

# 第一章 linux操作系统

## 一、简介

Linux 是一套免费使用和自由传播的类 Unix 操作系统，是一个多用户、多任务、支持多线程和多 CPU 的操作系统。

既然Linux也是一个操作系统，那大家肯定会下意识将它和Windows系统做比较，两者当然也是有优有劣。相对于Windows来说，Linux是完全免费的，开放源码，为用户提供了最大限度的自由度。且Linux更加稳定更加安全，所以通常作为服务器的操作系统使用。

Linux是一个诞生于网络、成长于网络且成熟于网络的奇特的操作系统。1991年10月，由芬兰大学生 Linus Torvalds带头编写而成，其后一大批知名的、不知名的电脑黑客、编程人员加入到开发过程中来，Linux逐渐成长起来。

### Linux的主要特点

- **完全免费**：Linux是一款免费的操作系统，用户可以通过网络或其他途径免费获得，并可以任意修改其源代码。
- **安全性**：Linux采取了许多安全技术措施，其中有对读、写进行权限控制、审计跟踪、核心授权等技术，这些都为安全提供了保障。
- **多用户**：操作系统资源可以被不同用户使用，每个用户对自己的资源有特定的权限，互不影响。
- **多任务**：计算机同时执行多个程序，而同时各个程序的运行互相独立。
- **独立性**：Linux是具有设备独立性的操作系统，内核具有高度适应能力。
- **可移植性**：Linux是一种可移植的操作系统，能够在从微型计算机到大型计算机的任何环境中在任何平台上运行。

内核：kernel

### 常见的操作系统

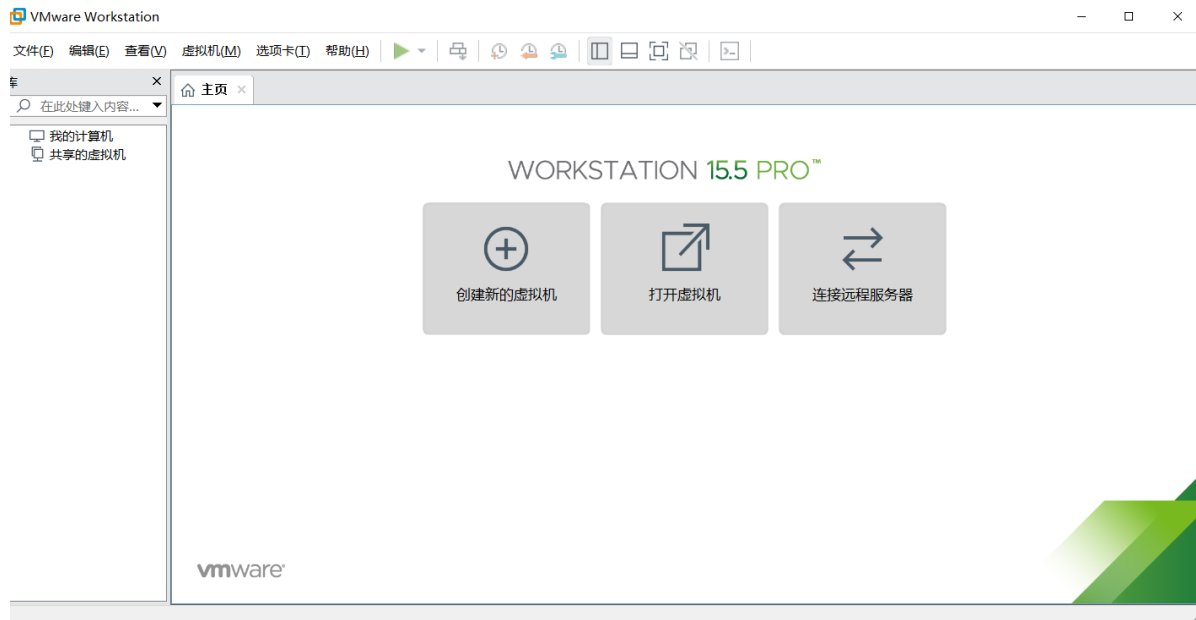
- **Windows**：由微软公司成功开发，是一个多任务的操作系统，它采用图形窗口界面，用户对计算机的各种复杂操作只需要通过点击鼠标就可以实现。
- **UNIX**：是一个强大的多用户、多任务的操作系统，支持多种处理器架构，按照操作系统的分类，属于分时系统。Linux的思想源于Unix。
- **Mac OS**：是一套运行于苹果Macintosh系列电脑上的操作系统。从本质上讲，Mac OS 也是UNIX的一个变体。
- **Android**：是一种基于Linux的自由及开放源代码的操作系统，主要用于移动设备，如智能手机和平板电脑，由Google公司和开放手机联盟领导及开发。
- **iOS**：是苹果公司开发的移动端的操作系统。iOS与苹果的MacOS操作系统一样，属于类Unix的商业操作系统。

## 二、安装vmware

直接腾讯软件中心下载 <https://pc.qq.com/search.html#!keyword=vmware>

一直下一步，选一个目录安装即可

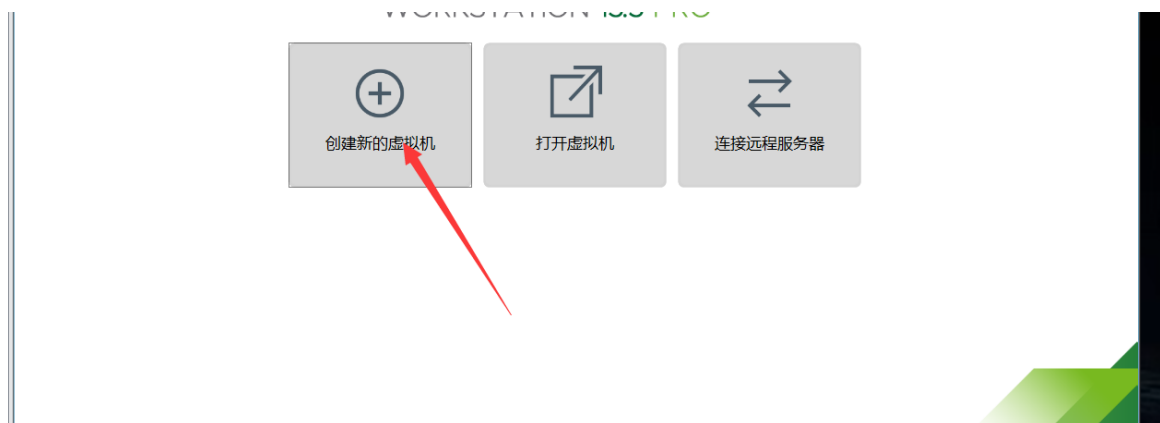
- 1 官网下载后用这个秘钥
- 2 YG5H2-ANZ0H-M8ERY-TXZZZ-YKRV8
- 3 UG5J2-0ME12-M89WY-NPWXX-WQH88
- 4 UA5DR-2ZD4H-089FY-6YQ5T-YPRX6
- 5 GA590-86Y05-4806Y-X4PEE-ZV8E0
- 6 ZF582-0NW5N-H8D2P-0XZEE-Z22VA
- 7 YA18K-0WY8P-H85DY-L4NZG-X7RAD



