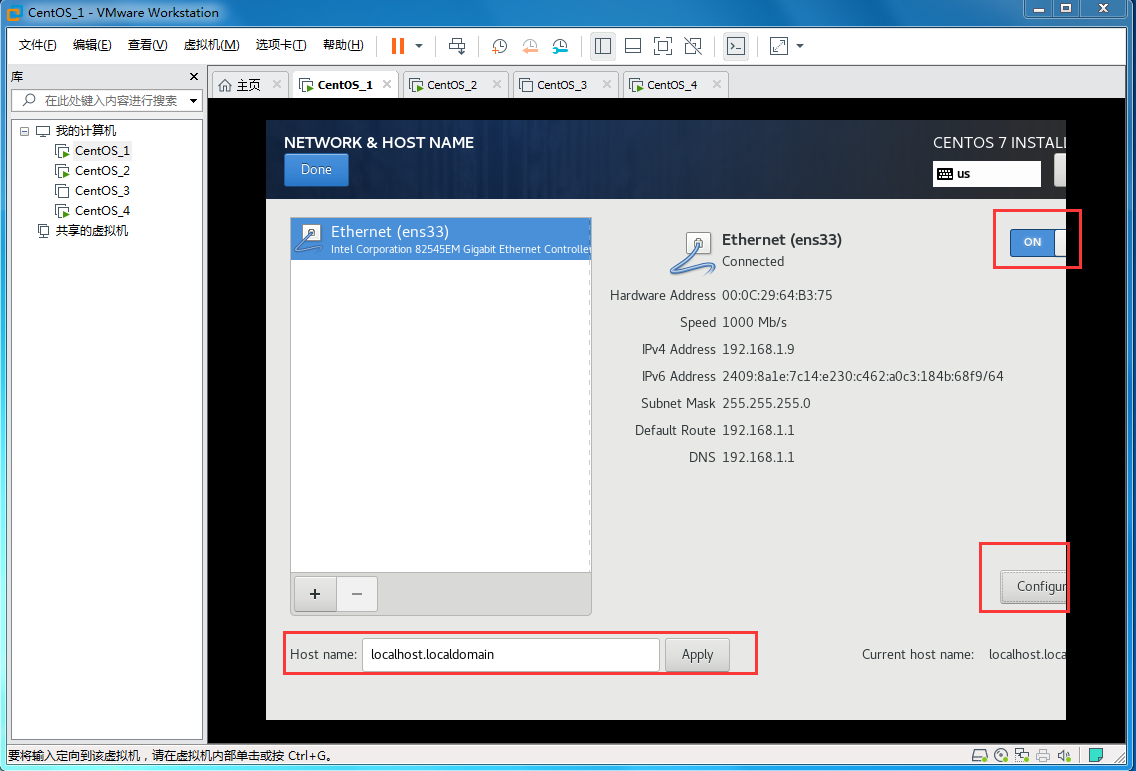


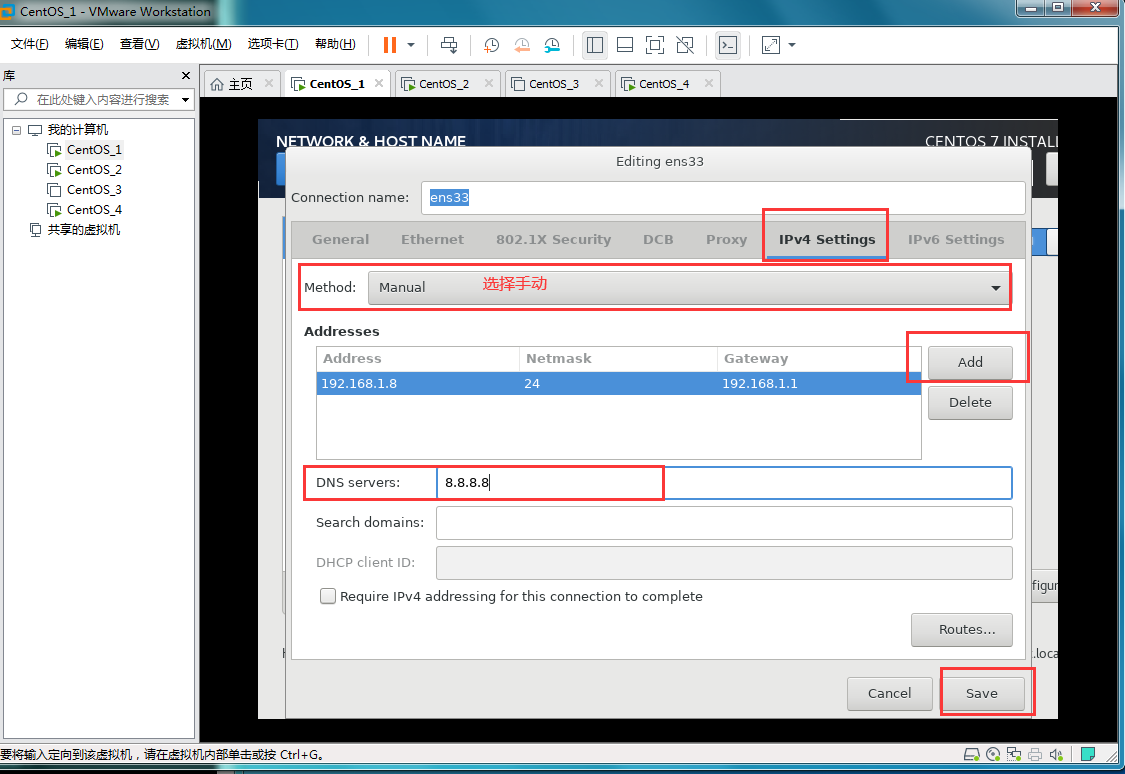
一般给10G或20G





1. 设置网络和主机

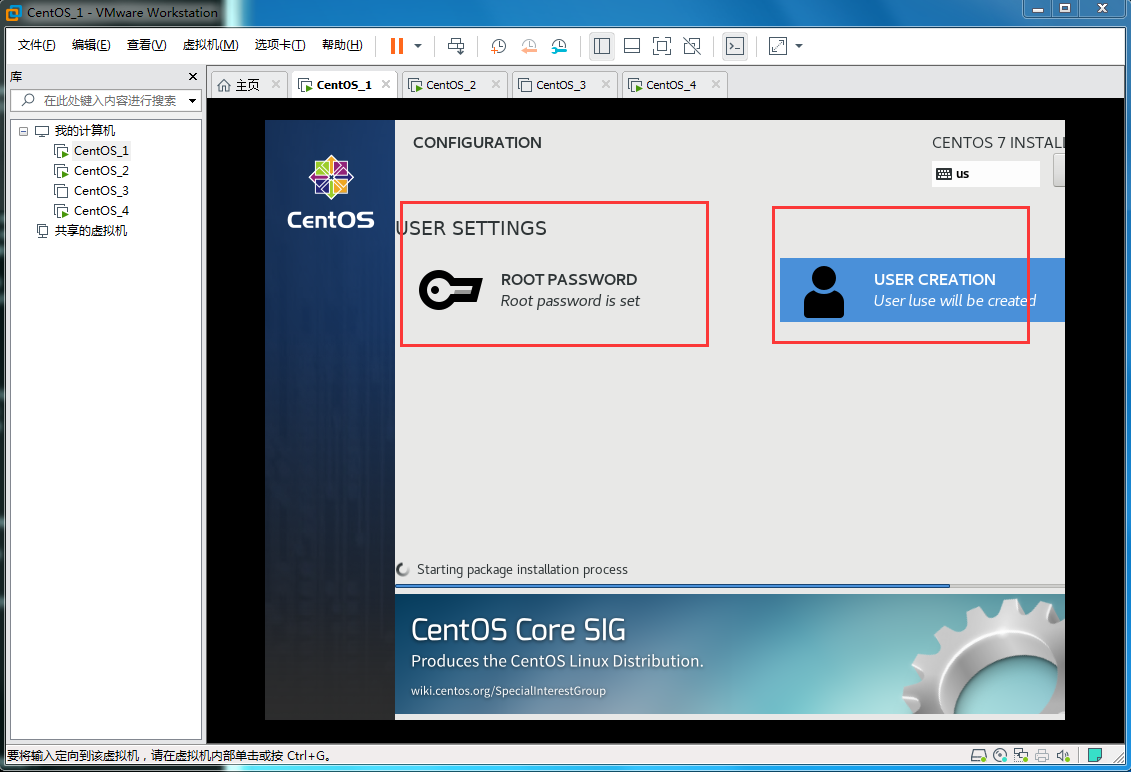




7.安装（点击开始安装）



1. 设置root用户密码并创建用户



## 安装操作系统注意事项

1. boot分区:引导分区，系统启动。
2. swap分区：内存拓展分区,给多大？1.5倍或2倍吗？



因此，一般来说可以按照如下规则设置swap大小：

* 4G以内的物理内存，SWAP 设置为内存的2倍，不超过4G。
* 4-8G的物理内存，SWAP 等于内存大小。
* 8-64G 的物理内存，SWAP 设置为8G。
* 64-256G物理内存，SWAP 设置为16G。

1. / 根分区：所有文件的根
2. sda ：第一块磁盘
3. RHEL7默认使用xfs文件系统，RHEL6默认使用ext4. RHEL5默认使用ext3.Win7，win10使用的是NTFS，XP使用的是Fat32.

## VMware虚拟机的使用

1. Vmware虚拟机增加虚拟机可以用物理内存必须在虚拟机关机的情况下操作。
2. Vmware虚拟机不能在开机状态下添加硬件设备。
3. Vmware虚拟机控制权限使用Ctrl+Alt返回正常模式(宿主机)
4. Vmware虚拟中使用Ctrl+L清屏
5. Vmware虚拟机中使用Ctrl+C 取消某一个操作。
6. Vmware虚拟机BIOS设置：开机按F2

手速慢技巧：虚拟机—>电源—>打开电源时进入固件—直接进入BIOS.

1. Vmware虚拟机安装VMware Tools解决屏幕分辨率问题（虚拟机—>安装VMware Tools）
2. Vmware虚拟机使用ISO挂载光盘（关机状态下操作）
3. Vmware快照管理(虚拟机—>快照)

# Linux创建基本的可以试验环境

## 1.配置固定的可以联网的IP地址

1.RHEL7中弱化了setup的功能,对于网络管理来说，network服务的功能弱化，主要通过NetworkManager对网络进行管理；

2.RHEL7通过nmtui修改网络配置（RHEL7默认安装,前提时需要开启NetworkManger.service才可以使用）;

3.首先确认NetworkManager.service是否正常启动： systemctl status NetworkManager如果没有启动,手动启动。

1.首先确认NetworkManager.service是否正常启动

systemctl status NetworkManager

2.如果NetworkManager.service没有启动,手动启动

systemctl start NetworkManager

**tip: 如果前缀唯一，按一下tab可以补齐命令。如果前缀不唯一按两下tab，列出所有以前缀开头的命令；**

1. 通过修改配置文件的方式修改IP

vim /etc/sysconfig/network-scripts/ifcfg-\*\*\*\*

#ONBOOT必须设置为yes

ONBOOT=yes

IPADDR="局域网的IP地址"

1. 使用ifconfig命令查看IP

## 2.关闭防火墙并设置开机自动关闭

1.查看防火墙是否开启

systemctl status firewalld.service

2.关闭防火墙

systemctl stop firewalld

3.设置开机自动关闭：

systemctl disable firewalld

## 3.关闭selinux并设置开机自动关闭

1.临时关闭selinux

setenforce 0

2.设置开机自动关闭：（重启后生效(reboot)）

vim /etc/sysconfig/selinux

SELINUX=disable

**tips:**  
1.)i(insert):插入模式  
2.)如果操作失误，可以按Esc 然后 u 撤销;  
3.)粘贴:可以选择要粘贴的内容按下滚轮（要在插入模式下）  
4.)Esc：退出插入模式  
5.)：wq 退出保存  
6.)重启命令:reboot

## 4.设置光盘开机自动挂载

1.设置光盘自动挂载，为了配置安装软件的yum源。

2.在Linux一切皆文件，光驱(光盘)也是一个文件。硬件是不能直接访问的需要挂载才可以访问。

### 基本步骤

1.打开对应的文件

vim /etc/fstab

2.向文件中加入如下代码，将/dev/cdrom(光驱)挂载到一个空目录（/mnt）中，访问/mnt目录就可以访问到/dev/cdrom中的内容。可以将光驱挂载到其他空目录中，只是习惯性挂载到/mnt中。

