



目录

Linux 发展史与安装.....	1
一、Linux 发展史.....	1
1、Linux 前身-Uinx.....	1
2、Linux 诞生.....	1
3、开源文化.....	2
4、Linux 系统特点.....	2
5、Linux 分支.....	2
二、Linux 系统的安装.....	2
1、安装方式.....	2
2、虚拟机软件（了解）.....	3
3、虚拟机的安装.....	3
3.1、VMware Workstation 的安装.....	3
3.2、Virtual Box 的安装.....	8
4、Linux 版本的选择.....	11
5、新建虚拟机.....	12
5.1、使用 VMware Workstation（重点）.....	12
5.2、使用 Virtual Box（了解）.....	20
6、Linux 操作系统安装.....	23
6.1、使用 VMware workstation 进行安装.....	23
6.2、使用 virtual Box 安装 Linux（了解）.....	33
7、终端.....	33
8、使用 VMware 备份操作系统.....	35
三、Linux 系统的文件.....	39
1、文件与文件夹（目录）.....	39
2、Linux 系统的文件目录结构.....	39
Linux 的基本指令.....	40
一、指令与选项.....	40
二、基础指令（重点）.....	41
1、ls 指令.....	41
2、pwd 指令.....	42
3、cd 指令.....	43
4、mkdir 指令.....	43
5、touch 指令.....	44
6、cp 指令.....	45
7、mv 指令.....	45
8、rm 指令.....	46
9、vim 指令.....	47
10、输出重定向.....	48
11、cat 指令.....	48
三、进阶指令（重点）.....	49

1、df 指令	49
2、free 指令	49
3、head 指令	49
4、tail 指令	50
5、less 指令	50
6、wc 指令	50
7、date 指令（重点）	51
8、cal 指令	51
9、clear/ctrl + L 指令	52
10、管道（重要）	52
Linux 的基本指令（2）	53
一、高级指令	53
1、hostname 指令	53
2、id 指令	53
3、whoami 指令	54
4、ps -ef 指令（重点）	54
5、top 指令（重点）	55
6、du -sh 指令	56
7、find 指令	56
8、service 指令（重点）	57
9、kill 指令（重点）	58
10、ifconfig 指令（重点）	58
11、reboot 指令	59
12、shutdown 指令	59
13、uptime 指令	60
14、uname 指令	60
15、netstat -tnlp 指令	60
16、man 指令	61
二、练习题	61
编辑器之神——vim 编辑器	62
一、vi 介绍	62
二、vim 三种模式（重点）	62
三、命令模式	63
1、光标移动	63
2、复制操作	63
3、剪切/删除	64
4、撤销/恢复	64
5、扩展 1：光标的快速移动	64
四、模式间的切换（重点）	65
五、末行模式	65
六、编辑模式	67
七、实用功能	67



1、代码着色.....	67
2、vim 中计算器的使用.....	68
八、扩展（3）.....	68
1、vim 的配置（重点）.....	68
2、异常退出.....	69
3、别名机制（实用）.....	70
4、退出方式.....	70
九、作业.....	71
Linux 自有服务（1）.....	71
一、运行模式.....	71
二、用户与用户组管理（重点）.....	72
1、用户管理.....	73
2、用户组管理.....	76
三、网络设置.....	78
四、ssh 服务（重点）.....	79
1、远程终端.....	80
2、SSH 服务文件传输.....	82
五、作业.....	84
Linux 自有服务（2）.....	84
一、设置主机名.....	85
二、chkconfig.....	86
三、ntp 服务.....	87
四、防火墙服务.....	88
五、rpm 管理（重点）.....	90
六、cron/crontab 计划任务（重点）.....	93
Linux 的权限管理操作.....	95
一、权限概述.....	95
1、权限介绍.....	95
2、身份介绍.....	95
Owner 身份（文件所有者，默认为文档的创建者）.....	95
Group 身份（与文件所有者同组的用户）.....	96
Others 身份（其他人，相对于所有者）.....	96
Root 用户（超级用户）.....	96
3、Linux 的权限介绍.....	96
二、权限设置.....	97
1、字母形式.....	98
2、数字形式.....	99
3、注意事项.....	100
三、属主与属组设置.....	101
1、chown（重点）.....	101
2、chgrp（了解）.....	101