### 三、安装linux

<https://developer.aliyun.com/mirror/>

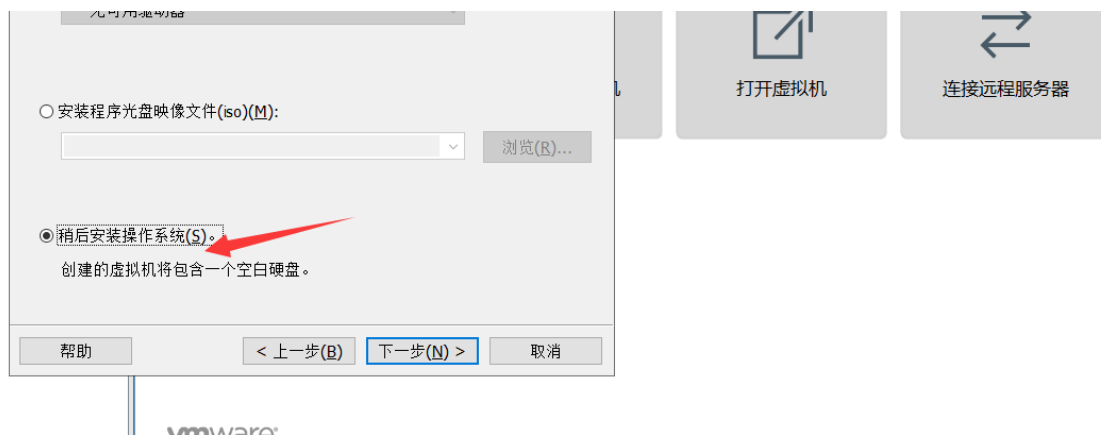
第一步：



第二步：

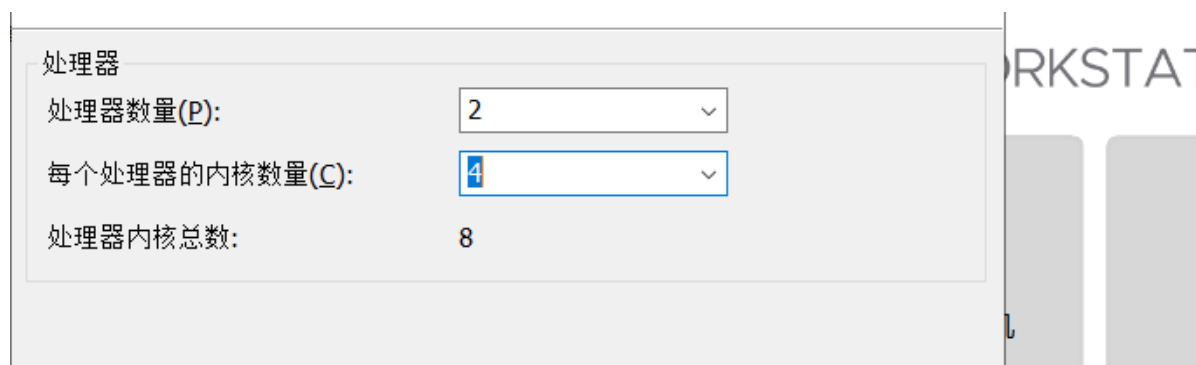
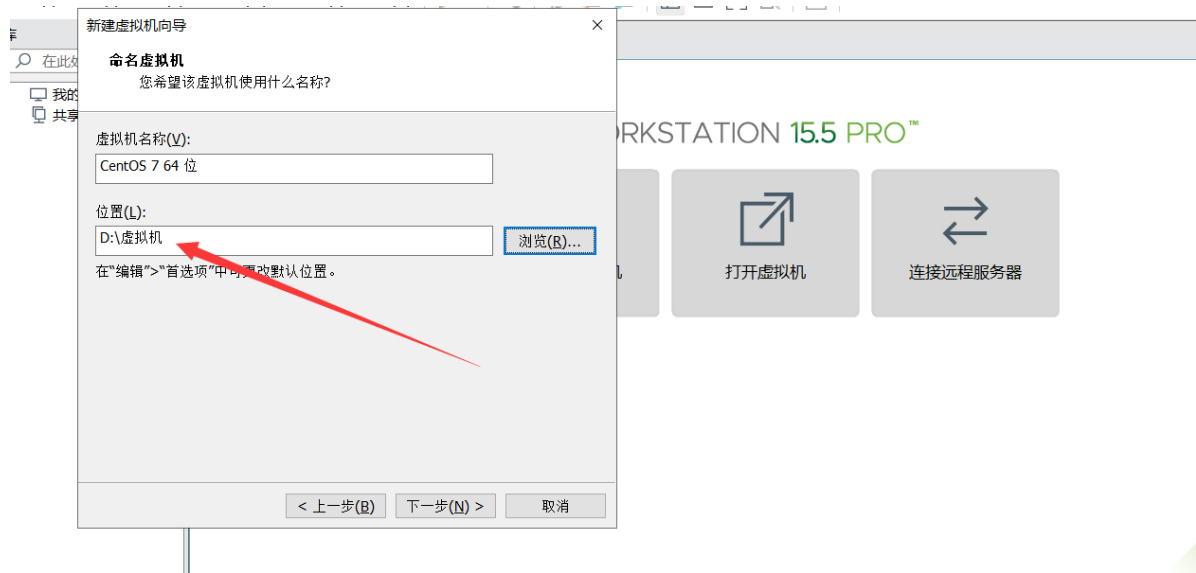


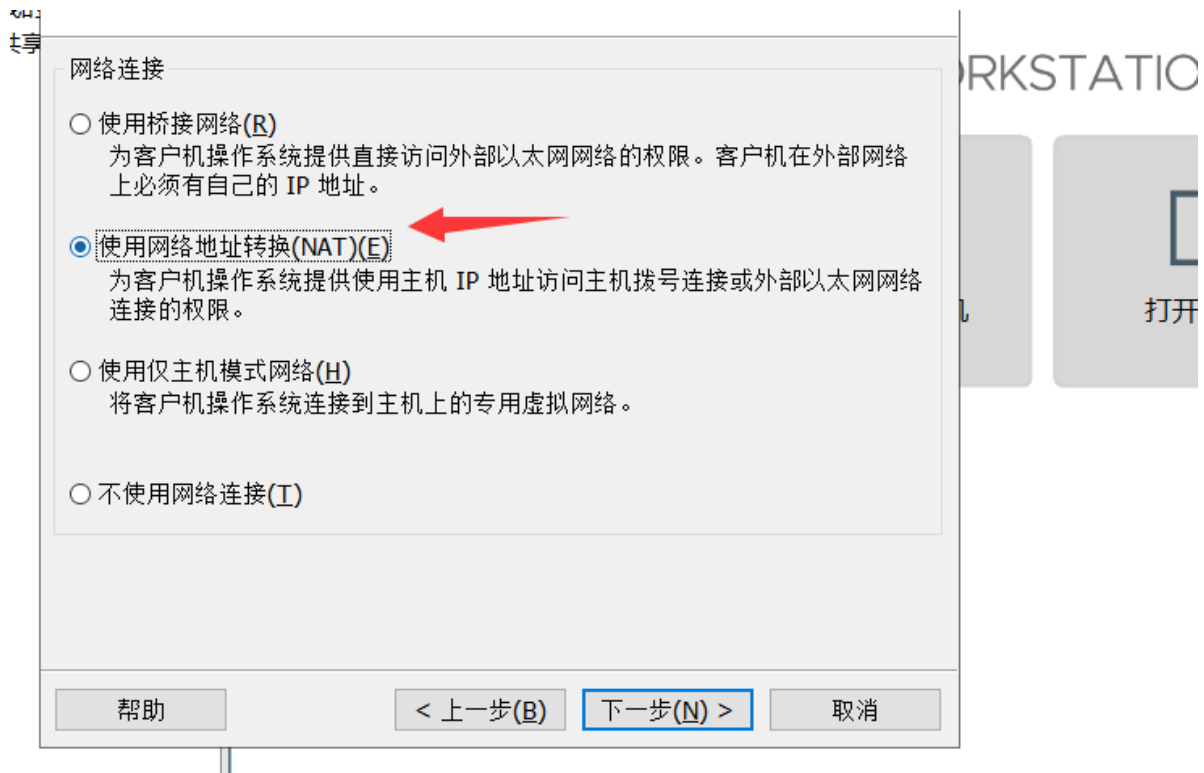
下一步



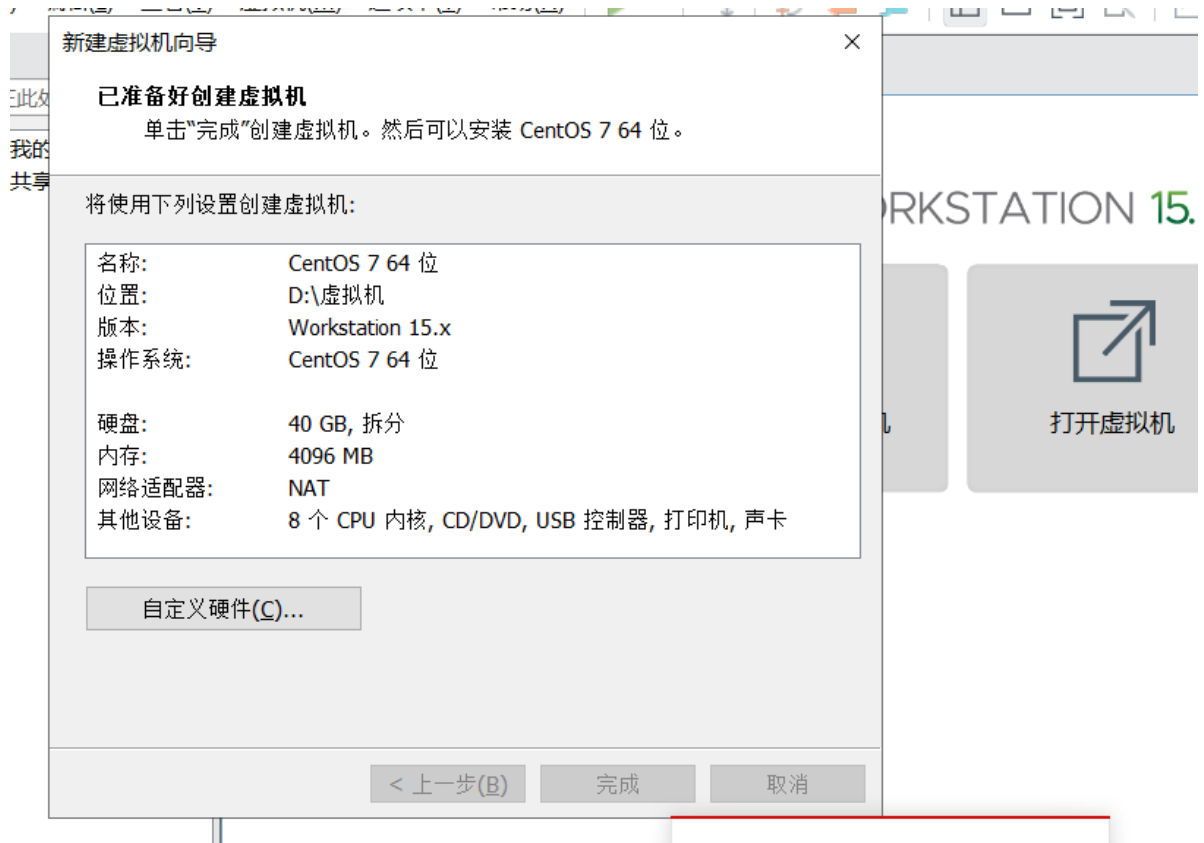
下一步:







看不懂就下一步：



关键：挂载系统



下一步



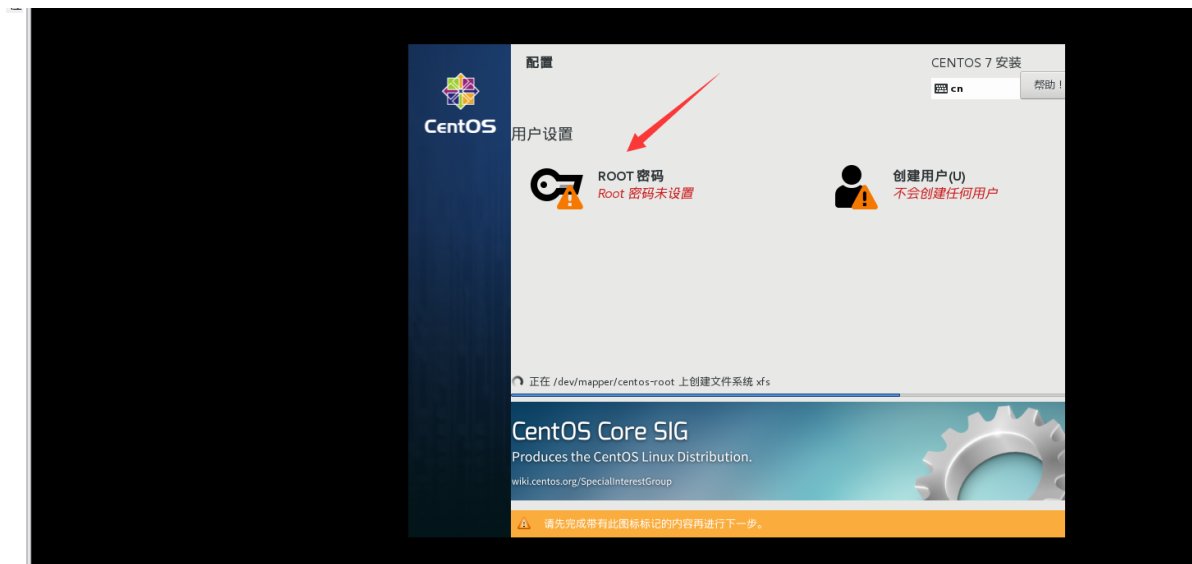
下一步



下一步



下一步



设置一个root用户的密码



有人有这个问题





此主机支持 Intel VT-x，但 Intel VT-x 处于禁用状态。

如果已在 BIOS/固件设置中禁用 Intel VT-x，或主机自更改此设置后从未重新启动，则 Intel VT-x 可能被禁用。

(1) 确认 BIOS/固件设置中启用了 Intel VT-x 并禁用了“可信执行”。

(2) 如果这两项 BIOS/固件设置有一项已更改，请重新启动主机。

(3) 如果您在安装 VMware Workstation 之后从未重新启动主机，请重新启动。

(4) 将主机的 BIOS/固件更新至最新版本。

此主机不支持“Intel EPT”硬件辅助的 MMU 虚拟化。

模块“CPUIDEarly”启动失败。

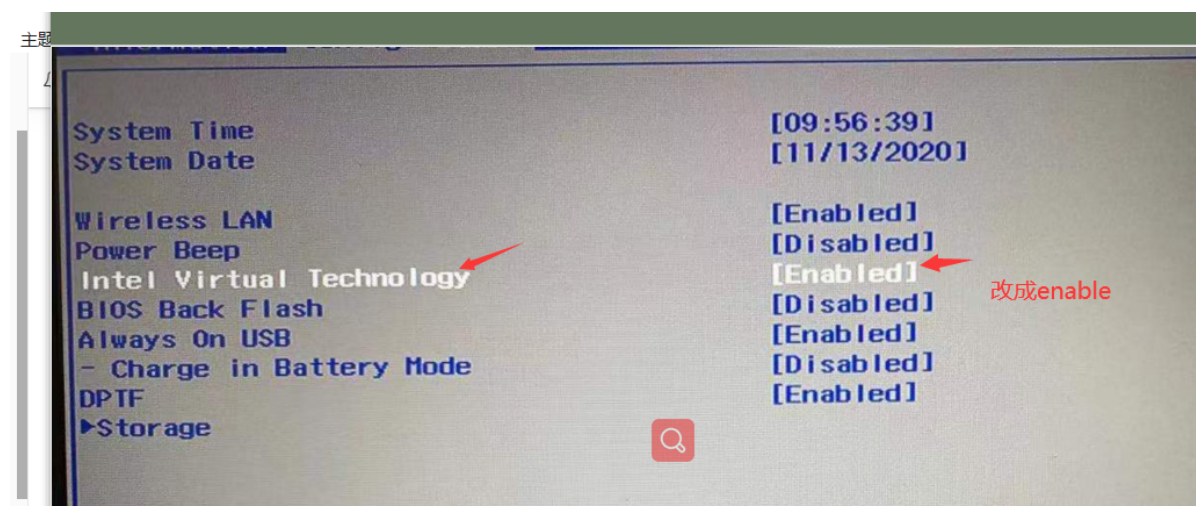
未能启动虚拟机。

确定

解决方法：

<https://jingyan.baidu.com/article/fc07f98976710e12ffe519de.html>

找到此选项，改成enable



## 四、安装finalshell

<http://www.hostbuf.com/>

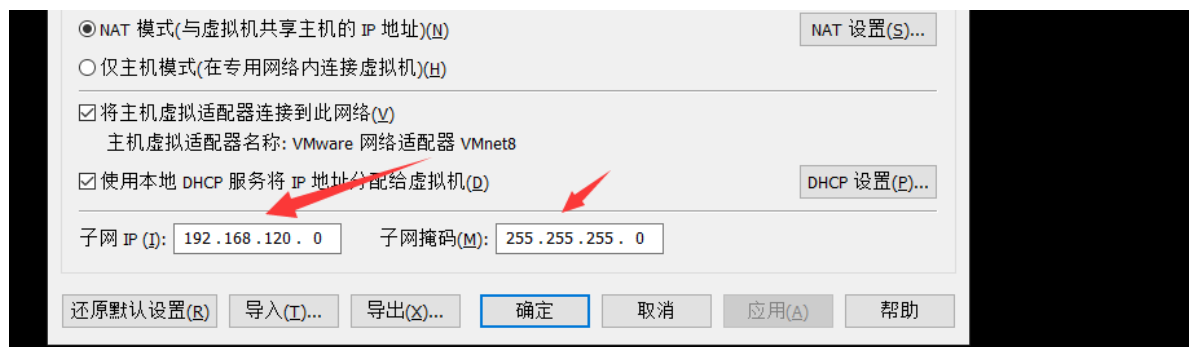
看见哪个点就行了。下一步 next install

我们需要IP地址和端口链接，本机的IP（127.0.0.1）localhost。

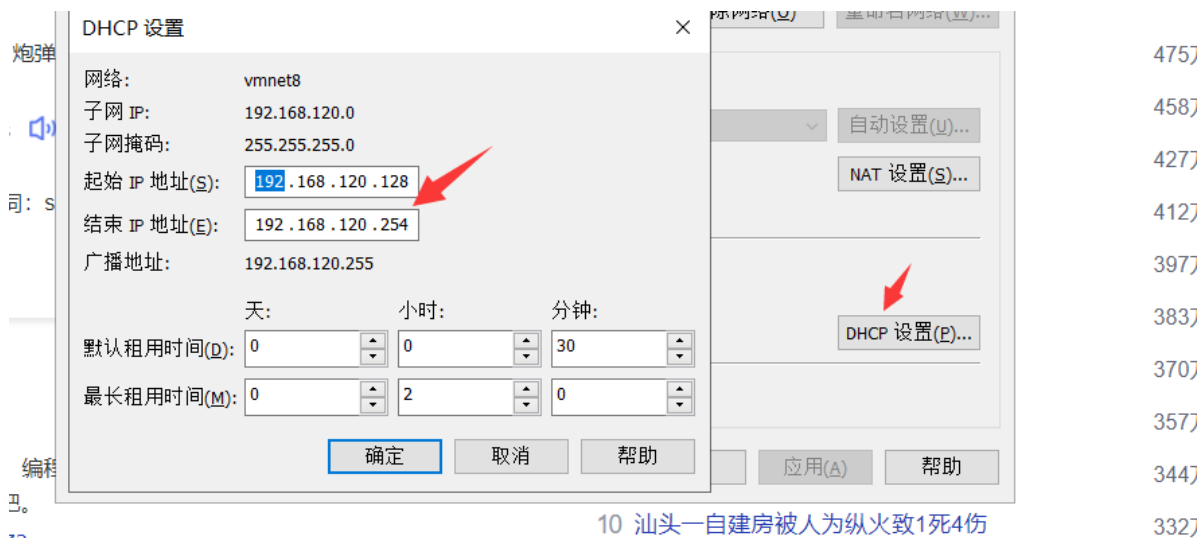
就必须使用虚拟机和主机建立一个局域网，能够互相连接上。

nat，创建局域网

### 查看网络号和子网掩码



### 查看ip范围



10 汕头一自建房被人为纵火致1死4伤

### 查看网关号



### ip addr 查看IP地址

```
1 | cd /etc/sysconfig/network-scripts/  
2 | vi ifcfg-ens33
```

点击i，进入编辑模式，左下角出现insert

```
1 | # 修改以下内容  
2 | BOOTPROTO=static  
3 | onboot=yes  
4 | # 增加以下内容  
5 | IPADDR=192.168.2.190  
6 | NETMASK=255.255.255.0  
7 | GATEWAY=192.168.2.1
```

先按`esc`退出编辑模式，再按`shfit+:`进入命令模式，再输入`wq`，点击回车保存并退出。

## 五、linux目录结构

### 1.基本介绍：

linux的文件系统是采用级层式的树状目录结构，在此结构中最上层是根目录“/”，然后在此目录下再创建其它的目录。

在linux世界里。一切皆文件

- 结构图

### 2.目录结构具体介绍：

- /
- root, 存放root用户的相关文件
- home, 存放普通用户的相关文件
- bin, 存放常用命令的目录，如vi, su
- sbin, 要具有一定权限才可以使用命令
- mnt, 默认挂载光驱和软驱的目录
- etc, 存放配置的相关文件
- var, 存放经常变化的文件，如网络连接的sock文件、日志
- boot, 存放引导系统启动的相关文件
- usr, 安装一个软件的默认目录，相当于windows下的program files
- proc, 这个目录是一个虚拟的目录，它是系统内存的映射，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
- tmp, 这个目录是用来存放一些临时文件的
- del, 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，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面的内容了。
- opt，这是给主机额外安装软件所摆放，如安装oracle数据库就可放到该目录下默认为空。
- selinux，selinux是一种安全子系统，它能控制程序只能访问特定文件。

### 3.linux目录总结

- 1) linux的目录中有且只有一个根目录/，
- 2) linux的各个目录存放的内容是规划好的，不用乱放文件，
- 3) linux是以文件的形式管理我们的设备，因此linux系统。**一切皆文件。**
- 4) linux的各个文件目录下存放什么内容，大家必须有一个认识，
- 5) 你的脑海里应该有一颗linux的目录树。