/dev/cdrom /mnt iso9660 defaults 0 0

**tips:**  
1.)/dev/cdrom :光驱  
2.)/mnt：目录  
3.)0：开机是否自检  
4.)0：自检的顺序;  
5.)把光驱挂载到/mnt：目录  
6.)iso9660:光驱的格式

3.如果写错了系统就起不来了,所以需要先验证是否配置正确，可以使用mount -a 命令验证是否配置正确；

3.1执行mount -a 命令，mount -a 会去读/etc/fstab 去挂载这里面的文件;

mount -a

3.2查看文件是否挂载到了挂载目录

ls /mnt

挂载相关命令

**df：查看当前的挂载点**

**umount /mnt:卸载对应的挂载点**

## 5.配置好本地yum源

**1.yum的配置信息都储存在一个叫yum.repos.d的配置目录中,通常位于/etc目录下，这是整个yum系统中的重中之重;**

**2.yum源是安装软件包的一个工具；**

### 配置yum源

**1.打开yum.repos.d的配置目录**

**cd /etc/yum.repos.d/**

1. **编写yum配置文件**

**vim ./rhel7-local.repo**

**[rhel7-source] #名称；如果有多个yum源，yum源的名称必须保持唯一；**

**name=rhel-server #描述信息 可以随意编写;**

**baseurl=file:///mnt #(系统盘)文件路径 (可以由多种方式)**

**enabled=1 #启用**

**gpgcheck=0 #取消验证**

**3.测试yum源是否配置成功 如果清空yum缓存, 然后生成缓存列表，可以生成则证明yum配置文件正确;**

**yum clean all 清空yum缓存**

**yum list 生成缓存列表**

# 桥接模式下不能ping通外网解决方案

root@localhost ~]# ping baidu.com

ping: baidu.com: Name or service not known

[root@localhost ~]#

1.查看宿主机中的网关：win+r-->cmd-->ipconfig

无线局域网适配器 WLAN:

连接特定的 DNS 后缀 . . . . . . . :

IPv6 地址 . . . . . . . . . . . . : 2409:8a1e:7c1f:3a10:cd12:97fc:ddf2:2d23

临时 IPv6 地址. . . . . . . . . . : 2409:8a1e:7c1f:3a10:6061:ec77:d72c:3a2a

本地链接 IPv6 地址. . . . . . . . : fe80::cd12:97fc:ddf2:2d23%8

IPv4 地址 . . . . . . . . . . . . : 192.168.1.4

子网掩码 . . . . . . . . . . . . : 255.255.255.0

默认网关. . . . . . . . . . . . . : fe80::14bf:87ff:fed0:75cc%8

192.168.1.1

2.查看虚拟机中配置的网关（GATEWAY）是否和宿主机一致

PROXY\_METHOD=none

BROWSER\_ONLY=no

BOOTPROTO=none

DEFROUTE=yes

IPV4\_FAILURE\_FATAL=no

IPV6INIT=yes

IPV6\_AUTOCONF=yes

IPV6\_DEFROUTE=yes

IPV6\_FAILURE\_FATAL=no

IPV6\_ADDR\_GEN\_MODE=stable-privacy

NAME=ens33

UUID=d62a3a73-0d44-4df9-9afa-1ccbe6cbc9ef

DEVICE=ens33

ONBOOT=yes

IPADDR=192.168.1.18

PREFIX=24

GATEWAY=192.168.1.1

DNS1=8.8.8.8

IPV6\_PRIVACY=no

3.重启网络服务

systemctl restart NetworkManager

4.修改虚拟机的虚拟网络 在虚拟机编辑菜单-虚拟网络编辑器-重置后然后修改配置

# Linux基本命令操作

## 1.单词整理

|  |  |
| --- | --- |
| **单词** | **说明** |
| Terminal | 终端 |
| network-scripts | 网络脚本 |
| passwd | 密码文件 |
| nologin | 禁止登录 |
| shutdown | 关机 |
| reboot | 重启 |
| poweroff | 关机 |
| Grep | 过滤 |
| localhost | 本机 |
| useradd | 添加用户 |
| graphical | 图形[ˈgræfɪkl] |

## 2.Linux的终端介绍（terminal）

### 1.terminal（终端）终端的分类

terminal（终端）分为tty(控制台终端)和pst（虚拟终端）

**1.tty ：控制台终端**  
在Linux系统中，整个图形界面是一个tty，整个终端有六个（tty1-tty6），tty1 就是图形界面,tty2-6 是字符界面。

图形界面切换到字符界面终端：ctrl+alt+F2-6；

字符界面切换到另一个字符界面终端：alt+F2-6；

字符界面切换到图形界面：alt+F1；

**2.pts虚拟终端**  
1.在centos系统中的图像界面中打开的terminal就是一个pts虚拟终端，但是整个图形界面是一个tty。

2.通过ssh相关工具连接上后,显示出来的终端为pts。

### 2.pts虚拟终端常用快捷键

|  |  |
| --- | --- |
| **快捷键** | **说明** |
| ctrl+shift+"+" | 放大字体 |
| ctrl+"-" | 缩小字体 |
| ctrl+shift+t | 新建虚拟终端 |
| alt +数字键 | 多个终端之间切换 |
| alt+F4 | 关闭终端 |

### 3.pts虚拟终端常用命令

|  |  |
| --- | --- |
| **命令** | **说明** |
| exit | 关闭当前终端 |
| whoami | 查看当前登录的用户 |
| who am i | 查看当前登录的终端对应的pts号 |
| w | 查看pst（pst是一个进程） |
| ssh | 远程连接linux 服务器。 shh root@192.168.1.44 |

## 3.shell的提示符

1.[用户@主机名 当前目录名称]# 其中#：代表root ;$:普通用户;

2.当前目录为~表示当前用户的家目录；

[root@wangtao /]#|$

## 4. Base shell 的基本语法

1.命令[选项]([参数]) [选项的值]([参数的值])

2.空格作为分割;

3.常用选项（参数）：-h ---help

特点：选项为： -字母或--加单词

## 5.Linux中不同颜色代表不同的文件类型

|  |  |  |
| --- | --- | --- |
| **颜色** | **说明** | **实例** |
| 蓝色 | 目录 | /etc |
| 黑色 | 文件 | /etc/passwd |
| 浅蓝色 | 链接 | /etc/grub2.cfg |
| 红色 | 压缩包 | boot.tar.gz |
| 绿色 | 可执行文件 | /etc/init.d/network |
| 黑底黄字 | 设备文件 | /dev/sda |

## 6.基本命令

### 1.ls

语法： ls 目录

说明：查看指定目录下的文件，如果不加目录查询当前目录。

参数：-l 显示详细信息，系统快捷方式ll= ls -l

-a 显示隐藏文件

-d 查看目录（不查看里面的内容）

-h 方便阅读文件的大小

### 2.pwd

语法：pwd

说明：查看当前目录位置，没有参数。

### 3.cd

语法：cd 目录名

说明：切换目录

参数: . 当前目录

..父目录（上层目录、上级目录）

实例：

cd: 回到用户自己的主目录；

cd~：回到用户自己的主目录；

cd -：回到上一个操作的目录；

cd /etc///sysconfig/和cd /etc/sysconfig一样

cd /etc/和 cd /etc一样

## 7.查看系统时间和BIOS硬件时间

### 1.查看系统时间date

[root@wangtao etc]# date

Sun Aug 28 05:20:48 EDT 2022

### 2.修改系统时间:date -s  时间

date -s 2022-8-28

### 3.格式化显示的时间:date '+%y- %m- %d %H: %M'

[root@wangtao etc]# date '+%y- %m- %d %H: %M'

22- 08- 28 05: 28

### 3.查看BIOS时间:hwclock

[root@wangtao etc]# hwclock

Sun 28 Aug 2022 05:26:14 AM EDT -0.772381 seconds

**tips:** 在RHEL6以前,系统时间和硬盘时间时不一样的，差8小时，因为操作系统以BIOS硬件为标准时间;我们是在东八区，所以相差8小时。

## 7.参数 -h 、--help和man命令

查看命令相关帮助信息

date --help

fdisk -h

man find

## 8.关机命令

关机命令有很多，如shutdown， init， reboot， poweroff。

### 1.shutdown

语法：shutdown [选项]

作用：关机,重启，定时关机;

参数：

-r 重启

-h 关机

-h +时间（+1） 定时关机 1分钟后关机

-c 取消

实例

shutdown -r now #立即重启

shutdown -r +10 #10分钟后重启

shutdown -r 22:22 #22:22以后重启;

shutdown -h now #立即关机

### 2.init

语法：init 0-6

作用：切换系统运行级别

说明：

1.在rhel6下使用时init 0-6；

2.RHEL7 不再使用/etc/inittab文件进行默认的启动级别配置。systemd使用比sysvinit的运行级更为自由的target替代。 第3运行级别用 multi-user.target替代;第5运行级别用graphical.target替代;其中runlevel3.target和runlevel5.target分别是指向multi-user.target和graphical.target。

实例

init 0 关机

#### 2.1.Linux的7个启动级别：

1.)0:**系统停机模式，**系统默认运行级别不能设为0，否则不能正常启动,机器关闭;

2.)1:**单用户模式,**root权限，用于系统维护,禁止远程登录,就像windows下的安全模式登录；

3.)2:**多用户模式,** 没有NFS网络支持。

4.)3:**完整的多用户文本模式，**有NFS网络的支持，登录后进入控制命令模式。

5)4:**系统未使用，保留一般不用，**在一些特殊情况下可以用它来做一些事情。如在笔记本的电.池用尽时，可以切换到这个模式来做一些设置;

6.)5:**图形化模式，**登录后进入图形GUI模式，Windows 系统；

7.)6:**重启模式，**默认允许级别不能设置为6，否则不能正常启动,运行init6 机器会重启;

#### 2.2.rhel7下切换系统运行级别

**rhel7下切换系统运行级别**

1.切换到第三运行级别（字符界面）

systemctl isolate multi-user.target

或

systemctl isolate runlevel3.target

2.切换到第五级运行级别

systemctl isolate graphical.target

或

systemctl isolate runlevel5.target

#### 2.3.rhel7设置默认的运行级别

**rhel7下设置默认的运行级别**

1.设置默认为第三启动级别

systemctl set-default multi-user.target

2.设置默认为第五启动级别

systemctl set-default graphical.target

3.查看默认Linux启动级别

systemctl get-default

4.查看运行级别

runlevel

### 3.定时开机

通过BIOS完成定时开机.

1.进入BIOS,一般是在开机后出现主板画面时按delete键,部分按F2，F1;

2.选择Power Management Setup,进入电源管理设置;

3.通过回车进入这个设置后，选择Wake Up Event Setup,回车选择 Press Enter;

4.继续回车，将Resume By RTC Alarm选项的Disabled更改为Enabled,然后继续回车确定.然后再继续设置时间和日。

5.F10 保存退出;

# Linux文件的基本管理

## Linux目录结构

### 1.linux的系统目录结构

LInux当中一切都是文件;linux中没有windows中的本地磁盘(c,d,e)，都是在根目录下；

**/ /home /root /dev /usr /etc /boot /lib /var /tmp /proc /bin /sbin**

|  |  |
| --- | --- |
| **目录** | **说明** |
| / | 根分区是所有的文件和目录的起始点。只有root用户对此目录拥有写权限; |
| /etc | 配置文件,包含所有应用程序的配置文件,也包含启动、关闭某个特定程序的脚本; |
| /boot | 存放Linux系统启动时需要加载的文件.kernel、grub等文件都存放在boot目录下; |
| /var | 是一个可以增长的目录,包含很多经常变的文件;如 /var/log（系统日志）/var/lib (包文件) |
| /root | root用户的家目录，管理员所有数据。 |
| /tmp | 临时文件存储位置 不能存放重要的数据,系统会定时清空tmp下的文件; |
| /usr | usr表示unix software source /usr/src 源代码目录; |
| /bin | 命令,此目录包含了二进制可执行文件 |
| /sbin | 系统命令,此目录中的命令主要供系统管理员使用,以进行系统维护;reboot、fdisk等; |
| /mnt | 挂载目录，挂载点,系统管理员可用于临时挂载文件系统. |
| /dev | 包含文件设备,在Linux 中，一切都被看做文件,终端设备、磁盘等都被看做文件; |
| /home | 普通用户所有数据存放在这个目录下; |
| /proc | 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; |
| /lib | 存放系统的库文件;lib 包含动态运行库和静态库;  lib\*\*\*.a 静态库  lib\*\*\*.so动态库  静态库在编译时被加载到二进制文件中；  动态库在运行时加载到进程的内存空间中;  简单的说,这些库是为了让程序能够正常编译运行;其作用类似于weindows里的.dll文件.几乎所有的应用程序都必须要用到这些共享库; |

### 2.Linux目录操作注意事项

1.本地管理员大多数情况下将额外的软件安装在/usr/local 目录下，且符号连接在/usr/local/bin下的主执行程序；

2. 系统的所有设置在/etc 目录下;

3. 不要修改根目录（“/”)或/usr目录下的任何内容,目录最好和LINUX发布时保持一致;

4.大多数工具和应用程序安装在目录：/bin,/usr/sbin,/sbin;

5.所有的文件在单一的目录树下,没有所谓的“驱动符”;

### 3.相对路径和绝对路径

1.路径:使用计算机找到需要的文件必须知道文件的位置,表示文件位置的方式就是路径;