四、扩展 (1)	102
Linux 的网络基础.....	104
一、网络相关概述.....	104
1、网络发展.....	104
2、ip 地址 (重点记忆)	105
3、网卡.....	106
4、网线.....	107
5、交换机.....	107
6、路由器.....	108
7、拓扑结构图 (扩展)	108
二、网络相关命令.....	109
1、ping.....	109
2、netstat	110
3、tracert.....	110
4、arp	111
5、tcpdump(了解).....	112
三、项目上线流程 (必须掌握)	113
1、服务器选配购买.....	113
2、域名购买.....	117
3、域名备案.....	118
4、域名解析.....	120
5、配置生产环境 (最后 1 天)	122
6、上传代码.....	122
Shell 基础	122
一、关于 shell.....	122
1、什么是 shell	122
2、shell 入门.....	123
二、shell 进阶 (重点)	125
1、变量 (重点)	125
1.1、变量的含义.....	125
1.2、变量的定义与使用 (重点)	125
1.3、只读变量 (了解)	127
1.4、接收用户输入 (重点)	127
1.5、删除变量 (了解)	127
2、条件判断语句.....	128
3、运算符.....	129
3.1、算数运算符.....	129
3.2、关系运算符.....	131
3.3、逻辑运算符.....	133
3.4、字符串运算符.....	134
3.5、文件测试运算符 (重点)	135
4、shell 脚本附带选项 (重点)	136



三、作业.....	138
MySQL 基础.....	138
一、关于数据库.....	138
1、什么是数据库.....	138
2、MySQL 简介.....	138
3、常见的其他数据库软件.....	139
二、MySQL 的安装与初始化.....	139
1、Linux 下的软件安装方式（初步）.....	139
a. 源码包（建议）.....	139
b. 二进制包（rpm）.....	141
c. yum 等傻瓜式安装.....	141
2、安装 MySQL（重点）.....	142
2.1、MySQL 安装.....	142
2.2、MySQL 初始化.....	143
2.3、MySQL 的启动控制.....	144
2.4、默认目录/文件位置（了解）.....	144
三、MySQL 的基本操作（难点）.....	144
1、名词介绍.....	144
2、库操作（重点）.....	144
3、表操作.....	145
4、记录/字段操作（重点）.....	147
4.1、增加记录.....	147
4.2、更新记录.....	148
4.3、查询记录.....	148
4.4、删除记录.....	149
5、备份与还原（重点）.....	149
5.1、备份（导出）.....	149
5.2、还原（导入）.....	150
四、扩展.....	152
1、mysql 的远程管理工具.....	152
五、作业.....	154
Yum 项目上线实战（网站运维）.....	154
一、编译安装与卸载 Nginx.....	154
1、安装 nginx.....	155
二、关于 LAMP.....	158
三、LAMP 环境部署.....	159
1、PHP 与 Apache 的安装.....	160
2、MySQL 的安装与初始化.....	161
3、项目上线.....	163

Linux 发展史与安装

一、Linux 发展史

1、Linux 前身-Unix

1968 年 Multics 项目

MIT、Bell 实验室、美国通用电气有限公司走到了一起，致力于开发 Multics 项目。到后期由于开发进度不是很好，MIT 和 Bell 实验室相继离开这个项目的开发，最终导致项目搁浅。

1970 年（Unix 元年，时间戳） Unix 诞生

当时在开发 Multics 项目的时候，实验室中有一个开发成员开发了一款游戏（travel space: 遨游太空），因为两个实验室相继离开项目开发，导致这名开发人员没法玩游戏，后来他提议组织人员重新在 Multics 项目之上重新的开发，也就出现了 1970 年的 Unix。当时 Unix 操作系统是使用的汇编语言（机器语言）开发的。

1973 年 用 C 语言重写 Unix

因为汇编语言有一个最大的局限性：对于计算机硬件过于依赖。导致移植性不好，所以后期在 1973 年使用了 C 语言对其进行重新开发。

1975 年 Bell 实验室允许大学使用 Unix。

1975 年，bell 实验室允许大学使用 Unix 操作系统用于教学作用，而不允许用于商业用途。

2、Linux 诞生

人物 Linus

Linux 的开发作者，Linux 之父，李纳斯·托瓦兹。Linux 诞生时是荷兰在校大学生。



1991 年 0.0.1 版本

李纳斯当时学校使用的就是 Unix 操作系统，然后其对系统的底层代码进行了修改，放到了学校为学生开放的网站上，原先他把文件名写成了 Linus's Unix，后期网络管理发现之后觉得这个名字不好，自己手动的将名字改成 Linux。随后其他同学下载之后发现这个版本还是挺好用的，随后都把自己代码贡献给李纳斯。

1992 年 0.0.2 版本

1994 年 1.0 版本

2003 年 2.6 版本

上述所提及的版本号并不是分支版本，而是指 Linux 的内核版本。

3、开源文化

Linux 是**开源**的操作系统。所谓开源就是指开放源代码。

人物 Stallman 斯特曼，开源文化的倡导人。

1983 年 GNU 计划

1985 年 FSF 基金会

1990 年 Emacs、**GCC**（c 语言的编译器）、程序库

1991 年 Stallman 去找 Linus，商谈让 Linux 加入其开源计划（GNU 计划）

1992 年 GNU/Linux

4、Linux 系统特点

开放性（开源）、多用户、多任务、良好的用户界面、优异的性能与稳定性

多用户多任务：

单用户：一个用户，在登录计算机（操作系统），只能允许**同时**登录一个用户；

单任务：一个任务，允许用户**同时**进行的操作任务数量；

多用户：多个用户，在登录计算机（操作系统），允许**同时**登录多个用户进行操作；

多任务：多个任务，允许用户**同时**进行多个操作任务；

Windows 属于：单用户、多任务。

Linux 属于：多用户、多任务。

5、Linux 分支

分支：Linux 分支有很多，现在比较有名的 ubuntu、debian、**centos**（Community Enterprise Operating System）、redhat、suse 等等。