相对路径： 相对当前的路径      我家隔壁是老王    相对于的是我家

绝对路径： 一个绝对的路径      中国太原市恒大华府1-1-2304家是老王

## 第二章 linux命令

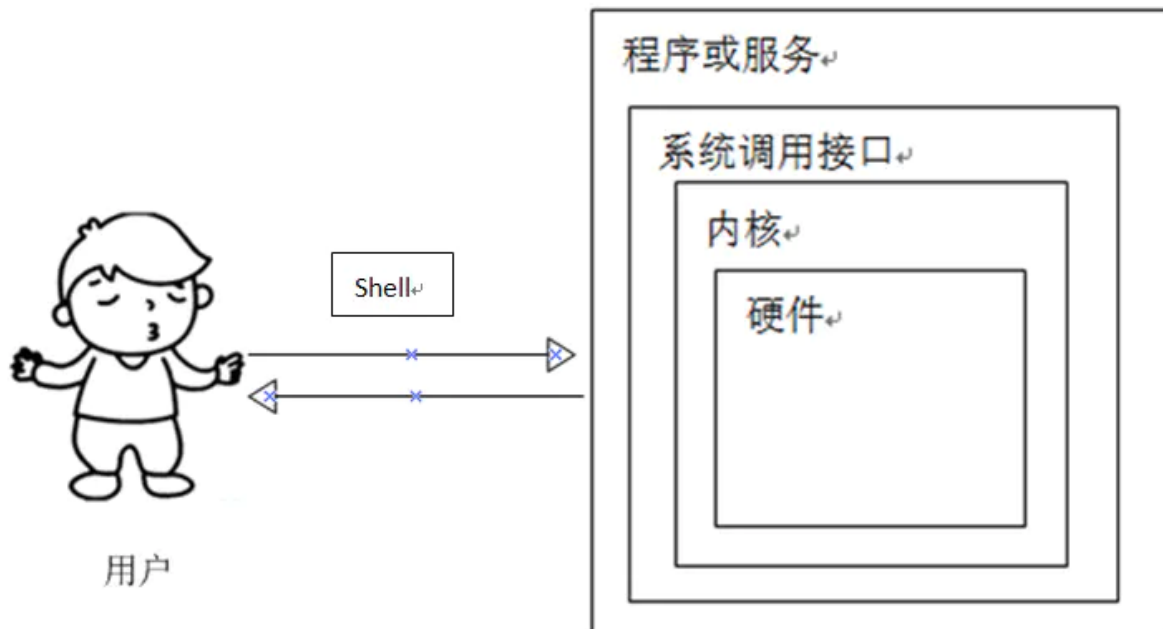
我们操作通常有2种方式：第一种就是通过桌面图形化操作；另外一种则是通过终端输入命令。

针对服务器而言我们通常是采用远程连接工具连接到服务器，然后以**命令**的方式进行管理。我们输入命令后计算机怎么就知道该做什么呢？那就得靠命令行工具（也称终端）进行翻译解释，然后调用相应的接口服务完成特定的功能。

用户与linux系统的交互是通过**shell终端**，下文讲先分析shell命令行工具，再讲解最基础的linux命令。

### 一、shell简介

计算机硬件是由**运算器、控制器、存储器、输入/输出**设备等共同组成的，而让各种硬件设备各司其职且又能协同运行的东西就是系统内核。Linux 系统的内核负责完成对硬件资源的分配、调度等管理任务。因此，系统内核对计算机的正常运行至关重要，一般不建议直接去编辑内核中的参数，而是让用户通过基于系统调用接口开发出的程序或服务来管理计算机，以满足日常工作的需要。而shell就是这样一个客户端命令行工具，充当用户与系统内核沟通的桥梁。



file

Shell也称“终端”、“壳”，是一种用c语言写的命令行工具，充当人与内核（硬件）之间的翻译官，用户把一些命令“告诉”终端，它就会调用相应的程序服务去完成某些工作。

## 1.1 shell版本

以下列举几种shell版本

- sh (Bourne Shell)：是UNIX最初使用的 shell，而且在每种 UNIX 上都可以使用。在shell编程方面做的很优秀，但是在处理与用户的交互方面做得不如其他几种shell。
- **\*bash\*** (Bourne Again Shell)：Linux默认，是Bourne Shell的扩展。完全兼容Bourne Shell，并在他基础上增加了很多特性如命令补全、命令历史等。它还包含了很多C Shell和Korn Shell中的优点，有灵活和强大的编辑接口，同时有很友好的用户界面。
- csh (C Shell)：一种比Bourne Shell更适合的变种Shell，语法与C语言很相似。
- Tcsh: Linux提供的C Shell的一个扩展版本。包括命令行编辑，可编程单词补全，拼写校正，历史命令替换，作业控制和类似 C 语言的语法，他不仅和 Bash Shell 提示符兼容，而且还提供比 Bash Shell 更多的提示符参数。
- ksh (Korn Shell)：集合了 C Shell 和 Bourne Shell 的优点并且和 Bourne Shell 完全兼容。
- pdksh: Linux 系统提供的 ksh 的扩展。支持人物控制，可以在命令行上挂起，后台执行，唤醒或终止程序。

现在包括红帽系统在内的许多主流 Linux 系统默认使用的终端是 **\*Bash\*** (Bourne-Again SHell) 解释器。主流 Linux 系统选择 Bash 解释器作为命令行终端主要有以下 4 项优势：

通过上下方向键来调取过往执行过的 Linux 命令；

命令或参数仅需输入前几位就可以用 Tab 键补全；

具有强大的批处理脚本；

具有实用的环境变量功能。

我们装好的centos7默认也是采用的bash终端，当采用远程连接工具连接到centos7，默认开启的终端就是bash，可以通过在终端中输入命令：echo \$SHELL 进行查看，得到的结果为：/bin/bash。

- 如何判断当前终端的类型？

通过"echo \$SHELL" 命令或通过"evn"终端命令

- 如何更改登录的终端类型？

通过终端命令chsh，如：chsh -s /bin/bash将登录终端改为bash

## 1.2 shell命令类型

shell是一个程序，用于用户和操作系统进行交互，相当于是一个命令解析器。

作为用户和内核交互的接口，既是一种命令语言又是一种程序设计语言。shell命令又分为内置命令和外部命令。

- 内置命令

内部命令实际上是shell程序的一部分，其中包含的是一些比较简单的linux系统命令，这些命令由shell程序识别并在shell程序内部完成运行，通常**\*在linux系统加载运行时shell就被加载并驻留在系统内存中\***。内部命令是写在bashy源码里面的，其执行速度比外部命令快，因为解析内部命令shell不需要创建子进程。比如：exit, history, cd, echo等。

- 外部命令

外部命令是linux系统中的实用程序部分，因为实用程序的功能通常都比较强大，所以其包含的程序量也会很大，**\*在系统加载时并不随系统一起被加载到内存中，而是在需要时才将其调用内存\***。通常外部命令的实体并**\*不包含在shell中，但是其命令执行过程是由shell程序控制的\***。shell程序管理外部命令执行的路径查找、加载存放，并控制命令的执行。外部命令是在bash之外额外安装的，通常放在/bin, /usr/bin, /sbin, /usr/sbin.....等等。可通过“echo \$PATH”命令查看外部命令的存储路径，比如：ls、vi等

如何区分是内部命令还是外部命令？

- 通过type命令，如：type cd 可以看到是内部命令；type mkdir可以看到是外部命令。
- 通过man命令，如：man cd, 会提示是bash的内置命令；man ls, 会提示是外部命令。

内部命令和外部命令最大的区别之处就是性能。内部命令由于构建在shell中而不必创建多余的进程，要比外部命令执行快得多。因此和执行更大的脚本道理一样，执行包含很多外部命令的脚本会损害脚本的性能。

## 二、最常用命令

既然 Linux 系统中已经有了 Bash 这么好用的“翻译官”，那么接下来就有必要好好学习怎么跟它沟通了。要想准确、高效地完成各种任务，仅依赖于命令本身是不够的，还应该根据实际情况来灵活调整各种命令的参数。

linux执行命令一般格式：**\*命令名称 [命令参数] [命令对象]\***

命令名称、命令参数、命令对象之间用空格键分隔。命令对象一般是指要处理的文件、目录、用户等资源，命令参数则可以用长格式(-- )或短格式(-)。

长格式：man --help

短格式：man -h

linux命令很多，最重要的是各种参数的搭配使用，新手记住所有参数是不可能的，需要经常通过帮助命令进行查阅。

特别说明：在终端中，有时候显示的内容很多，可以通过快捷键进行翻看。

shift+PgUp：向上翻页

shift+PgDn：向下翻页



## 2.1 帮助命令

帮助命令主要有2个：man、help。平时要多于帮助命令，遇到不清楚的命令就用帮助命令查看，无须死记硬背，熟能生巧。

### 2.1.1 man命令

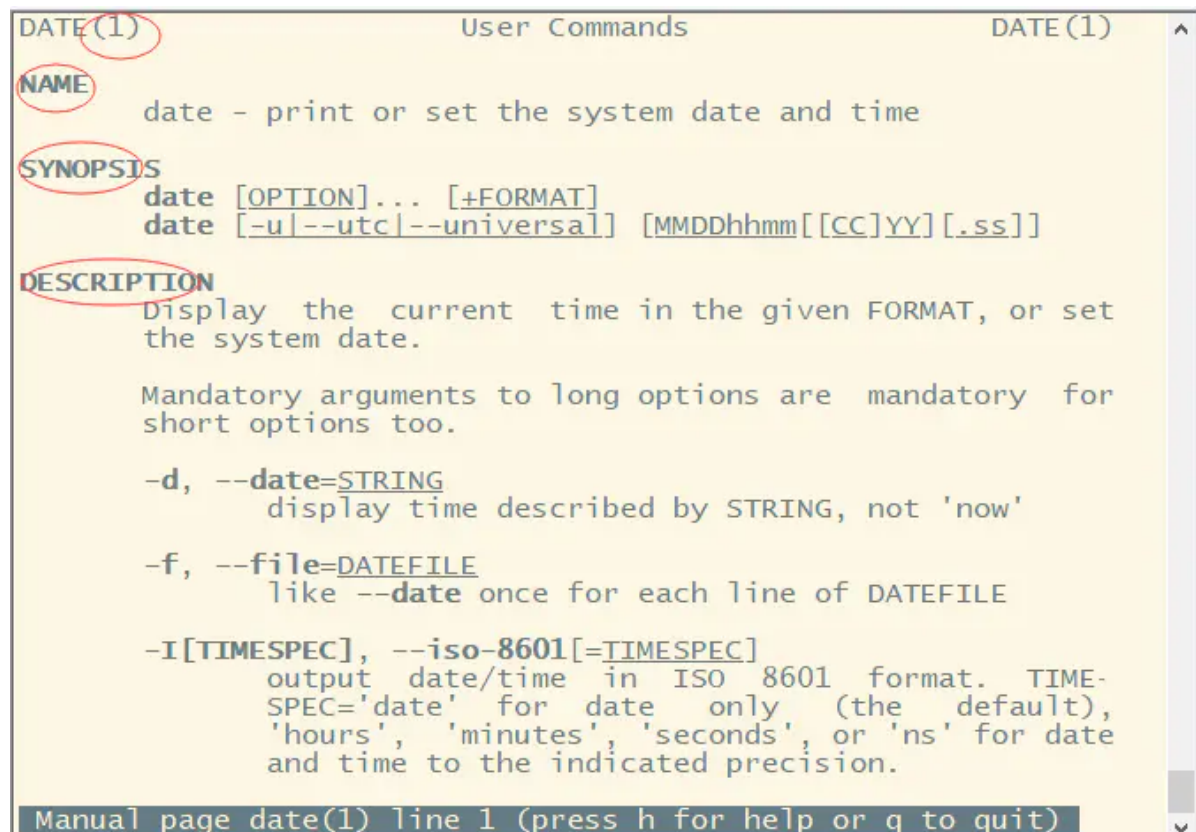
语法：man [命令或配置文件]