2.绝对路径:在Linux中,绝对路径时从"/"开始的,比如 /usr,/usr/passwd;如果一个路径是从/开始的，它一定是绝对路径;

3.相对路径：以.或..开始的,.表示用户当前操作所处位置,..表示上级目录;

要把.和..当作目录来看；

## 文件管理相关命令

### 1.touch -创建文件

touch：

作用：常用来创建文件

语法：touch 文件名

### 2.mkdir -创建目录

mkdir

作用：创建目录

语法：mkdir 目录名

参数:-p：在创建时连同父目录一起创建;

说明:linux中不能创建再同一目录下创建一个目录和文件相同的两个文件;

### 3. cat -查看文件

cat

作用：查看文件内容

语法：cat 文件名

### 4.more -查看文件内容-往下翻

more

语法：more 文件名

作用：查看文件内容

特点：按下回车刷新一行，按下空格刷新一屏;

          q 退出;

缺点：只能往后翻;

### 5.less -查看文件内容-上下翻页

less

 语法：less 文件名

 作用：查看文件内容

 特点:  使用光标键可以上下翻页;

           q 退出

more 和less的区别

1.more 只能往后翻，但几乎不需要加参数,空格键向下翻页，enter键时向下翻一行，在不需要后退的情况下比较方便;

2.less支持前后翻滚，既可以向上翻页（pageup键）,也可以向下翻页（pagedown键），空格键时向下翻页，enter键时向下翻一行 ;

### 6.head-查看文件内容-前N行

head

语法：head 文件名；

作用：查看文件内容；

 说明：-n 数字 显示多少行，默认前10行；

### 7.tail-查看文件内容-后N行

tail

语法：tail 文件名

作用：查看文件内容, 结束查看：ctrl+c

 参数：从最后一行开始，默认后10行

          -n 数字 显示多少行

          -f 动态显示数据（不关闭） 常用来查看日志

         -1 可以查看最后一行

### 8.cp -复制文件

cp

语法： cp  源文件  目标文件

作用：复制文件

参数： -r 包含子目录和文件

           \* 匹配多个

实例: cp /root/\*.txt /opt/

说明:把/root 目录下以.txt的文件复制到/opt/目录下;

### 9.rm -删除文件或目录

rm

作用：删除文件或目录

语法：rm -rf 文件或目录名

          -r 递归删除（可以删除目录和目录里面的东西）

          -f 强行删除

### 10.mv -重名名文件或目录

mv

作用：重命名

语法：mv 源(文件或目录名)   目标（文件或目录名）

## 文件查找方法

### 1.which

语法：which 命令

作用：查看可执行文件位置

实例：

[root@VM-0-10-centos ~]# which useradd

/usr/sbin/useradd

### 2.whereis

语法：whereis命令

作用：查看可执行文件位置及相关文件

实例：

[root@VM-0-10-centos ~]# whereis useradd

useradd: /usr/sbin/useradd /usr/share/man/man8/useradd.8.gz

### 3.grep

语法：grep 关键字 文件

参数：-v 反转(取反)

作用：过滤

实例：grep 8 a.txt

说明：查找指定文件中包含指定关键字的行

实例：grep -v 8 a.txt

说明：查找指定文件中不包含指定关键字的行

实例：grep ^# a.txt

说明：查找以#开头的行

实例:grep ^$ a.txt

说明：过滤空行(以$开头,刚开始就结束了)

实例：grep #$ a.txt

说明：过滤以#结尾的行

实例：cat a.txt | grep 5$

说明：查看文件并且过滤以5结尾的行

### find

#### 语法

语法：find pathname -options [-print]

参数：pathname：find命令所查找的目录路径；如用.表示当前目录;用/表示根目录；

选项：-name 按照文件名字查找文件

      -perm 按照文件权限来查找文件

      -user  按照文件属主来查找文件

      -group 按照文件所属用户组查找文件

      -mtime -n +n 按照文件的更改时间来查找文件

        -n表示文件更改时间距现在n天以内

        +n 表示文件更改时间距现在n天以前

      -type 表示查找某一类文件

         b-块设备文件

         d-目录

         c-字符设备文件

         p-管道文件

         I-符合链接文件

         f-普通文件

         -size  n:[c] 查找文件长度为n块的文件，带有c时表示文件长度以字节计。

         -exec

说明：find命令时在目录结构中搜素文件，并执行指定的操作。

      Find命令提供了相当多的查找条件，功能很强大。

#### find常用参数

###### 使用name选项

实例：find ./ -name “\*.txt”

说明;查看当前目录下的以txt结尾的文件。“”必须时英文下的。

实例：find /etc/ -name “host\*”

说明：在ect目录中查找 文件名以host开头的文件

###### 使用perm选项

实例：find ./ -perm 755

说明：1.在当前目录下查找权限为755的文件,

属主可以读、写、执行，其他用户可以读、执行文件。

2.按照文件权限模式查找文件时，最好使用十进制的权限表示法。

###### 使用user选项

实例：find /home/ -user san

说明：按照文件属主去查找文件,在$home目录中查找文件属于san的文件

###### 使用group选项

实例：find /home/ -group mk

说明：按照文件所属组去查找文件，在$/home目录中查找文件属于mk用户组的文件

###### 按照更改时间或访问时间等查找文件

如果希望按照更改时间来查找文件，可以使用mtime,atime或ctime选项。

如果系统没有可用空间了，很有可能某一个文件的长度在此期间增长迅速，这时可以使用mtime选项来查找这样的文件。

**用减号-来限定更改时间在距今n日以内的文件**

**用加号+来限定更改时间在距今n日以前的文件**

实例：find /root/ -mtime -5

说明：在/root目录下查找更改时间在5日以内的文件

实例：find / -mtime +3

说明：在系统根目录下查找更改时间在3日以前的文件

###### 使用type选项

-type 表示查找某一类文件

         b-块设备文件

         d-目录

         c-字符设备文件

         p-管道文件

         I-符合链接文件

         f-普通文件

实例：find /etc -type d

说明：在/etc目录下查找所有的目录

# 远程连接linux操作系统工具

## 1.常用的SSH客户端软件

### Xshell（xmanger系）

远程调用lInux图形界面

gnome-terminal

### 2.SecureCRTPortable（CRT）

其他：<https://www.v1tx.com/post/best-ssh-client/>

## 2.常用的FTP客户端软件

### Xftp（xmanger系）

### SecureFXPortable

rz/sz 命令上传下载文件

其他：<https://www.v1tx.com/post/best-ftp-client/>

# Linux中vim编辑器的基本操作

## 1.VIM工具的安装

**1.首先确保系统已经安装了VIM工具**

which命令用于查找并显示给定命令的绝对路径

[root@VM-0-10-centos ~]*# which vim*

/usr/bin/vim

[root@VM-0-10-centos ~]*# rpm -qf '/usr/bin/vim'*

vim-enhanced-7.4.629-6.el7.x86\_64

或者

[root@VM-0-10-centos ~]*# rpm -q `which vim`*

package /usr/bin/vim is not installed

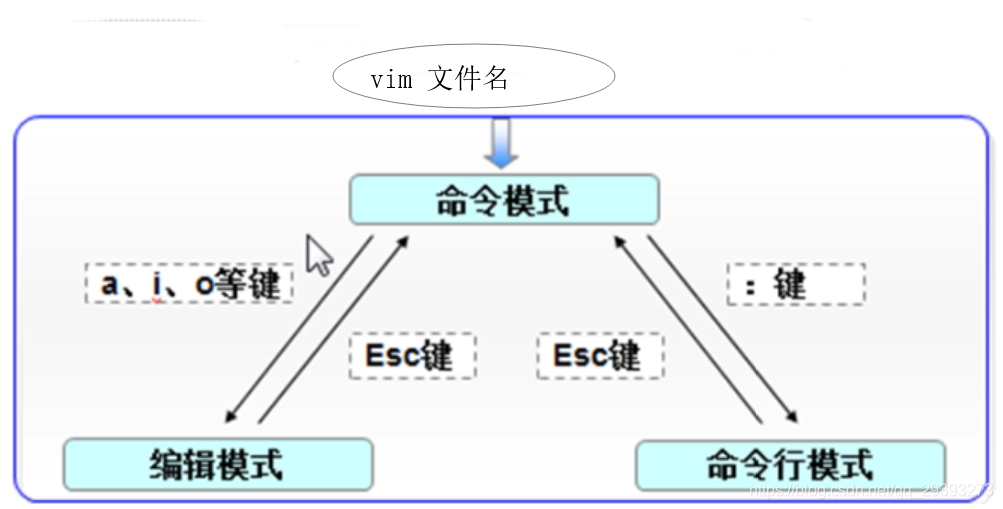
!$ ：表示上一个命令的最后一个参数

**2. 如果没有安装**

rpm -ivh /mnt/Packages/vim-enhanced-7.4.160-1.el7.x86-64.rpm

前提：是你的包已经挂载到/mnt目录下了;

## 2. vim 编辑器模式切换



**vim中的模式**

 1.首次进入文件:命令模式（vim）

 2.输入"："表示 命令行模式

 3.出现 Insert ：编辑模式

**Vim中的模式切换**

1. 切换至命令模式 输入：
2. 切换至编辑模式  输入 a i o
3. 从编辑模式到命令模式怎么切换？
4. 编辑模式--》esc--》命令模式---》：命令行模式

## 3.VIM在命令模式中的操作

1. i：当前字符之前插入（光标前）
2. I：行首插入（行首）
3. a ：当前字符之后插入（光标后）
4. A:行尾插入（行尾）
5. o:下一行插入（另外一行）
6. O:上一行插入（上一行插入）
7. 常用 i o
8. x 向后删除一个字符 等同于delete
9. X 向前删除一个字符
10. u 撤销一步 每按一次就撤销一次;
11. 快速行首行尾

a)home键或^ 行首

b)$行尾 end键

## 4.vim在编辑模式中的操作

**i 进入插入(编辑)模式**

### 删除（剪切）

 删除（剪切）

 dd 删除一行 数字+dd 删除多行

删除5行 5dd

### 复制粘贴

 yy 复制一行 数字+yy 复制N行

 复制三行 使用3yy

P 将复制的行粘贴

### 拓展-剪切

拓展：剪切

先删除再剪切

删除到行首 d+home或^(shift+6)

删除到行尾 d+end或$(shift+4) 或**shift+D**

### 词操作

注意词按空格为分隔符 每个空格前都是一个词；

w 切换单词

dw 删除一个单词，删除时要将光标移动到这个词的行首;

如果光标不再行首，则删除光标之后的字母;

yw 复制一个单词

### 块操作

**脚本中的注释信息，怎么批量的额添加注释符合**

进入v模式 移动光标选择区域

编程的时候需要进行多行注释

1.注释：ctrl+v 进入编辑模式

2.向下或向上移动光标

3.把需要的注释的行的开头标记起来

4.然后按大写的I

5.再插入注释符，比如'#'

6.再按Esc,就会全部注释了;

### 删除多行注释

删除：按ctrl+v 进入编辑模式，向下或向上移动光标;选择注释部分后按 d,就会删除注释符合;

## 5.VIM命令在命令行模式操作

1. :w 保存 save
2. :q 没有进行任何修改，退出quit;
3. :q! 修改了，不保存，强制退出
4. :wq 保存并退出
5. :wq! 强制保存并退出;
6. :ZZ 保存
7. :set nu 显示行号
8. :set nonu 不显示行号
9. :/正向查找内容 :/root n往下查找，N往上查找
10. ：noh 取消高亮显示
11. 如果不小心打开了目录，直接退出就可以了;

vim 中定位到某行

1. gg 定位到行首
2. G  定位到最后一行，行首
3. 数字+gg 定位到某一行3gg

**命令需要在命令模式下操作**

# Linux用户管理

## Linux账号管理

### 1.Linux用户的分类

1.根据账号位置分：本地账号、远程（域）账号

2.根据账号的功能:

  超级用户(root) uid:0

  普通用户

     系统用户：uid 1-999  （RHEL6 1-499）

     普通用户： uid ：1000+ （RHEL6 500）

3.说明：

  UID：每个用户的身份标识,类似于每个人的身份证号码;

  超级用户：root 至高无上;

  本地用户：用管理员创建,权限受到一定的限制,在自己的宿主目录时有完整的权限;

  系统用户：一般是不会登录系统的，用于维持某个服务程序的正常运行;

### 2.系统用户的作用

**系统用户（伪用户）--为了安全**

1. Linux中任何一个命令的操作都必须有一个用户的身份;
2. 伪用户一般和系统或者程序服务相关
3. bin,daemon，shutdown,halt linux 默认都有这些伪用户;
4. 伪用户通常不需要或不可以登录的;

### 3.用户和组的相关配置文件

 账号信息                密码信息

  用户：/etc/passwd     /etc/shadow

  组：/etc/group        /etc/gshadow

### 4.用户账号管理

#### 添加用户

语法：useradd 用户名

说明：添加用户

常用参数：

  -u    UID

  -d    宿主目录

  -g    起始组  #只能有一个

  -G    附加组 #可以有多个

  -s    登录shell

创建一个用户

实例： useradd SAN

说明：创建一个用户 SAN

创建后可以查看用户的宿主目录 ls /home/

创建一个用户并指定用户的uid

实例：adduser -u 2018 mk

说明：创建一个用户并指定用户的uid

      可使用id 账号 查看用户的UID

查看用户uid

语法：id 账号

说明：查看用户的uid

实例：id mk

结果:  uid=2018(mk) gid=2018(mk) 组=2018(mk)