作用：获取帮助信息

案例：

```
1 [root@heimatengyun ~]# man date
```

比如欲查看date命令的用法，输入命令man date之后出现下图信息



```
DATE(1)                                User Commands                                DATE(1)
NAME
date - print or set the system date and time

SYNOPSIS
date [OPTION]... [+FORMAT]
date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

DESCRIPTION
Display the current time in the given FORMAT, or set
the system date.

Mandatory arguments to long options are mandatory for
short options too.

-d, --date=STRING
    display time described by STRING, not 'now'

-f, --file=DATEFILE
    like --date once for each line of DATEFILE

-I[TIMESPEC], --iso-8601[=TIMESPEC]
    output date/time in ISO 8601 format. TIME-
    SPEC='date' for date only (the default),
    'hours', 'minutes', 'seconds', or 'ns' for date
    and time to the indicated precision.

Manual page date(1) line 1 (press h for help or q to quit)
```

file

由于帮助信息比较多，需要翻页才能看完，因此先说明一下本界面可能需要用到按键及用途。

按键	用途
空格键	向下翻页
PgDn (page down)	向下翻页
PgUp (page up)	向上翻页
home	前往首页
end	前往页尾
/	从上至下搜索关键字，如“/linux”
?	从下至上搜索关键字，如“?linux”
n	定位到下一个搜索到的关键词
N	定位到上一个搜索到的关键词
q	退出帮助文档

初学者一看，这么多的帮助信息反而感到困惑了，其实很简单的，我们来理一下帮助信息的结构（上图中圈红线的部分）。

结构名称	说明
NAME	命令的名称
SYNOPSIS	命令的基本语法格式
DESCRIPTION	详细说明语法格式对应的选项和参数的用法
EXAMPLES	如何使用命令的示例
OVERVIEW	概述
DEFAULTS	默认功能
OPTIONS	具体的可用选项
ENVIRONMENT	环境变量
SEE ALSO	相关的资料，通常是手册页

通过这些帮助信息，就可以轻松的掌握各个命令的用法。

### 2.1.2 help命令

语法：help 命令

作用：获取shell内置命令的帮助信息

案例：

```
1 | [root@heimatengyun ~]# help cd
```

在正式进入命令学习之前，再介绍几个\*linux常用的快捷键\*



快捷键	作用
ctrl+l	清屏
ctrl+q	退出
ctrl+c	停止进程
上下键	查找曾今执行过的命令
tab	命令补全，多用不仅提高效率还能防止敲错

既然linux一切皆文件，了解万能的帮助命令后，我们就从目录文件命令开始学习。

## 2.2 目录相关命令

目录可以对文件进行分类管理，因此在讲解文件命令之前先讲解目录相关命令。

### 2.2.1 pwd

语法：pwd

功能描述：显示当前工作目录的绝对路径

案例：

```
1 [root@heimatengyun ~]# pwd
2 /root
```

### 2.2.2 ls

语法：ls [选项] [目录或文件]

功能描述：列出目录下的文件

选项：

-a(--all):显示全部文件，包括隐藏文件（以.开头的文件）

```
1 -d: directory, 列出目录本身
```

-l: long长数据串列出，包含文件的属性和权限等。每行列出的信息依次为：**\*文件类型与权限、链接数、文件属性、文件属组、文件大小（byte）、创建或最近修改时间、名字\***

案例：

```
1 [root@heimatengyun test]# ls -al
2 total 12
3 drwxr-xr-x.  2 root root   37 Nov 24 10:43 .
4 dr-xr-x---. 16 root root 4096 Nov 24 00:02 ..
5 -rw-r--r--.  1 root root   26 Nov 23 20:40 test1.txt
6 -rw-r--r--.  1 root root   66 Nov 23 21:56 test.txt
```

### 2.2.3 cd

语法: `cd [选项][目录名称]`

功能描述: 切换到指定目录

案例:

`cd` 绝对路径或相对路径 (跳转到指定目录)

`cd` 或 `cd ~` (返回当前用户家目录)

`cd -` (返回上一次所在的目录)

`cd ..` (返回当前目录的上一级目录)

`../` 上一级目录

`./` 当前目录

`~` 用户的home家目录

### 2.2.4 mkdir

语法: `mkdir [选项] 目录名称`

功能描述: 创建指定目录

选项:

`-p`: parents, 创建多层目录

案例:

```
1 [root@heimatengyun ~]# mkdir test
2 [root@heimatengyun ~]# mkdir -p test1/test1
```

### 2.2.5 rmdir

语法: `rmdir [选项] 目录名称`

功能描述: 删除空目录

选项:

`-p`: 删除多级目录

案例:

```
1 [root@heimatengyun ~]# rmdir test
2 [root@heimatengyun ~]# rmdir -p test1/test1/ (如果不指定-p参数, 则只删除最后一级不为空的目录)
```

### 2.2.6 cp

语法: cp 源目录或文件 目标目录或文件

功能描述: 复制目录或文件

选项:

-r: recursive, 递归复制整个文件夹

案例:

将test复制为test.txt, 复制时重命名

```
1 [root@heimatengyun ~]# cp test test.txt
```

将所有test目录内容复制到test1目录 (自动创建test1目录)

```
1 [root@heimatengyun ~]# cp -r test test1
```

### 2.2.7 mv

语法: mv [选项] 源 目标

功能描述: 移动文件或重命名文件

案例:

将test文件重命名为test.txt

```
1 [root@heimatengyun test]# mv test test.txt
```

将test目录及其里边所有内容移动到test2目录, 自动创建test2目录 (可以理解为将test文件夹重命名为test2)

```
1 [root@heimatengyun ~]# mv test/ test2
```

### 2.2.8 rm

语法: rm [选项] 文件

功能描述: 删除文件及目录

选项:

-f: force强制执行

-r: recursive递归执行

案例:

递归删除目录及其下所有内容

```
1 | [root@heimatengyun ~]# rm -rf test2/
```

通配符

\*

rm -f \*

## 2.3 文件相关命令

linux一切皆文件，在连接目录相关命令后，本节学习文件相关命令。

### 2.3.1 touch

语法：touch [选项] 文件名

功能描述：touch

案例：

```
1 | [root@heimatengyun test]# touch test.txt
```

echo

语法：echo 字符串或变量

功能描述：输出字符串或变量值，还可以搭配从定向符将内容存储到文件

案例：

```
1 | [root@heimatengyun test]# echo hello
2 | hello
3 | [root@heimatengyun test]# echo $SHELL
4 | /bin/bash
5 | [root@heimatengyun test]# echo linux >> test.txt
```

### 2.3.2 cat

语法：cat [选项] 文件名

功能描述：查看文件内容，从第一行开始显示

选项：

-A：列出特殊字符而非空白

-b：列出行号，空白行不算行号

-n：列出行号，空白行也会有行号

-v：列出一些看不出来的特殊字符

案例：

```
1 [root@heimatengyun test]# cat test.txt
2 hello
3 who are you
4
5 where are you from
6 [root@heimatengyun test]# cat -A test.txt
7 hello$
8 who are you$
9 $
10 where are you from$
11 [root@heimatengyun test]# cat -b test.txt
12     1 hello
13     2 who are you
14
15     3 where are you from
16 [root@heimatengyun test]# cat -n test.txt
17     1 hello
18     2 who are you
19     3
20     4 where are you from
```

### 2.3.3 more

语法: more [选项] 文件

功能描述: 查看文件内容, 一页一页的显示

使用说明:

空格键 (space) : 向下翻一页

enter: 向下翻一行

q: 退出more, 不在显示文件内容

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号

### 2.3.4 head

语法: head [选项] 文件

功能描述: 查看文件内容, 只看头几行

选项:

-n: 查看头n行

案例:

```
[root@heimatengyun test]# head -n 2 test.txt
```

### 2.3.5 tail

语法: tail [选项] 文件

功能描述: 查看文件内容, 只查看文件末尾几行

选项:

-n: 末尾几行

-f: follow输出文件修改的内容, 用于追踪文件修改

案例:

```
[root@heimatengyun test]# tail -n 2 test.txt
```

### 2.3.6 wc

语法: wc [选项] 文本

功能描述: 统计指定文本的行数、字数、字节数

选项:

-l: lines显示行数

-w: 显示单词数

-c: 显示字节数

案例:

```
1 [root@heimatengyun test]# cat test.txt
2 hello
3
4 who are you
5 where are you form?
6
7 wellcome
8 hahah
9 test
10 [root@heimatengyun test]# wc -l test.txt
11 8 test.txt
12 [root@heimatengyun test]# wc -c test.txt
13 60 test.txt
14 [root@heimatengyun test]# wc -w test.txt
15 11 test.txt
```

### 2.3.7 stat

语法: stat [选项] 文件

功能描述: 查看文件的具体存储信息和时间等信息

案例:

```
1 [root@heimatengyun test]# stat test.txt
2   File: 'test.txt'
3   Size: 60          Blocks: 8          IO Block: 4096   regular file
4 Device: fd01h/64769d Inode: 2160373    Links: 1
5 Access: (0644/-rw-r--r--)  Uid: (  0/   root)   Gid: (  0/   root)
6 Context: unconfined_u:object_r:admin_home_t:s0
7 Access: 2019-11-23 18:27:02.153101936 +0800
8 Modify: 2019-11-23 18:26:56.254259870 +0800
9 Change: 2019-11-23 18:26:56.254259870 +0800
10 Birth: -
```

### 2.3.8 file

语法: file 文件名

功能描述: 查看文件类型

案例:

```
1 [root@heimatengyun test]# file test.txt
2 test.txt: ASCII text
```

### 2.3.9 wget

语法: wget [参数] [url地址]

功能: 下载网络文件

参数:

- b: background后台下载
- P: directory-prefix下载到指定目录
- t: tries 最大尝试次数
- c: continue断点续传clear
- p: page-requisites下载页面所有内容, 包括图片、视频等
- r: recursive递归下载

案例:

下载百度logo图片

```
1 [root@heimatengyun test]# wget https://www.baidu.com/img/bd_logo1.png
2 --2019-11-23 22:29:45-- https://www.baidu.com/img/bd_logo1.png
3 Resolving www.baidu.com (www.baidu.com)... 14.215.177.38, 14.215.177.39
4 Connecting to www.baidu.com (www.baidu.com)|14.215.177.38|:443... connected.
5 HTTP request sent, awaiting response... 200 OK
6 Length: 7877 (7.7K) [image/png]
7 Saving to: 'bd_logo1.png'
8
9 100%[=====>] 7,877      --.-K/s   in 0.002s
10
11 2019-11-23 22:29:45 (4.43 MB/s) - 'bd_logo1.png' saved [7877/7877]
```

## 2.4 查找命令

有时候需要从大量文件中找出需要的文件或者从指定文件中查找特定内容，这就需要用到查找相关的命令。

### 2.4.1 find

语法：find [搜索范围] [匹配条件]

功能描述：查找文件或目录

参数说明

-name：按文件名称查找

-user：按文件拥有者查找

-size：根据文件大小查找文件（+n大于，-n小于，n等于）

案例

在test目录下查找test1.txt文件

```
1 [root@heimatengyun ~]# find test/ -name test1.txt
2 test/test1.txt
```

查找test目录下查找用户root的文件

```
1 [root@heimatengyun ~]# find test/ -user root
2 test/
3 test/test.txt
```

在test目录下查找小于100M的文件

```
1 [root@heimatengyun ~]# find test/ -size -102400
2 test/
3 test/test.txt
```



### 2.4.2 grep

语法: grep [参数] 查找内容 源文件

功能描述: 在文件内搜索字符串匹配的行并输出

参数:

-c: count只输出匹配行的计数

-n: line-number

案例:

```
1 [root@heimatengyun test]# grep -n who test.txt
2 3:who are you
3 [root@heimatengyun test]# grep -c who test.txt
4 1
```

### 2.4.3 which

语法: which [选项] 命令

功能描述: 搜索命令所在目录及别名信息

案例:

搜索cd命令所在的目录

```
1 [root@heimatengyun test]# which cd
2 /usr/bin/cd
```

## 2.5 压缩解压

为了便于传输或节省存储空间有时候文件是以压缩包的形式存在, 因此就需要了解压缩与解压相关命令。

### 2.5.1 tar

语法: tar [参数] 包名.tar.gz 待打包的内容

功能描述: 打包目录, 压缩后的文件格式为.tar.gz

参数:

-c: create生成.tar打包文件

-x: extract解包.tar文件

-v: verbose显示详细信息

-f: file指定压缩后的文件名

-z: 打包同时压缩

-C: 解压到指定目录

案例:

压缩多个文件，将test.txt和test1.txt压缩为test.tar.gz

1	<code>tar -czvf</code>	名字	文件名	打包并压缩
2	<code>tar -zxvf</code>	文件名		解压缩并提取
3	<code>tar -cvf</code>	名字	文件名	打包

```
1 [root@heimatengyun test]# tar -czvf test.tar.gz test.txt test1.txt
2 test.txt
3 test1.txt
4 [root@heimatengyun test]# ll
5 total 10252
6 -rw-r--r--. 1 root root      26 Nov 23 20:40 test1.txt
7 -rw-r--r--. 1 root root    210 Nov 23 23:57 test.tar.gz
8 -rw-r--r--. 1 root root      66 Nov 23 21:56 test.txt
```

压缩目录

```
1 [root@heimatengyun ~]# tar -czvf test.tar.gz test/
```

解压到当前目录

```
1 [root@heimatengyun test]# tar -zxvf test.tar.gz
```

## 2.5.2 zip和unzip

语法:

压缩: `zip [参数] 包名.zip 待压缩内容`

解压: `unzip 包名.zip`

功能描述: 压缩文件和目录, windows和linux通用且可以压缩目录并保留源文件

参数:

-r: recurse-paths递归压缩目录

案例:

压缩test.txt或test1.txt为test.zip

```
1 [root@heimatengyun test]# zip test.zip test.txt test1.txt
2   adding: test.txt (deflated 15%)
3   adding: test1.txt (stored 0%)
```

解压test.zip

```
1 [root@heimatengyun test]# unzip test.zip
2 Archive:  test.zip
3   inflating: test.txt
4   extracting: test1.txt
```

### 2.5.3 gzip和gunzip

语法:

压缩: gzip [参数] 文件

解压: gzip [参数] 文件.gz

功能描述:

压缩: 压缩文件, 只能将文件压缩为.gz文件。只能压缩文件不能压缩目录, 压缩解压后不保留原来的文件。对单个文件压缩\*

解压: 解压文件

案例:

压缩文件

```
1 [root@heimatengyun test]# ll
2 total 8
3 -rw-r--r--. 1 root root 26 Nov 23 20:40 test1.txt
4 -rw-r--r--. 1 root root 66 Nov 23 21:56 test.txt
5 [root@heimatengyun test]# gzip test1.txt
6 [root@heimatengyun test]# ll
7 total 8
8 -rw-r--r--. 1 root root 54 Nov 23 20:40 test1.txt.gz
9 -rw-r--r--. 1 root root 66 Nov 23 21:56 test.txt
```

解压文件

```
1 [root@heimatengyun test]# ll
2 total 8
3 -rw-r--r--. 1 root root 54 Nov 23 20:40 test1.txt.gz
4 -rw-r--r--. 1 root root 66 Nov 23 21:56 test.txt
5 [root@heimatengyun test]# gunzip test1.txt.gz
6 [root@heimatengyun test]# ll
7 total 8
8 -rw-r--r--. 1 root root 26 Nov 23 20:40 test1.txt
9 -rw-r--r--. 1 root root 66 Nov 23 21:56 test.txt
```

## 2.6 日期命令

日期相关命令用于设置或获取系统日期。

### 2.6.1 date

语法: date [选项] [格式]

功能描述: 显示或设置时间

参数:

-s: set 以字符串格式设置时间

格式: \* (注意区分大小写) \*

+%Y: 显示当前年份

+%m: 显示当前月份

+%d: 显示当前是哪一天

+%H: 显示当前小时

+%M: 显示当前分钟

+%S: 显示当前秒数

+%Y%m%d: 显示当前年月日

"+%Y-%m-%d %H:%M:%S": 显示当前年月日时分秒 (用引号括起来)

案例:

设置时间

```
1 [root@heimatengyun test]# date -s "2019-11-24 11:05:10"
2 Sun Nov 24 11:05:10 CST 2019
```

显示时间

```
1 [root@heimatengyun test]# date
2 Sun Nov 24 11:02:21 CST 2019
3 [root@heimatengyun test]# date +%Y%m%d
4 20191124
5 [root@heimatengyun test]# date "+%Y-%m-%d %H:%M:%S"
6 2019-11-24 11:02:55
```

date查看非当前时间 (比如前一天, 后一天、取下周周一等等) 以及cal查看日历命令暂时不做讨论, 以后有机会在讨论。

## 2.7 进程线程命令

任务都以进程或线程的形式存在, 因此需要随时关注系统的进程, 查看是否有异常进程以及各进程占用系统资源的情况并使用不同的进程管理命令对进程管理和控制。

### 2.7.1 ps

语法: ps [选项]

功能描述: 查看系统中所有进程

参数:

-a: all 显示现行终端机下的所有程序, 包括其他用户的程序 (比如多克隆几个会话执行不同命令, 也会列出来)

-u: userlist 以用户为主的格式来显示程序状况

-x: 显示所有程序, 不以终端机来区分 (前面讲过终端有很多类型, 不仅显示当前终端)

案例：

```
1 [root@heimatengyun test]# ps -aux
2 USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
3 root          2966  0.0  0.1 116340  3192 pts/0    Ss   Nov22   0:03 -bash
4 ...省略部分内容
```

每一项内容的解释

项	含义
USER	进程是由哪个用户产生的
PID	进程ID
%CPU	该进程占用CPU的百分比，占用越高，进程越耗费资源
%MEM	该进程占用内存的百分比，占用越高，进程越耗费资源
VSZ	占用虚拟内存的大小，单位KB
RSS	占用实际物理内存的大小，单位KB
TTY	表示该进程在哪个终端中运行，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，常用状态有：R（运行）、S（睡眠）、T（停止状态）、s（包含子进程）、+（位于后台）
START	进程启动时间
TIME	进程执行时间，即占用cpu的运算时间，不是系统时间
COMMAND	产生此进程的命令名

ps -aux | grep xxx

2.7.2 top

语法：top [选项]

功能描述：查看系统健康状态

参数：

- d 秒数：Delay-time，指定top命令每隔几秒更新，默认是3秒。
- i: Idle-process，使top命令不显示任何闲置或者僵死进程
- p: Monitor-PIDs，通过指定监控进程ID来仅仅监控某个进程的状态
- s: Secure-mode，使top在安全模式运行，去除交互命令所带来的潜在危险

案例：

查看非僵死进程，一秒钟刷新一次

```
1 [root@heimatengyun test]# top -i -d 1
2 top - 12:39:19 up 1 day, 14:31, 3 users, load average: 0.04, 0.04, 0.05
3 Tasks: 395 total, 1 running, 391 sleeping, 3 stopped, 0 zombie
4 %Cpu(s): 0.0 us, 1.0 sy, 0.0 ni, 99.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0
5 st
6 KiB Mem: 1870784 total, 720520 used, 1150264 free, 880 buffers
7 KiB Swap: 2097148 total, 0 used, 2097148 free. 238616 cached Mem
8
9      PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
10 29987 root        20   0 123936   1868  1160 R   2.0   0.1   0:08.32 top
...省略部分内容
```

在此界面可以通过如下相应按键进行排序操作

操作键	说明
shift+n	以PID排序
shift+m	以内存排序
shift+p	以cpu使用率排序，默认选项
q	退出top

命令结果解释

第一行：任务队列信息

内容	说明
12:20:42	当前系统时间
up 1 day, 14:12	系统运行时间
3 users	当前登录用户数
load average: 0.08, 0.01, 0.05	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说明系统已经超出负载

第二行：进程信息

内容	说明
Tasks: 395 total	系统中的总进程数
1 running	正在运行的进程数
391 sleeping	睡眠的进程
3 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：cpu信息