创建一个用户并指定用户的宿主目录

实例：adduser good -d /tmp/good

说明：指定用户的宿主目录

创建一个用户并指定用户的起始组

实例：adduser -g 2018 xiaoyifeng

说明：指定用户的起始组

创建一个用户并指定用户的附加组

实例：adduser -G root mk

说明：指定用户的附加组，开源 一切都可以修改;

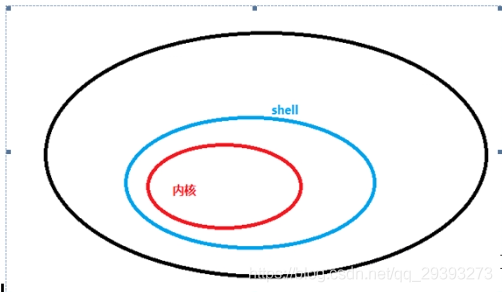
创建一个用户，指定用户的登录shell

1.查看当前系统的shell

chsh -l

2.指定用户的登录shell（壳）

adduser -s /sbin/nologin zabbix



#### 创建用户的另外的命令

useradd 或adduser

#### 删除用户

实例：userdel san  ---不会删除宿主目录

说明：删除用户

参数： -r  --连同宿主目录一起删除;

#### 切换用户

语法：su 用户

实例：su -aaa

说明：加-,在切换时会包环境变量一起进行切换

     不加，保留原本的环境变量;

#### 查看用户信息相关 命令

1. id 用户名

 作用：查看用户信息

2.w 查看登录系统的用户及正在执行的命令

3.who  查看登录系统的用户

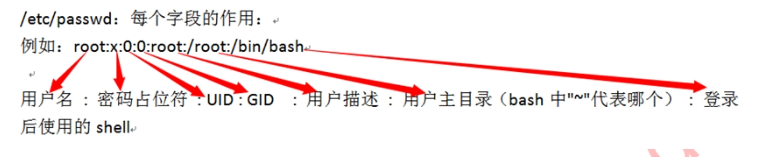
4.whoami 查看当前登录的用户名

### 5.passwd文件

用户账号的全部信息被保存在/etc/passwd文件,这个文件以如下格式保存了每个系统账号的所以信息（字段以":" 分割）；

root:x:0:0:root:/root:/bin/bash

用户名：密码占位符:UID:GID:用户描述：用户主目录(bash中'~'代表那个)：登录后使用shell;



### 6.Linux shadow文件结构的说明

/etc/shadow

root:$1$BARFOzwC$EMFOKpNaHHEtV7L6MGcI90:19025:0:99999:7:::



**同一个密码，得到的加密后的密码是不一样的;**

### 7.修改用户密码的两种方式：

方式1：

语法：passwd 用户名

说明：回车设置密码

方式2:

语法：echo 密码 | passwd--stdin 用户名

说明：修改密码

## Linux组管理

### 组的分类

根据账号位置分：本地组、远程（域）组

根据账号的功能：超级用户组（root）GID 0

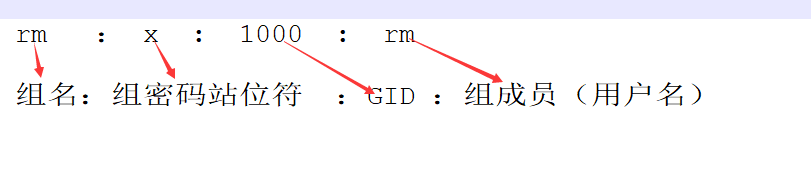
普通用户组 gid:1-999

本地用户: gid:1000+

**每一个用户都有一个同名的组,除非额外指定；**

vim /etc/group 配置文件结构

grep rm /etc/group



### 创建和删除组

创建组：groupadd 组名

删除组：groupdel 组名

密码存放在/etc/gshadow;

## 控制添加用户规则的文件

当添加新的账号时,默认会执行下列操作

1.它的主目录就会被创建（一般是"/home/用户名"，除非特别指定）

2.一些隐藏文件 如.bash\_logout,.bash\_profile以及.bashrc会被复制到用户的主目录；

3.会创建一个和用户名同样的组（除非你给新创建的用户指定了组）

## 解决模板文件(隐藏文件)被删除之后不正常的问题

1.rm -rf /home/用户名/

rm -rf /hom/mk/.bash\*

2.cp /etc/skel/.bash\*   /home/用户名下

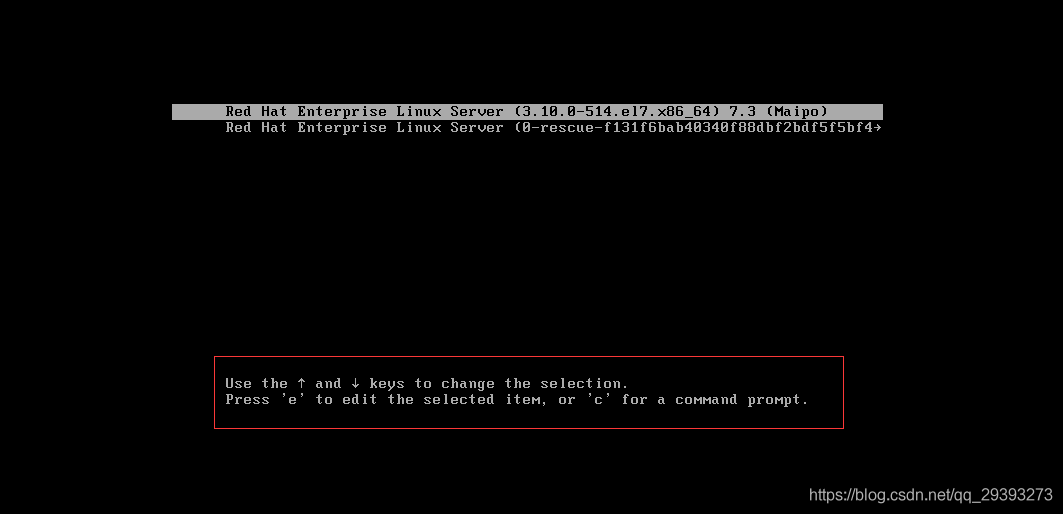
   cp /etc/skel/.bash\* /home/mk/

3.chown 用户名：用户名  /home/用户名/.bash

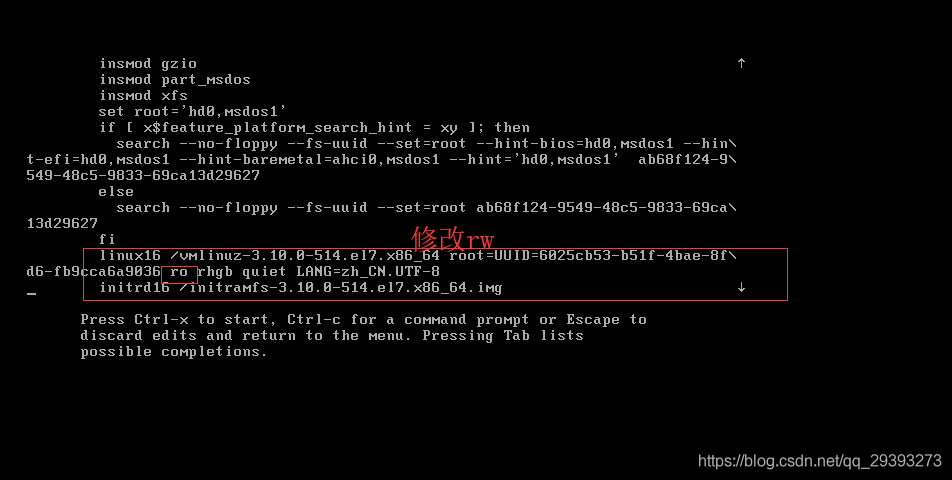
   chown mk:mk /home/mk/.bash\*

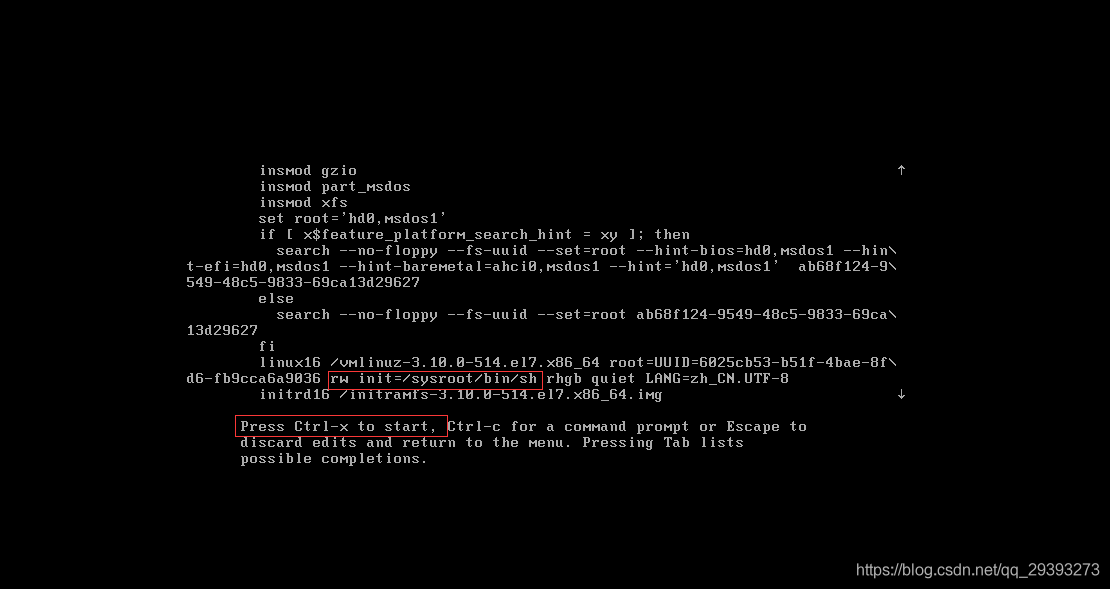
## Linux恢复root密码

1.重启,安pg 键,进入如下界面,按下e键进行编辑



2.在此界面找到ro这一项,并进行修改,改为 rw init=/sysroot/bin/sh





3.改完之后，按下Ctrl+X进入紧急模式

原理：启动了一个shell环境，系统并没有真正的启动;

4.换根，修改密码；

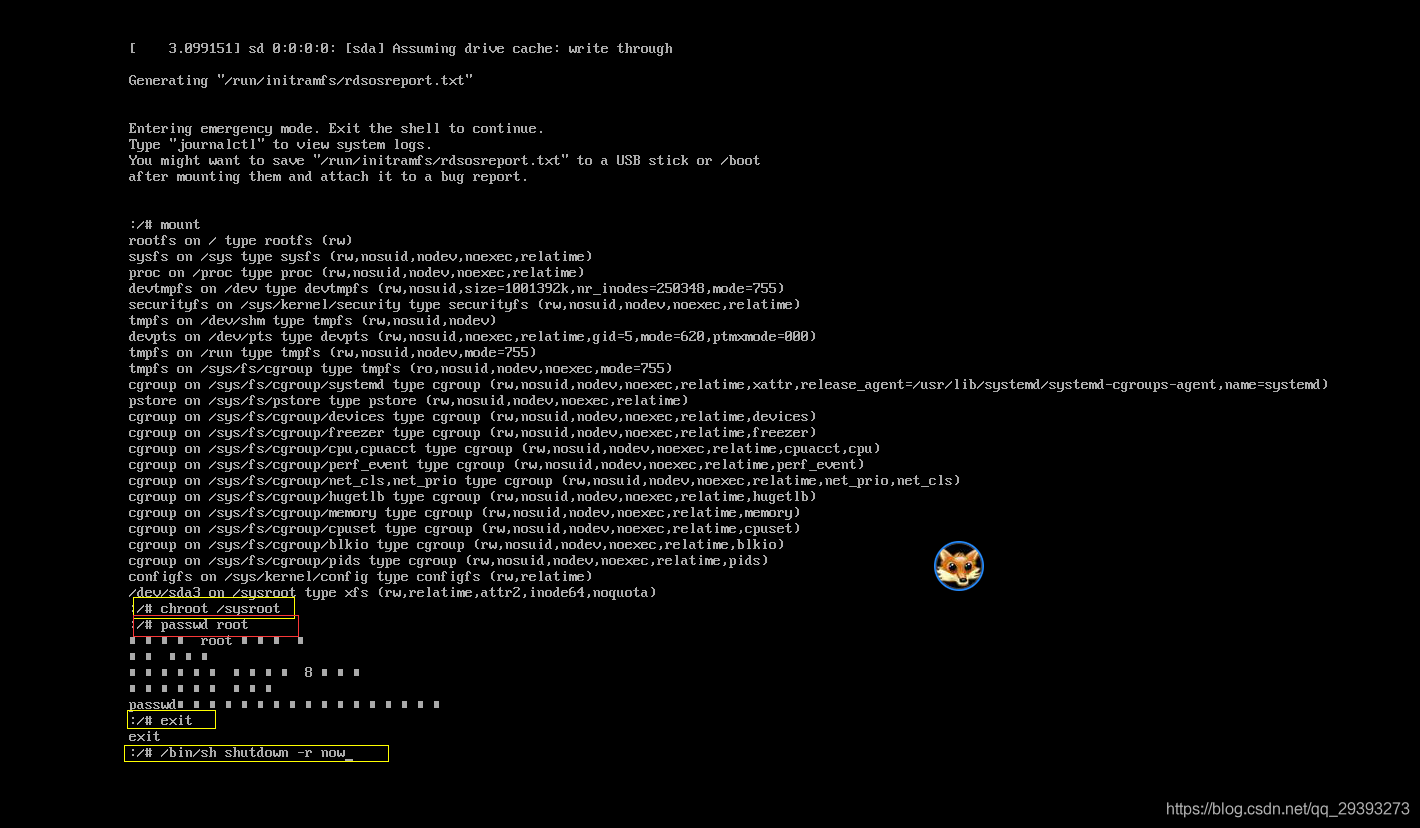
1.使用 mount可以查看一些信息

2.chroot 命令用来在指定的根目录下运行指令;