内容	说明
%Cpu(s): 0.0 us	用户模式占用的cpu百分比
1.0 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
99.0 id	空闲cpu的百分比
0.0 wa	等待输入、输出的进程占用cpu的百分比
0.0 hi	硬中断请求服务占用的cpu百分比
0.0 si	软中断请求服务占用的cpu百分比
0.0 st	steal time虚拟世界百分比，当有虚拟机时，虚拟cpu等待时机cpu的时间百分比

ps：如果服务器有多个cpu，将显示多行

第四行：物理内存信息

内容	说明
KiB Mem: 1870784 total	物理内存的总量，单位KB
720520 used	已使用的物理内存数量
1150264 free	空闲的物理内存数量
880 buffers	作为缓冲的内存数量

第五行：交换分区信息

内存	说明
KiB Swap: 2097148 total	交换分区（虚拟内存）的总大小
0 used	已经使用的交换分区的大小
2097148 free	空闲交换分区的大小
238616 cached Mem	作为缓存的交换分区的大小

第六行：空行

第七行：表头信息





### 2.7.5 kill 常用

语法: kill [选项] 进程id

功能描述: 终止某个指定pid的服务进程

选项:

-9: 强迫进程立即停止

案例:

```
1 [root@heimatengyun test]# kill -9 20385
```

### 2.7.6 killall

语法: killall [选项] 服务名

功能描述: 终止某个指定名称的服务对应的所有进程

案例:

终止httpd服务的所有进程

```
1 [root@heimatengyun test]# killall httpd
```

## 2.8 系统状态检测命令

通过相关命令检查系统状态以及资源耗用情况, 保证系统健康稳定运行。

### 2.8.1 ifconfig

语法: ifconfig [网络设备] [参数]

功能描述: 获取网卡配置和网络状态信息

案例:

```
1 [root@heimatengyun test]# ifconfig
2 eno16777736: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.78.100 netmask 255.255.255.0 broadcast 192.168.78.255
4     inet6 fe80::20c:29ff:febc:5eef prefixlen 64 scopeid 0x20<link>
5     ether 00:0c:29:bc:5e:ef txqueuelen 1000 (Ethernet)
6     RX packets 143336 bytes 103106029 (98.3 MiB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 92576 bytes 143399144 (136.7 MiB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

主要查看内容: 网卡名称 (eno16777736)、ip地址 (inet之后) 物理网卡地址即mac地址 (ether)、TX和RX接收和发送数据包的个数和累计流量

## 2.8.2 netstat 重要

语法: netstat [参数]

功能描述: 显示整个系统目前网络情况, 比如目前的链接、数据包传递数据、路由表内容等

案例:

```
1 [root@heimatengyun test]# netstat
2 Active Internet connections (w/o servers)
3 Proto Recv-Q Send-Q Local Address           Foreign Address         State
4 tcp        0      96 192.168.78.100:ssh      192.168.78.1:59688
   ESTABLISHED
5 Active UNIX domain sockets (w/o servers)
6 Proto RefCnt Flags               Type               State              I-Node    Path
7 unix    2      [ ]                 DGRAM              13149
   /run/systemd/shutdown
8 ...省略部分内容
```

## 2.8.3 uname

语法: uname [选项]

功能描述: 查看系统内核和系统版本等信息

参数:

- a: all显示系统完整信息
- s: kernel-name系统内核名称
- n: nodename节点名称
- r: kernel-release内核发行版
- v: kernel-version内核版本
- m: machine硬件名称
- i: hardware-platform硬件平台
- p: processor处理器类型
- o: operating-system操作系统名称

案例:

```
1 [root@heimatengyun test]# uname
2 Linux
3 [root@heimatengyun test]# uname -a
4 Linux heimatengyun 3.10.0-123.el7.x86_64 #1 SMP Mon Jun 30 12:09:22 UTC 2014
   x86_64 x86_64 x86_64 GNU/Linux
```

显示信息依次为: 内核名称 (Linux)、主机名 (heimatengyun)、内核发行版 (3.10.0-123.el7.x86\_64)、内核版本 (#1 SMP Mon Jun 30 12:09:22 UTC 2014)、硬件名称 (x86\_64)、硬件平台 (x86\_64)、处理器类型 (x86\_64) 及操作系统名称 (GNU/Linux)。

如果要查看单签系统版本的详细信息, 通过查看/etc/redhat-release文件

```
1 [root@heimatengyun test]# cat /etc/redhat-release
2 CentOS Linux release 7.0.1406 (Core)
```

### 2.8.4 uptime

语法: uptime [选项]

功能描述: 查看系统的负载信息, 可以显示当前系统时间、系统已运行时间、启用终端数量以及平均负载值等信息。平均负载值指系统在最近1分钟、5分钟、15分钟内的压力情况, 负载值越低越好, 尽量不要长期超过1, 生产环境不要超过5。

案例:

```
1 [root@heimatengyun test]# uptime
2 21:30:44 up 1 day, 23:23, 3 users, load average: 0.00, 0.01, 0.05
```

### 2.8.5 free

语法: free [选项]

功能描述: 显示当前系统中内存的使用信息

选项:

-m: megabytes以兆字节显示

-h: human带单位输出

案例:

```
1 [root@heimatengyun test]# free -h
2              total        used        free      shared    buffers     cached
3  Mem:           1.8G         702M         1.1G          8.9M          880K          233M
4  -/+ buffers/cache:         468M         1.3G
5  Swap:           2.0G           0B          2.0G
```

字段说明: total (内存总量)、used (已用量)、free (可用量)、shared (进程共享的内存量)、buffers (磁盘缓存的内存量)、cached (缓存的内存量)

### 2.8.5 who

语法: who [参数]

功能描述: 查看当前登入主机的用户终端信息

案例:

```
1 [root@heimatengyun test]# who
2 root    pts/0        2019-11-22 22:10 (192.168.78.1)
3 root    pts/1        2019-11-23 11:53 (192.168.78.1)
```

### 2.8.6 last

语法: last [参数]

功能描述: 查看所有的系统登录记录。但是要注意, 这些信息是以日志文件保存的, 因此黑客可以很容易进行修改, 所以不能单纯以该命令来判断是否有黑客入侵。

案例:

```
1 [root@heimatengyun test]# last
2 root pts/2 192.168.78.1 Sun Nov 24 13:09 - 13:36 (00:26)
3 root pts/1 192.168.78.1 Sat Nov 23 11:53 still logged in
4 ... 省略部分内容
```

### 2.8.7 history

语法: history [参数]

功能描述: 显示历史执行过的命令

选项:

-c: 清除所有历史记录, 但是.bash\_history文件内容不会删除

案例:

```
1 [root@heimatengyun test]# history
2 1 history
3 2 ll
4 3 ls
5 4 history
6 [root@heimatengyun test]# !2
7 ll
8 total 8
9 -rw-r--r--. 1 root root 26 Nov 23 20:40 test1.txt
10 -rw-r--r--. 1 root root 66 Nov 23 21:56 test.txt
```

可以用"!编号"执行某一次曾经执行过的命令

默认可以查看1000条历史记录, 但可以在/etc/profile中修改HISTSIZE的值

历史命令保存在一个农户家目录的.bash\_history文件中(.开头为隐藏文件通过ls或ll -a列出), 可以用cat命令查看

## 2.9关机命令

linux一般用在服务器上, 很少遇到关机的情况, 毕竟关机服务就会中断, 除非特殊情况不得已才会关闭。

正确的关机流程: sync>shutdown或reboot或halt

**\*无论重启还是关机, 都需要先sync将内存数据同步到硬盘中, 避免数据丢失\***

### 2.9.1 reboot

语法: reboot [选项]

功能描述: 重启系统, 等同于shutdown -r now

案例:

```
1 [root@heimatengyun ~]# reboot
```

### 2.9.2 poweroff

语法: poweroff [选项]

功能描述: 关闭系统

案例:

```
1 [root@heimatengyun ~]# poweroff
```

### 2.9.3 halt

语法: halt [选项]

功能描述: 关闭系统, 等同于shutdown -h now和poweroff

案例:

```
1 [root@heimatengyun ~]# halt
```

### 2.9.4 shutdown

语法: shutdown [选项] [关机时间] [提示内容]

功能描述: 关机

选项:

-h: 关机

-r: 重启

关机时间:

hh:mm: 指定24小时制的小时和分钟后关机

+m: m分钟后关机(+1: 默认值, 1分钟后关机; +0: now, 立刻关机)

案例:

1分钟后关机并提示所有已登录系统的用户

```
1 [root@heimatengyun ~]# shutdown -h 1 "this server will shutdown after 1min"
2 this server will shutdown after 1min
3 The system is going down for power-off at Sun 2019-11-24 22:25:55 CST!
```

用户及文件相关权限命令将在以后的文章中讲解，还有很多其他的命令，由于篇幅所限不在此讨论。下一篇文章文章将分享“linux入门系列6--软件包管理之rpm和yum仓库”。

## 第三章 权限管理

### 一、权限简介

- Linux系统上对文件的权限有着严格的控制，用于如果相对某个文件执行某种操作，必须具有对应的权限方可执行成功。
- Linux下文件的权限类型一般包括读，写，执行。对应字母为 r、w、x。
- Linux下权限的粒度有 拥有者、群组、其它组 三种。每个文件都可以针对三个粒度，设置不同的 rwx(读写执行)权限。通常情况下，一个文件只能归属于一个用户和组，如果其它的用户想有这个文件的权限，则可以将该用户加入具备权限的群组，一个用户可以同时归属于多个组。
- Linux上通常使用chmod命令对文件的权限进行设置和更改。

### 二、创建用户和组

1、添加用户，Centos 没有任何交互动作！创建用户完毕后，必须修改密码否则无法登陆

```
1 # useradd zn #创建用户
2 # passwd zn #修改密码
```

切换命令

```
1 su - zn
```

建工作组

```
1 groupadd test
```

新建用户同时增加工作组

```
1 useradd -g test phpq //新建phpq用户并增加到test工作组
```

注：-g 所属组 -d 家目录 -s 所用的SHELL

给已有的用户增加工作组

 usermod命令

Linux usermod命令用于修改用户帐号。

usermod可用来修改用户帐号的各项设定。

## 语法

```
1 usermod [-LU][-c <备注>][-d <登入目录>][-e <有效期限>][-f <缓冲天数>][-g <群组>][-G <群组>][-l <帐号名称>][-s <shell>][-u <uid>][用户帐号]
```

### 参数说明：

- -c<备注> 修改用户帐号的备注文字。
- -d<登入目录> 修改用户登入时的目录。
- -e<有效期限> 修改帐号的有效期限。
- -f<缓冲天数> 修改在密码过期后多少天即关闭该帐号。
- -g<群组> 修改用户所属的群组。
- -G<群组> 修改用户所属的附加群组。
- -l<帐号名称> 修改用户帐号名称。
- -L 锁定用户密码，使密码无效。
- -s 修改用户登入后所使用的shell。
- -u 修改用户ID。
- -U 解除密码锁定。

<https://www.jianshu.com/p/5c393e414736>

## 实例

### 更改登录目录

```
1 # usermod -d /home/hnlinux root
```

### 改变用户的uid

```
1 # usermod -u 777 root
```

## 永久性删除用户账号

```
1 userdel testuser
2 groupdel testgroup
3 usermod -G testgroup testuser //（强制删除该用户的主目录和主目录下的所有文件和子目录）
```

## 显示用户信息

```
1 id user
2 cat /etc/passwd
```

## 三、基础权限

Linux文件一共有三种身份：

- u: 文件的拥有者(user)
- g: 文件所属的群组(group)
- o: 其他用户(other)

对于每个身份，又有四种权限：

- r: 读取文件的权限 (read)
- w: 写入文件的权限 (write)
- x: 执行的权限 (execute)
- s: 特殊权限(special)

## 四、【如何查询文件权限属性】

通过ls -al 文件名 指令来查询文件的属性

```
1 ls -al /Users/xyz/Desktop/crm副本.zip
2 -rw-r--r-- 1 xyz staff 746807 Jul 25 09:56 /Users/xyz/Desktop/crm副本.zip
```

以上7项依次表示【文件的属性-rw-r--r--】、【连接数1】、【文件的拥有者xyz】、【文件所属的群组staff】、【文件大小】、【文件创建时间】和【文件名称】

## 五、【文件属性解释】

在Linux系统中，文件的属性由10个字符来表示。

第一个字符表示文件的类型，其余9个字符分为三组，每组三个，分别表示文件的拥有者、群组以及其他人对该文件的访问权限。

每组依次表示读取、写入、和执行的权限，如果没有该权限，则以-显示。

对于

**-rw-r--r--**

分别为：

文件类型	文件拥有者的权限	群组的权限	其他人的权限
-	rw-	r--	r--

文件类型：

- |   |   |  |
|---|---|--|
| 1 | - | 一般文件   |
| 2 | d | 文件夹（或者叫目录）   |
| 3 | l | 符号链接文件（类似windows下的快捷方式一样的东西）   |
| 4 | b | 磁盘设备文件   |
| 5 | c | 字符设备文件（和磁盘设备文件，主要是和周边硬件连接，作为系统和硬件之间的接口）                                  |
| 6 | s | socket文件（内部进程通信的一种特殊文件，也可作为和远程主机通信的管道）                                   |
| 7 | p | 连接文件（是一种内部进程通信的机制，一个进程把数据写入Pipe中，另一个进程则由Pipe读取数据，数据采用先进先出（FIFO）的次序，称为管道） |

## 六、【chmod以字符形式改变文件】



给三种身份都赋予执行的权限

```
1 | chmod +x 文件名
```

等价于

```
1 | chmod a+x 文件名
```

也拆开写，a表示all所有人分别用u、g、o替换

对应：

去掉某个身份的某个权限，只需要将+变为-即可。比如

```
1 | chmod u-r 文件名
```

## 七、【chmod以数字形式改变文件】

读取权限：r 或者4

写入权限：w或者2

执行权限：x或者1

可读写可执行： $rw x = 4 + 2 + 1 = 7$

可读写不可执行： $rw - = 4 + 2 = 6$

可读不可写可执行： $r-x = 4 + 1 = 5$

### 常见权限形式

- -rw----- (600) 只有拥有者有读写权限。
- -rw-r--r-- (644) 只有拥有者有读写权限；而属组用户和其他用户只有读权限。
- -rwx----- (700) 只有拥有者有读、写、执行权限。
- -rwxr-xr-x (755) 拥有者有读、写、执行权限；而属组用户和其他用户只有读、执行权限。
- -rwx--x--x (711) 拥有者有读、写、执行权限；而属组用户和其他用户只有执行权限。
- -rw-rw-rw- (666) 所有用户都有文件读、写权限。
- -rwxrwxrwx (777) 所有用户都有读、写、执行权限。

范例：

```
1 | #设置所有人可以读写及执行
2 | chmod 777 file (等价于 chmod u=rwx,g=rwx,o=rwx file 或 chmod a=rwx file)
3 |
4 | #设置拥有者可读写，其他人不可读写执行
5 | chmod 600 file (等价于 chmod u=rw,g=---,o=--- file )
```

## 八、权限提升

除了切换用户

比如在zn用户下新增用户不被允许：

```
1 [zn@localhost root]$ useradd zn3
2 bash: /usr/sbin/useradd: 权限不够
```

## 使用sudo命令

```
1 [zn@localhost root]$ sudo useradd zn3
2 [sudo] zn 的密码:
3 zn 不在 sudoers 文件中。此事将被报告。
4 [zn@localhost root]$ [zn@localhost root]$ useradd zn3
5 bash: [zn@localhost: 未找到命令
6 您在 /var/spool/mail/root 中有新邮件
7 [zn@localhost root]$ bash: /usr/sbin/useradd: 权限不够
8 bash: bash:: 未找到命令
```

## 需要修稿sudoers

sudo 命令的配置文件为 /etc/sudoers。(注意, /etc/sudoers 的配置内容十分丰富, 我们仅做简单的介绍。要了解更多信息, 请参考 man sudoers。)

编辑这个文件是有单独的命令的 visudo(这个文件我们最好不要使用 vim 命令来打开), 是因为一旦你的语法写错会造成严重的后果, 这个工具会替你检查你写的语法, 这个文件的语法遵循以下格式:

### who where whom command

说白了就是哪个用户在哪个主机以谁的身份执行那些命令, 那么这个 where, 是指允许在那台主机 ssh 连接进来才能执行后面的命令, 文件里面默认给 root 用户定义了一条规则:

```
1 root    ALL=(ALL:ALL)    ALL
```

root 表示 root 用户。

ALL 表示从任何的主机上都可以执行, 也可以这样 192.168.100.0/24。

(ALL:ALL) 是以谁的身份来执行, ALL:ALL 就代表 root 可以任何人的身份来执行命令。

ALL 表示任何命令。

那么整条规则就是 root 用户可以在任何主机以任何人的身份来执行所有的命令。

sudo 组中的所有用户都具有通过 root 权限执行命令的能力!

再看个例子:

```
1 nick 192.168.10.0/24=(root) /usr/sbin/useradd
```

上面的配置只允许 nick 在 192.168.10.0/24 网段上连接主机并且以 root 权限执行 useradd 命令。

## 设置 sudo 时不需要输入密码

执行 sudo 命令时总是需要输入密码事件很不爽的事情(抛开安全性)。有些应用场景也需要在执行 sudo 时避开输入密码的交互过程。

那么需要如何设置呢? 其实很简单, 只需要在配置行中添加 NOPASSWD: 就可以了:

```
1 zn ALL=(ALL) NOPASSWD: ALL
```

再试试看, 是不是已经不需要输入密码了?

# 第四章 安装软件

## 一、CentOS 更换阿里yum源

### 1、备份

```
1 | mv /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo.backup
```

### 2、下载新的 CentOS-Base.repo 到 /etc/yum.repos.d/

#### CentOS 6

```
1 | wget -O /etc/yum.repos.d/CentOS-Base.repo  
https://mirrors.aliyun.com/repo/Centos-6.repo
```

或者

```
1 | curl -o /etc/yum.repos.d/CentOS-Base.repo  
https://mirrors.aliyun.com/repo/Centos-6.repo
```

#### CentOS 7

```
1 | wget -O /etc/yum.repos.d/CentOS-Base.repo  
https://mirrors.aliyun.com/repo/Centos-7.repo
```

或者

```
1 | curl -o /etc/yum.repos.d/CentOS-Base.repo  
https://mirrors.aliyun.com/repo/Centos-7.repo
```

#### CentOS 8

```
1 | wget -O /etc/yum.repos.d/CentOS-Base.repo  
https://mirrors.aliyun.com/repo/Centos-8.repo
```

或者

```
1 | curl -o /etc/yum.repos.d/CentOS-Base.repo  
https://mirrors.aliyun.com/repo/Centos-8.repo
```

### 3. 运行 yum makecache 生成缓存

### 4. 其他

非阿里云ECS用户会出现 Couldn't resolve host 'mirrors.cloud.aliyuncs.com' 信息，不影响使用。用户也可自行修改相关配置: eg:

```
1 | sed -i -e '/mirrors.cloud.aliyuncs.com/d' -e '/mirrors.aliyuncs.com/d'
   | /etc/yum.repos.d/CentO
```

安装一个vim试试

```
1 | yum install vim -y
```

## 二、安装jdk，配置环境变量

### 1、下载jdk8

登录网址：<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

选择对应jdk版本下载。（可在Windows下下载完成后，通过文件夹共享到Linux上）

文件包中有。

### 2. 在usr目录下建立java安装目录

```
1 | cd /usr
2 | mkdir java
```

### 3.将jdk-8u60-linux-x64.tar.gz拷贝到java目录下

```
1 | cp /mnt/hgfs/linux/jdk-8u60-linux-x64.tar.gz /usr/java/
```

### 4.解压jdk到当前目录

```
1 | tar -zxvf jdk-8u60-linux-x64.tar.gz
2 | 得到文件夹 jdk1.8.0_60
```

### 5.安装完毕为他建立一个链接以节省目录长度

可有可无，就是创建一个快捷键

```
1 | （我没用这一步）
2 | ln -s /usr/java/jdk1.8.0_60/ /usr/jdk
```

## 6.编辑配置文件，配置环境变量

```
1 vim /etc/profile
2
3 JAVA_HOME=/usr/jdk
4 CLASSPATH=$JAVA_HOME/lib/
5 PATH=$PATH:$JAVA_HOME/bin
```

## 7.执行命令：

```
1 source /etc/profile
```

## 8.查看安装情况

```
1 java -version
2 java version "1.8.0_60"
3 Java(TM) SE Runtime Environment (build 1.8.0_60-b27)
4 Java HotSpot(TM) Client VM (build 25.60-b23, mixed mode)
```