chroot 即change root directory(更改root目录)。在linux系统中，系统默认的目录结构都是以/,即是以根（root）开始的。而在使用chroot之后,系统的目录结构将以指定的位置作为/位置在经过chroot命令之后，系统读取到的目录和文件将不是在旧系统根下的(即被指定的新的位置)的目录结构和文件；

chroot /sysroot

passwd 用户名 回车输入密码



5.重启系统

注：恢复密码时需要把selinux关闭

  a.退出当前根 exit

  b.执行/bin/sh shutdown -r now

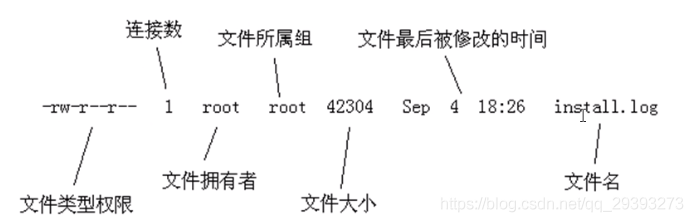
# 文件的基本权限管理

## 1.查看文件权限信息

语法：ll 文件名

 说明：相当于ls -l

-rw-r--r-- 1 root root 0 SEP 4 18:26 install.log



**文件基本的权限**

- rw- r-- r-- root root filename

类型 拥有者权限 所属组权限 其他人的权限 拥有者 属组 文件名

## 2.文件类型

-rw-r--r-- 1 root root 0 Oct 19 09:35 a.txt

dr-xr-xr-x. 5 root root 4096 Feb 2 2022 boot

-:表示文件

d:表示目录 wxr

### 对于文件

r:读  w:写   x:执行

 读：可以查看,不能更改,删除;

 写: 可以插入;

 执行：一般指的是脚本文件;

### 对于目录：

r读（可以看到目录里面有什么及内容）ls

w 创建,删除,移动   touch mkdir rm mv cp

x 进入             cd

## 3.用十进制的数字表示权限

r 可读  ,w可写 ,x 可执行,也可以用十进制的数字代表权限 0，1，2，3，4，5，6，7

    0  代表什么权限也没有

    1  x 可执行的

    2  w 可写的

    3  w+x wx 可以执行可写

    4  r 可读的

    5  r+x

    6  r+w

    7  r+w+x

-rw-r--r-- 1 root root 0 Oct 19 09:35 a.txt

- rw- r-- r-- root root a.txt

类型 拥有者的权限 所属组的权限 其他人的权限 属主 属组 文件名称(对象)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| rwx |  |  |  |  |
| r-- | -w- | --x |  |  |
| 100 | 010 | 001 | 二进制 | 进制转换器 |
| 4 | 2 | 1 | 十进制 | 4+2+1=7 |
| rw- |  |  |  |  |
| r--- | -w- | --- |  |  |
| 100 | 010 | 000 | 二进制 | 进制转换器 |
| 4 | 2 | 0 | 十进制 | 4=2=6 |
| r-x |  |  |  |  |
| r-- | --- | --x |  | 4+1=5 |
| rw-r--r-- |  |  |  |  |
| rw- =4+2 | r-- =4 | r--=4 | rw-r--r-- =644 |  |

rwx

r-- -w- --x

100 010 001 二进制 进制转换器

4 2 1 十进制 4+2+1=7

rw-

4+2=6

r-x

4+1=5

rw-r--r--

rw- =4+2=6 r-- =4 r-- =4

rw-r--r-- =644

## 4.chmod修改文件权限

语法：chmod 权限 文件名

  实例：chmod 622 a.txt

  参数：-a  所有的权限

   -u  文件所有者的权限

   -g 组权限

   -o 其他用户权限

## 5.使用运算符来设定权限+-=

 u-w      user     拥有者

    g+x      group    组

    o=r      other    其他人

    a+x      all      所有人

实例：

--查看文件的权限信息 ll a.txt

 --用户删除写的权限 chmod u-w a.txt

 --组添加可执行权限 chmod g+x a.txt

 --其他人没有读取的权限  chmod o-r a.txt

## 7.chmod修改目录的权限

1.查看目录的权限

    ll -d 目录

    ll -d test/

2.修改目录权限

语法：chmod 权限 目录

    ----删除拥有着的写的权限

    chmod u-w test/

    ll -d test/

    ---一次修改多个权限

    chmod u=rwx test/

## 8.chown 修改文件的拥有者和所属组

1.语法：chown user：group 对象

       chown 用户：组  文件

  说明:修改文件拥有者和所属组

2.语法：chown 用户 对象

说明：单独更改属主

3.语法：chown ：组 对象

说明：单独更改所属组

4.实例

1.修改文件拥有者和所属组

  chown mk:bin a.txt

 2.单独修改拥有者

  chown deemon a.txt

3.单独更改组

  chowm:sys a.txt

## 9.-R 递归操作

语法：chown 用户名 目录 -R

实例：chown manzhe wang/ -R

说明：将指定目录下的内容的所有者改为指定用户

## 10.一个文件只有读的权限，拥有着可以写这个文件吗？

一个文件只有读的权限，拥有者可以写这个文件（文件所有者一定可以写文件）

保存时用wq!

## 11.root用户和普通用户创建文件的权限相同吗？

使用root创建文件和普通用户创建文件的权限时不一样的。

(设置文件默认权限的补码);

# Linux软件包管理

## Linux软件包管理方式

1.RPM和YUM两种方式安装软件

2.源码编译安装软件

## RPM和YUM两种包管理区别

1.主要有RPM和YUM两种包管理;两种包管理各有用处,其中主要区别是：

   a.YUM使用简单但是需要联网,YUM会去网上的YUM包源去获取所需要的软件包。

   b.RPM的需要的操作精度比较细，需要我们做的事情比较多;

## RPM软件包管理

RPM 即Redhat Package Manager简称。

### 挂载光盘

umount/dev/sr0 卸载

 mount /dev/sr0   /mnt/ 挂载

 cd /mnt/Packages

### rpm包名字结构

zsh-5.0.2-25.el7.x86\_64.rpm

zsh      -5          .0         .2       el7     x86              64

软件名    主版本号    次版本号   修订号   RHEL7  CPU架构平台   支持系统位数

### rpm安装软件

语法： rpm [OPTION...] 软件包

参数： -i,---install 安装软件包

      --nodeps       不验证软件包依赖

      -v,--verbose    提供更多的详细信息输出

      -h,--hash       软件包安装的时候列出哈希标记

实例：rpm -ivh zsh-5.0.2-25.el7.x86\_64.rpm

rpm -ivh  mariadb

说明：1.**需要自己来处理依赖关系**

2.方便的解决方案

        yum install mariadb-5.5.52-1.el7.x86\_64.rpm

### rpm软件包下载网站

如果没对应的rpm包可以到rpm包相关网站下载软件包

<http://rpmfind.net/>

<http://rpm.pbone.net/>

<http://www.rpmseek.com/index.html>

### rpm的查询功能

语法：rpm -qa 软件包名称

参数：-a 查看所有已经安装的软件包

           -f 查询 文件所属软件包

           -p  查询软件包（通常用来查看下还没有安装的软件包）

           -i  显示软件信息

           -l显示软件包中的文件列表

           -d 显示被标记未文档的文件列表

           -c 显示被标记未配置文件的文件列表

说明：通常可以配合管理|more 来使用，使得结果更容易读;

实例：rpm -qa zsh

说明：查看是否安装了zsh软件包

实例： which zsh

            rpm -qf /usr/bin/zsh

            或直接

           rpm -qf `which zsh`

说明：查看zsh命令是那个目录生成的

### rpm 包 升级

rpm -Uvh  /mnt/Packages/mariadb-5.5.52-1.el7.x86\_64.rpm

### rpm 包 卸载

rpm -e zsh （包名）

## 安装YUM 安装软件

YUM解决依赖关系问题、自动下载软件包。基于c/s架构;（client/ftp,file,http）

（repodata目录中存放了软件包的依赖关系，所以，可以自动处理依赖）

**1.编写YUM配置文件**

**在**/etc/yum.repos.d/编写以.repo结尾的yum配置文件。

vim /etc/yum.repos.d/redhat.repo

 [rhel7-yum] #yum源名称，唯一的,用来区分不同的YUM源;

      name=rhel-server #描述信息 可以随意编写;

      baseurl=file:*///mnt #yum源的路径 （repodata目录所在的目录）*

                            #或者baseurl=http:*//192.1668.1.63/pub*

                            #或者baseurl=ftp:*//192.1668.1.63/pub*

      enabled=1 #1表示启用yum源

      gpgcheck=0 #0取消验证 #1，表示使用共钥检验rpm的正确性

2. 检查是否配置成功，如果清空yum缓存后能生产列表表名配置成功。

yum clean all #清空yum缓存

yum list      #生成列表

## YUM的使用

**需要连网，操作与RPM类似,只是会自动安装依赖项**

 1.安装 yum install -y

 2.检测升级 yum check-update

 3.升级  yum update

 4.软件包查询 yum list

 5.软件包信息yum info

 6.卸载 yum remove

 yum remove httpd -y

7.帮助 yum -help|man yum

8.安装一组软件包

   查看包组 yum grouplist

       yum groupinstall`Security tools`

          或者

          yum groupinstall`安全工具`-y

## 源码编译安装软件

**优点：可以安装最新版本，灵活可以自己定义安装参数（指定安装目录|支持某些库）**

### 安装步骤

**1.解压解包**

tar -zxvf 源代码包的安装

   x 为解包

   z 为解压（仅适用于gzip,j适合bz2）

   v 显示过程

   f 指定被解压包名

**2.配置**

进入解压后的目录，用./configure[--prefix=/usr/local/filename]，

这个过程主要是收集系统信息，设置安装目录等（卸载的时候只需要将这个目录删除即可）

**3.编译：源代码编译为可以执行的二进制文件**

 make -j 4

-j 4 表示以4个进程同时编译，速度快，-j后的数字要和你的cpu的核数一样.

**4.安装**

make install

### 安装nginx 服务器

检查依赖

有时候可能需要配置临时的ip地址

ifconfig eno\*\*\*\*：1 ip地址

* 1. tar -zxvf 包名

2./configure

error:the HTTP rewrite module requires the PCRE library.

yum -y install pcre-devel zlib-devel

rpm -ivh /mnt/Packages/zlib-devel-1.2.7\*\*\*.rpm

添加运行用户

nobody 建议添加用户 nginx

useradd -s /sbin/nologin -M nginx

-M ：不产生家目录

指定安装目录

./configure --prefix=/usr/local/nginx --user=nginx --group=nginx

echo $? 查看编译是否成功 0成功

make -j

总结

编译安装Nginx

tar解压--》./configure配置--》编译生成 makefile 文件--》make install 安装

--prefix 指定安装位置

--user 指定运行的用户

--group 指定运行的组

iptables -F

netstat -antup | grep 80

看到4个0 且是LISTEN 说明Nginx服务启动成功

cp /usr/local/nginx/sbin/nginx /usr/sbin/

可以直接启动

nginx -s stop/start

nginx -t 检查配置文件是否有问题

# Linux中文件的归档和压缩

文件的归档命令tar

压缩命令

## 1.归档的好处

a.方便使用、查询、阅读

 b.易于管理（批量删除文件）

## 2.为啥要压缩？

 a.文本文件压缩后很省空间;

   b.图片、视频 压缩后可能占用更大的空间,因为图片和视频本身就是压缩格式文件；

   3.主机A要跟主机B传输一个大小为10G的文件估计传送100s,如果直接传递会大量的占用流量宽带，导致公司的内网访问速度缓慢;这时如果将传输前压缩---》传输后解压，可以达到把10G的文件压缩成5G，传送的时间时50s;

总结(文件压缩的好处)：

   a.节约硬盘资源

   b.加快文件传输效率;

## tar 命令的使用

tar 文件是把几个文件和（或）目录集合一个文件夹里，是创建备份和归档的最佳工具;

tar --help

### 打包

语法：tar 选项  包名称 目标文件/目录

参数: c create 创建

      v 详细

      f  filename

实例：tar cvf grub2.tar  /boot/grub2/

说明：将 /boot/grub2/ 打包成grub2.tar

### 查看文件或目录大小

语法：du 文件|目录

参数：s:显示目录的总大小

      h:方便阅读

实例：du  -sh /boot/grub2/

说明:查看指定目录大小

### 确定文件类型

file 命令

语法：file 文件名

作用：确定文件类型

说明：a.linux系统不根据后缀名识别文件类型；

     b.用file命令查看文件类型

### 把两个目录或目录+文件打包成一个软件包

tar -cvf aa.tar /boot/ /etc/passwd

说明：将/boot/和/etc/passwd打包为aa.tar

### 查看tar中的内容

1.不解包，查看tar中的内容

  tar tvf grub2.tar

  说明：查看grub2.tar中的内容

2.解包

tar xvf grub2.tar

说明：查看grub2.tar,并解压到当前目录

3.解包 指定路径

tar xvf grub2.tar -C /opt/

说明：查看grub2.tar,并解压/opt/目录

4.对比文件的大小

1.使用du命令查看文件夹的大小

du -sh /boot/grub2

2.查看文件的大小

      ll -h grub2.tar

### tar 归档+压缩

1.文件格式：gzip zbip2 zip tar

2.压缩格式：gz,bz2,xz,zip,Z

#### gz压缩与解压方法：

格式（文件名格式）:压缩包名称.tar.gz 或压缩名.tgz

**压缩：**

语法： tar zcvf newFile.tar.gz SOURCE

参数：z: 通过gzip过滤归档

实例：tar zcvf grub2.tar.gz  /boot/grub2/

说明：将目录或文件打包并压缩为\*.tar.gz

对比大小：

       ll -h gurb2.tar\*

**解压**

语法：tar zxvf 压缩文件名.tar.gz

说明：解压到当前目录

语法：tar zxvf 压缩文件名.tar.gz -C 指定目录

说明：解压到指定目录

实例：tar zxvf grub2.tar.gz -C/opt/

说明：将压缩文件解压到指定目录

#### bz2压缩和解压方法：

格式（文件名格式）:压缩包名称.tar.bz2

**压缩：**

语法：tar jcvf newfile.tar.bz2 RESOURCE

参数：j通过bzip2过滤归档

实例：tar jcvf grub2.tar.bz2 /boot/grub2

说明：将文件或目录打包压缩为\*.tar.bz

对比大小 ll -h grub2.tar\*

**解压：**

语法：tar jxvf  \*.tar.bz2压缩文件 选项

参数：x：从存档中提取文件

实例： tar jxvf grub2.tar.bz2

说明：解压文件到当前目录

语法： tar jxvf grub2.tar.bz2 -C/opt/

说明：解压文件到指定目录

#### zip 软件包解压缩命令

**zip是压缩程序，unzip 是解压程序；**

**压缩文件**

语法：zip 压缩名.zip 文件

实例： zip passwd.zip /etc/passwd

说明：压缩文件

**压缩目录**

语法：zip 选项 名称  源

选项：-r 压缩目录

实例：zip -r grub2.zip /boot/grub2/

说明：压缩目录

比较大小：ll-h grub2.\*

**解压**

语法：unzip 包名称.zip 选项

选项：-d 指定解压目录

实例：unzip grub2.zip -d /opt/

#### Linux系统能不能解压rar格式的压缩包?能的话，怎么解压？

#### 其他压缩命令

补充

缩命令 gzip bzip2 xz

语法格式：

    gzip 文件

    bzip2 文件

    xz     文件

用法：和zip的使用方式一样

解压：gzip -d 文件

     bzip -d 文件

     xz    -d  文件 或unxz 文件

不解压的情况查看压缩文件的内容分别用zcat，bzcat，xzcat

# 进程管理命令ps和top

## 程序和进程

程序：程序是为了完成某种任务而设计的软件比如Open Office，QQ;

进程：进程是运行中的程序;一个运行着的程序，可能由多个进程;

## 进程的属性

1. 进程ID（PID）：是唯一的数值,用来区分进程;
2. 父进程和父进程的ID（PPID）;
3. 启动进程的用户ID（UID）和所归属的组（GID）
4. 进程状态：状态分为运行R,休眠S,僵尸Z;
5. 进程执行的优先级;
6. 进程所连接的终端名;
7. 进程资源占用：比如占用的资源大小（内存，CPU占用量);

## 父进程和子进程

他们的关系是管理和被管理的关系，当父进程终止时，子进程也随之而终止。但子进程终止，父进程并不一定终止;比如httpd服务器运行时，我们可以杀掉其子进程，父进程并不因为子进程的终止而终止;

在进程的管理中，当我们发现占用资源过多，或无法控制的进程时，应该杀死它，以保护系统稳定安全运行;

对于Linux进程的管理，是通过进程管理工具实现的；比如ps,kill等;

## Ps top管理进程

ps 提供了进程的一次性的查看，它所提供的查看结果并不是动态连续的；如果想要对进程的时间监控应该用top工具;

### ps 监视进程工具

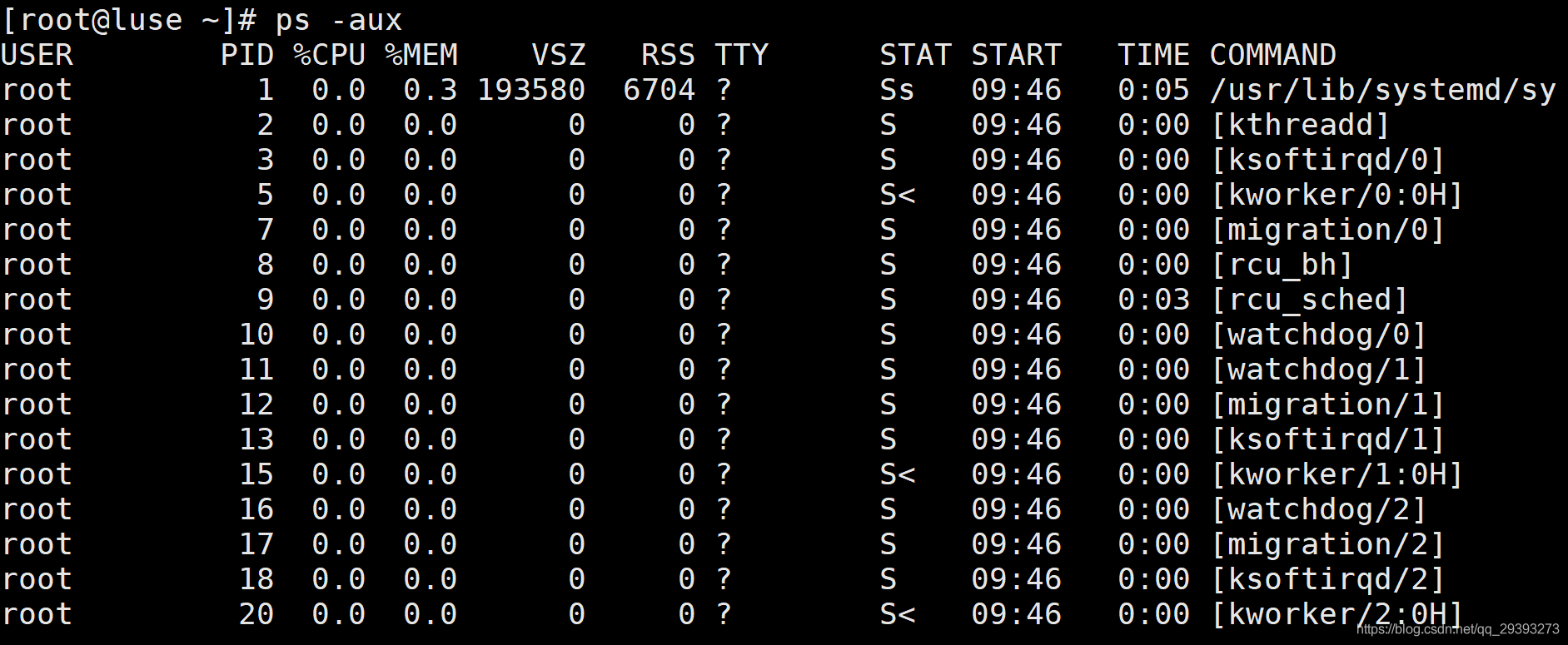
ps查看结果是静态的、当前状态的。

ps的参数说明

ps提供了很多的选项参数,常用的有以下几个：

1. l 长格式输出；
2. u 按用户名和启动时间的顺序来显示进程；
3. f 用树形格式来显示进程；
4. a 显示所有用户的所有进程（包括其他用户）
5. r 显示运行中的进程;
6. -e显示所有进程，包括没有控制终端的进程
7. -x 显示没有控制终端的进程
8. 常用选项是组合aux

列出目前所有的正在内存当中的进程;



1. USER：进程属主
2. PID：进程的ID
3. PPID：父进程
4. %CPU ：进程占用的CPU的百分比；
5. %MEM：占用内存的百分比
6. NI：进程的NICE(优先级)值
7. VSZ：进程占用虚拟内存的大小
8. RSS：固定的内存使用数量;
9. TTY:进程启动的终端ID;
10. STAT:进程状态;

R:运行状态

S：处于休眠状态；

T：停止或被追踪；

Z：僵尸进程;

N:优先级较低的进程;

L:有一些页面被锁进内存;

s:进程的领导者（在它之下有子进程）

1. Start：启动进程的时间
2. Time：进程消耗CPU的时间；
3. COMMAND 命令名称和参数;

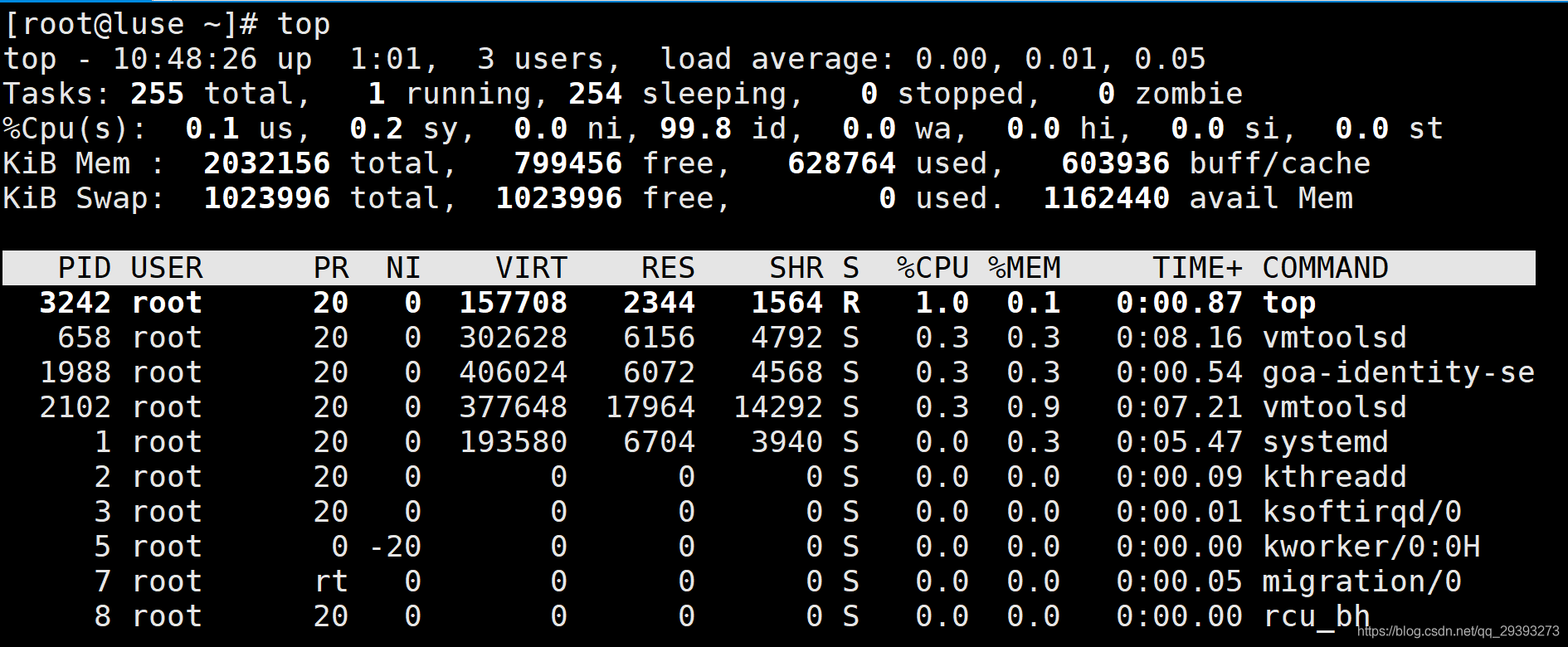
拓展

* ps -aux 是用BSD的格式来显示进程;
* ps -ef  是用标准的格式显示进程;
* ps -aux | grep httpd 使用管道进行过滤，grep可以过滤关键字
* pstr 清楚的列出进程

### top 动态查看进程

统计信息区前五行是系统整体的统计信息；

使用：top 回车



#### 系统的信息解释

##### 任务队列信息（第一行）

top - 10:50:54 up  1:04,  3 users,  load average: 0.02, 0.02, 0.05

1. 10:50:54 :当前时间
2. up  1:04 :系统运行时间;格式为时:分;
3. 3 users  :当前登录用户数;
4. load average: 0.02, 0.02, 0.05 系统负载,即任务队列的平均长度，三个值分别为1分钟，5分钟，15分钟前到现在的平均值。(对于一个4核心CPU，如果负载为12时,队列太长，需要加CPU，3\*核数）

##### 进程和CPU的信息（第二行和第三行）

当有多个cpu的时候，这些内容可能超过两行;

 Tasks: 253 total,   1 running, 252 sleeping,   0 stopped,   0 zombie

%Cpu(s):  0.3 us,  0.3 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st

1.Tasks: 253 total:进程总数

      1 running :正在运行的进程数；

      252 sleeping：正在休眠的进程数；

      0 stopped： 已经停止的进程数；

      0 zombie ：僵尸进程数；

2.%Cpu(s):  0.3 us ：系统用户进程使用CPU百分比

            0.3 sy ：内核中的进程占用的CPU百分比

            0.0 ni ：用户进程空间改变过优先级的进程占用的CPU百分比

            99.3 id：空闲CPU百分比

##### 第四行和第五行（内存信息）

Linux 使用内存的思想是：尽最大可能使用内存。

 KiB Mem :  2032156 total,   795204 free,   633080 used,   603872 buff/cache

    KiB Swap:  1023996 total,  1023996 free,        0 used.  1158176 avail Mem

1. Mem :  2032156 total 物理内存总量;

          795204 free   空闲内存总量;

          633080 used   使用的物理内存总量;

          603872 buff/cache 用作内核缓存的内存量 和free -k 一个意思;

2.Swap:  1023996 total 交换区总量

         1023996 free  空闲交换区总量

         0 used        使用的交换区总量

         1158176 avail Mem  总的可利用内存时多少;

##### 进程信息（第六行及后面行）

1. PID 进程ID
2. USER 进程所属者用户名
3. NI： 进程优先级，nice值，负值表示高优先级，正值表示低优先级;
4. RES 实际使用内存大小;
5. S   进程状态 D=不可中断的睡眠状态；

                 R=运行

                 S=睡眠

                 T=跟踪/停止

                 Z=僵尸进程

1. %CPU 上次更新到现在的CPU时间占用百分比
2. %MEM  进程使用的物理内存百分比;
3. TIME+ 进程使用CPU时间总计，单位1/100秒
4. COMMAND 命令名/命令行

#### top的快捷键

1. 默认3s刷新一次
2. 空格：立即刷新
3. q 退出
4. M 按内存排序（大写M））
5. P 按CPU排序（大写P）

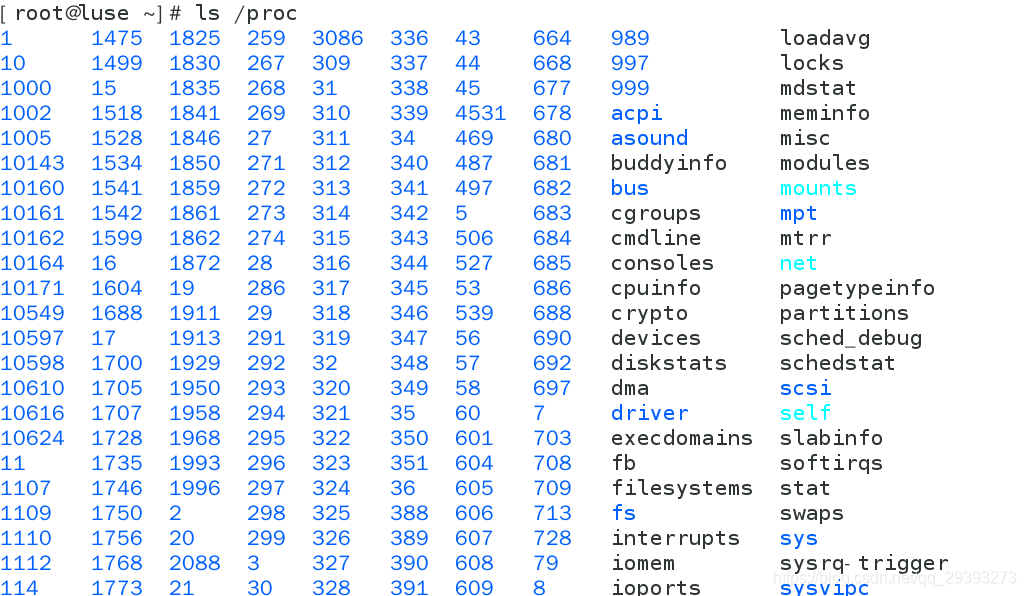
# Kill和进程优先级管理

## proc目录

这个目录比较特殊，它并不在我们的磁盘上,而在内存当中;

当前系统运行的所有进程都动态的存放在这个目录中

主要是做监控的时候可以用到；



上面的每个编号就是我们当前进程PID所命名的目录名

!加命令 可以快速调用上一个命令;

## 关闭进程

1.语法：kill [进程号]

说明：通过信号的方式控制进程;

2.语法：Kill -l

说明：列出全部的信号名称

[root@VM-0-10-centos ~]# kill -l

 1) SIGHUP  2) SIGINT  3) SIGQUIT  4) SIGILL  5) SIGTRAP

 6) SIGABRT  7) SIGBUS  8) SIGFPE  9) SIGKILL 10) SIGUSR1

11) SIGSEGV 12) SIGUSR2 13) SIGPIPE 14) SIGALRM 15) SIGTERM

16) SIGSTKFLT 17) SIGCHLD 18) SIGCONT 19) SIGSTOP 20) SIGTSTP

21) SIGTTIN 22) SIGTTOU 23) SIGURG 24) SIGXCPU 25) SIGXFSZ

26) SIGVTALRM 27) SIGPROF 28) SIGWINCH 29) SIGIO 30) SIGPWR

31) SIGSYS 34) SIGRTMIN 35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3

38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8

43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13

48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12

53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7

58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2

63) SIGRTMAX-1 64) SIGRTMAX

关闭进程：

语法：kill [进程号]

说明：通过信号的方式控制进程;用9来终止进程;

实例：kill -s 9[进程号]（强行关闭）

kill -9 [进程号] （强行关闭）

语法：killall 进程名

实例：killall httpd

说明：杀死指定名字(httpd)的所有进程

语法：pkill 进程名

实例：pkill httpd

说明：杀死指定名字(httpd)的所有进程

## 进程的优先级管理

### 进程优先级定义

cpu是分时运行的，所以可以同时运行多个程序；

进程也分快慢缓急，所以需要对进程分优先级;

### 进程优先级范围

优先级取值范围（-20-19）

越小优先级越高，默认优先级为0

设置的时候可以设置-30，单系统默认为-20

### 指定程序的优先级

1.使用nice 指定程序的优先级

2.使用top命令查看优先级

语法：nice -n command

作用：指定程序的运行优先级，n为优先级数字

实例：nice -5 command #以5级别运行;

nice -n -5 command #以-5级别运行；

操作实例：

1.指定程序的运行优先级

nice -n 5 vim a.txt

2.查看程序的优先级

ps -axu |grep vim #查看vim的进程号

top -p pid 查询运行级别；

### 改变正在运行的优先级

renice 改变正在运行的优先级;

作用：改变一个正在运行的进程的优先级;

格式：renice -n pid

语法：renice -n 优先级数字;

实例：renice -n -5 4110

说明：给pid为4110的进程设置优先级为-5

### 查看系统内存使用情况

free 查看系统内存使用情况;

free -m

# Linux中的计划任务

计划任务：在某一个时段自动执行某个任务。

Linux系统中提了at和cron,其中at时只能执行一次

## at计划任务的使用

### 语法

语法：at 时间

说明：服务atd必须开启

语法： systemctl status atd

说明：查看atd服务是否开启

语法：systemctl restart atd

说明：重启atd服务

语法：systemclt enable atd

说明：设置atd服务开机自启

### 使用at创建计划任务

[root@VM-0-10-centos ~]# date

Sat Oct 22 20:52:39 CST 2022

[root@VM-0-10-centos ~]# at 20 59  #回车

at> mkdir /root/linux1609  #输入要执行的命令

at><EOT> #按ctrl+d结束

job 1 at Sat Oct 22 20:59:00 2022

[root@VM-0-10-centos ~]# at -l #查看计划任务

1 Sat Oct 22 20:59:00 2022 a root

任务编号 执行时间 执行者

也可以这样写(在某天某事执行计划任务)

[root@VM-0-10-centos ~]# at 20:00 2022-10-28 在某一天

[root@VM-0-10-centos ~]# at now +10min 在10分钟后执行

### 删除计划任务

atrm 命令

语法：atrm +计划任务的编号

实例：trm 1

### 总结：

at计划任务执行一次，下次就不能再次执行。

## cron计划任务的使用

在生产环境中：一般都是周期性的执行某个任务，如备份日志。

周期性计划任务：有cron主程序和Cron服务构成。

### Cron计划日志的分类

1. 系统级别的计划任务
2. 用户级别的计划任务

### 系统级别

主文件：/etc/crontab

[root@VM-0-10-centos ~]# vim /etc/crontab

SHELL=/bin/bash

PATH=/sbin:/bin:/usr/sbin:/usr/bin

MAILTO=root

# For details see man 4 crontabs

# Example of job definition:

# .---------------- minute (0 - 59)

# |  .------------- hour (0 - 23)

# |  |  .---------- day of month (1 - 31)

# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...

# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sa

t

# |  |  |  |  |

# \*  \*  \*  \*  \* user-name  command to be executed

~

\* \* \* \* \*

分 时 日 月 星

1. 查看crond服务状态是否为开启

一般默认开启的，不需要我们去开启或重启，如果需要可以使用相关命令操作。

相关命令

语法：systemctl status crond

说明：查看crond服务的状态

语法：systemctl restart crond

说明：重启crond服务

语法：systemctl enable crond

说明:开机自启

### 用户级别的计划任务

对于root用户：

命令：

crontab -e 创建一个计划任务

crontab -l 显示

crontab -r 删除计划任务

语法：crontab -e

说明：创建定时任务

分 时 日 月 星 谁做后面的事情 命令

每个取值范围

分：0-59

小时：0-23

日：1-31

月：1-12

周：0-7 0,7都是周日

实例：每月9，18，22号这几天的凌晨1点1分，执行一个备份脚本。

1. crontab -e （回车）
2. 1 1 9,18,22 \*  \*  /root/back.sh
3. 保持退出

实例：每月9-22号这几天的凌晨1点1分，执行一个备份脚本

1. crontab -e （回车）
2. 1 1 9-22 \*  \*  /root/back.sh
3. 保持退出

实例：每5分钟执行一次

1. crontab -e （回车）
2. \*/5 \* \* \*  \*  /root/back.sh
3. 保持退出

### anacron

cron用于控制循环执行例行性工作，如果我们要设定机器每天早上8点进行备份服务。

除非我们的机器8点这个时间不会关机，如果关机了，cron中的脚本，在下次开机将不会被执行。

**解决cron错过时间后，不执行cron计划任务的问题**

Anacron并没有取代cron的意思，anacron的作用：机器重启后，会侦测在关机的这一段时间期间，有没有 cron 没有执行的计划任务，如果有，开机后，anacron 会立即执行一下没有执行的任务。

相关博客：<http://c.biancheng.net/view/1095.html>

# Linux网络管理

## Linux下的端口的分配

TCP和UDP采用16为的端口号来识别应用程序。2^16 =65536,最大是65536.

TCP/IP的临时分配1024-5000之间的端口号

大于5000以上的是为其他服务保留的

## TCP 端口分配

21 ftp 文件传输服务  
22 ssh 安全远程连接服务  
23 telnet 远程连援服务，  
25 smtp 电子邮件服务

53 DNS域名解析服务  
80 http web 服务  
443 https安全web 服务

## UDP端口分配

69 tfp 简单文件传输协议

123 ntp时间同步服务（云主机）

161 snmp简单网络管理

如果你不知道服务对应的端口号，编辑 /etc/services，文件 包含所有的端口号  
例如：查看 ftp 的端口号。

grep ftp /etc/services  
说明:有的服务是 UDP 和TCP端口都会监听的。

## 查看端口的监听状态

语法：netstat 选项

参数：-a, --all.  
-n, --numeric  
-P, --programs. 看端口的监听状态  
-t 显示tcp连接

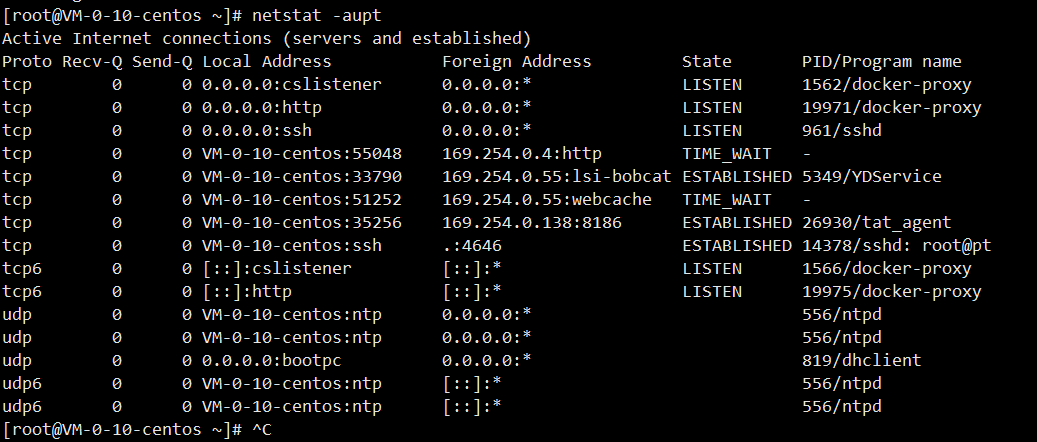
-u 显示udp连接

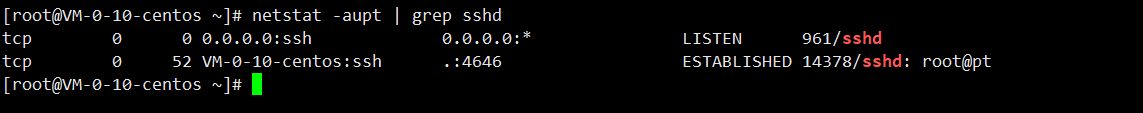
常用选项：tcp端口：anpt

udp端口：anpu

全部：anput

实例：netstat -antup





## 配置网络和IP地址

方式1：

nmtui

方式2：

vim /etc/sysconfig/network-scripts/ifcfg-\*\*\*

BOOTPROTO=dhcp

DEVICE=eth0

HWADDR=52:54:00:d2:26:ea

NM\_CONTROLLED=no

ONBOOT=yes

PERSISTENT\_DHCLIENT=yes

TYPE=Ethernet

USERCTL=no



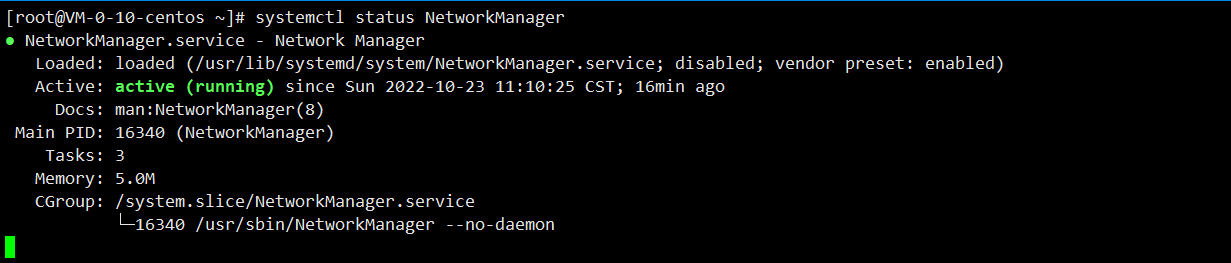
BOOTPROTO=dhcp 动态IP

BOOTPROTO=static 静态

BOOTPROTO=none 无（不指定）

说明：在RHEL7 中增强了NetworkManager 服务功能，弱化了 network 的功能…  
RHEL7中要确定 NetworkManager服务是开启的状态。此服务不开启，则无法通过nmtui工具配置网络。

systemctl status NetworkManager



 systemctl restart NetworkManager#重启服务

systemctl enable NetworkManager#设置服务开机自动启动。

或 systemctl restart netwok

## 修改主机名

RHEL7修改主机配置文件

vim /etc/hostname

如果是RHEL6操作系统，通过下面的文件修改主机名

vim /etc/sysconfig/network

说明：修改完配置文件，想永久生效，需要重启机器。

临时配置主机名

[root@VM-0-10-centos ~]# hostname wangtao

[root@VM-0-10-centos ~]# hostname

wangtao

[root@VM-0-10-centos ~]#

## Hosts文件

IP-->主机名的对应

[root@VM-0-10-centos ~]# vim /etc/hosts

127.0.0.1 VM-0-10-centos VM-0-10-centos

127.0.0.1 localhost.localdomain localhost

127.0.0.1 localhost4.localdomain4 localhost4

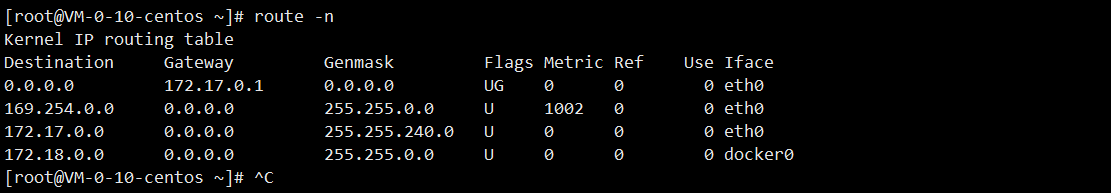
::1 VM-0-10-centos VM-0-10-centos

::1 localhost.localdomain localhost

::1 localhost6.localdomain6 localhost6

## 查看路由信息：查看默认网关

1. 通过配置文件查看
2. 通过route -n 命令



## ping

语法：ping 地址或域名

参数：-c 数目 在发送指定数目的包后停止。  
-i 秒数 设定间隔几秒送一个网络封包给一台机器，预设值是一秒送一次。

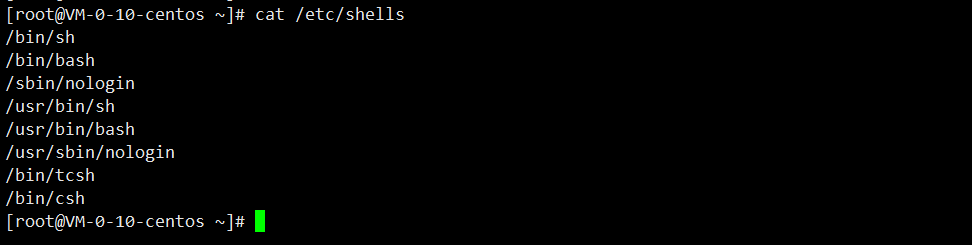
# Shell脚本

随看企业的应用越来越多，服务器的自动化管理也是越来越重要，在linux 中提供了一个非常强大的功能这种功能名字为 shell 脚本。

## shell 脚本概述

我们所输入的命令计算机是不识别的，这时就需要一种程序帮助我们进行翻译，变成计算机所识别的二进制程序，同时又将计算机所执行的结果翻译给我们，这个就是shell（壳)。

linux 的shell 脚本是一种特殊的应用程序，常见的shell 解释器有很多种使用不同的shell 脚本内部指令命令提示方式方面会存在一些区别，可以通过/etc/shells 文件查看。



/bin/bash 是大多数liunux中默认的shell解释器。

## 编写一个shell脚本

Linux不以后缀区分文件，为了方便管理一般以.sh结尾。

[root@VM-0-10-centos ~]*# vim hello.sh*

#! /bin/bash

*# 这是一个shell脚本*

echo "hello world"

**编写shell脚本的过程**

**编写shell脚本—>赋予执行权限-🡪执行**

[root@VM-0-10-centos ~]*# vim hello.sh #1.编写脚本*

#! /bin/bash

*# 这是一个shell脚本*

echo "hello world"

[root@VM-0-10-centos ~]*# chmod +x hello.sh #2.赋予执行权限*

[root@VM-0-10-centos ~]*# ./hello.sh #3.执行*

hello world

**编写Shell脚本注意事项**

1．#！/bin/bash 主要是为了声明，以下内容均为 bash 语言；

2.第二行为注释行，注释信息不生效，#后面表示注释；

3.当写一个较大的脚本时，如果没有一个好的注释，就没有人能看懂其中的意思了。

4.再往后即执行的命令

## 执行脚本的不同方式

弟一种便用绝对路径执行。

[root@VM-0-10-centos ~]*# /root/hello.sh*

第二种使用相对路径执行，如-/的方式

[root@VM-0-10-centos ~]*# ./hello.sh*

第三种使用 sh 命令来执行

语法：sh /路径/脚本文件名.sh

说明：不需要执行权限

[root@VM-0-10-centos ~]*# sh /root/hello.sh*

第四种便用 bash 命令来执行

格式：bash /路径/脚本文件名.sh

说明：不需要执行权限

[root@VM-0-10-centos ~]*# bash /root/hello.sh*

## Shell变量

变量：可以存放一个可变的值的空间。

常见的 shell 变量有自定义变量、坏境变量、位置变量、预足义变量。

一般使用 echo 输出变量的值,变量名的语法：$变量名

[root@VM-0-10-centos ~]*# echo $Linux*

**自定义变量**

自定义变量是用户根据目己的环境目己定义的变量；

Bash 中比较简单的变量，不用进行提前声明，而是直接指定变量名称并赋给初始值；

自定义变量的基本格式为变量名=变量值等号两遍不允许出现空格;

变量名称只能以字母和下划线开头，名称中不能包含＋、-、\*、/、?、%......等一些特殊字符。

变量名区分大小写

示例：1.定义一个变量名为Linux值为7.2(注意大小写)

[root@VM-0-10-centos ~]*# Linux=7.2*

[root@VM-0-10-centos ~]*# echo $Linux*

7.2

[root@VM-0-10-centos ~]*# linux=6.5*

[root@VM-0-10-centos ~]*# echo $linux*

6.5

[root@VM-0-10-centos ~]*# echo $Linux $linux*

7.2 6.5

[root@VM-0-10-centos ~]*#*

说明：1.可以直接在命令行中定义变量并赋值，通过echo进行输出变量，$是引用变量的特殊字符(必须使用$符号)

2.变量区分大小写

3.echo和调用的变量之间必须有空格隔开；

4.可以同时调用多组变量，如果有多个通过空格隔开；

5.当变量和后面的字符容易混淆的时候应该使用{}将变量名包起来；

[root@VM-0-10-centos ~]*# echo $linuxsystem*

[root@VM-0-10-centos ~]*# echo ${Linux}system*

7.2system

[root@VM-0-10-centos ~]*#*

## Read命令

除了上面的赋值之外还可以使用read 命令进行赋值，read 命令用来提示用户输入信息，从而实现简单的交互式过程（其实我们所输入的命令就是一种交互式的过程）。

执行时需要从标准输入设备键盘读取一行，并以空格为分隔符,比如同时定义两个变量操作如下所示

[root@VM-0-10-centos ~]*# read a b*

1 2

[root@VM-0-10-centos ~]*# echo $a*

1

[root@VM-0-10-centos ~]*# echo $a $b*

1 2

[root@VM-0-10-centos ~]*#*

为了交互式更加形象，提高易读性，加上-p选项来设置提示提示信息

[root@VM-0-10-centos ~]*# read -p "input your passwd:" passwd*

input your passwd:123456

[root@VM-0-10-centos ~]*# echo $passwd*

123456

[root@VM-0-10-centos ~]*#*

## 数值变量的运算

shell 脚本的数值运算多用于脚本程序的过程控制（如循环次数，使用量比较等）。  
在shell 环境中，只能进行比较简单的整数运算。

运算符与变量之间必须有一个空格，整数的运算主要是通过内部命令 expr 命令进行运算，  
语法： 变量1运算符 变量2

说明：其中 变量1、变量2 ……对应的需要计算的数值变量（需要＄符号调用）

常用的几种运算符如下所示

加法运算：+

减法运算：-

乘法运算:\\*

除法运算：/

求模（取余）运算：%



将运算结果赋值给其他变量

[root@VM-0-10-centos ~]*# a=1*

[root@VM-0-10-centos ~]*# b=8*

[root@VM-0-10-centos ~]*# sum=$(expr $a + $b)*

[root@VM-0-10-centos ~]*# echo $sum*

9

[root@VM-0-10-centos ~]*#*