二、Linux 系统的安装

1、安装方式

目前安装操作系统方式有 2 种：真机安装、虚拟机安装。

真机安装：使用真实的电脑进行安装，像安装 windows 操作系统一样，真机安装的结果就



是替换掉当前的 windows 操作系统：

虚拟机安装：通过一些特定的手段，来进行模拟安装，**并不会影响当前计算机的真实操作系统；**

如果是学习或者测试使用，强烈建议使用虚拟机安装方式。

2、虚拟机软件（了解）

什么是虚拟机？

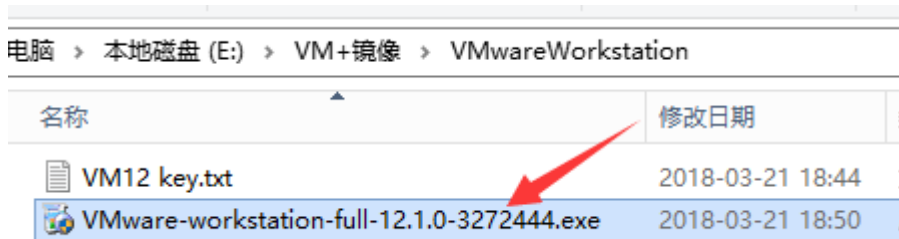
虚拟机，有些时候想模拟出一个真实的电脑环境，碍于使用真机安装代价太大，因此而诞生的一款可以模拟操作系统运行的软件。

虚拟机目前有 2 个比较有名的产品：vmware 出品的 vmware workstation、oracle 出品的 virtual Box。

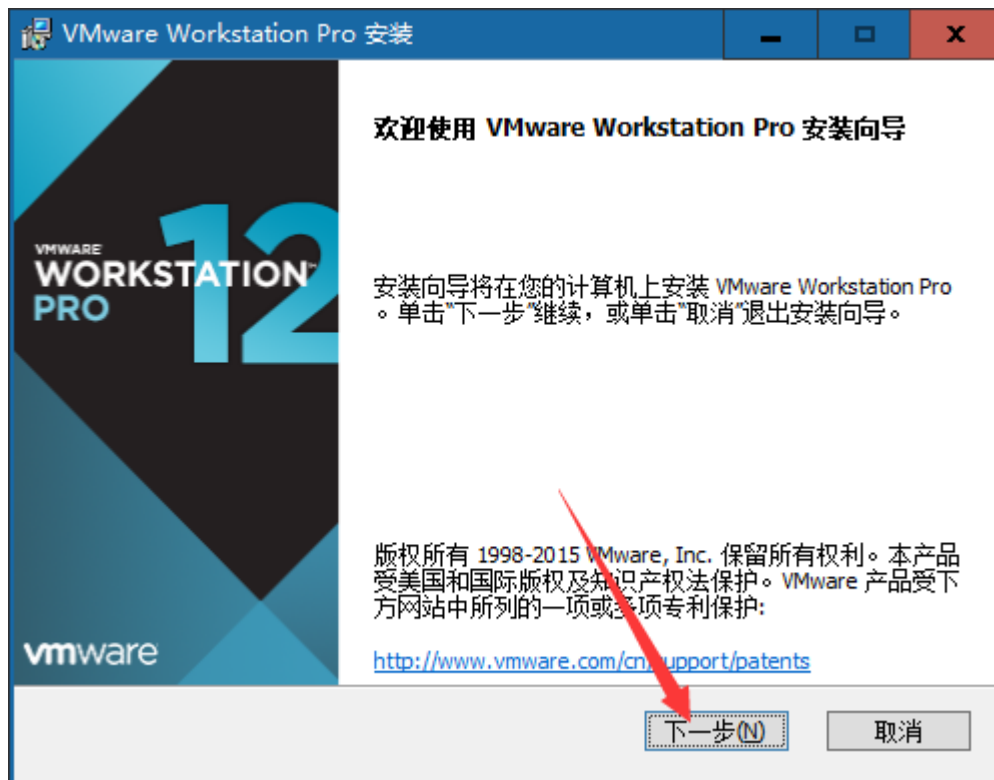
3、虚拟机的安装

3.1、VMware Workstation 的安装

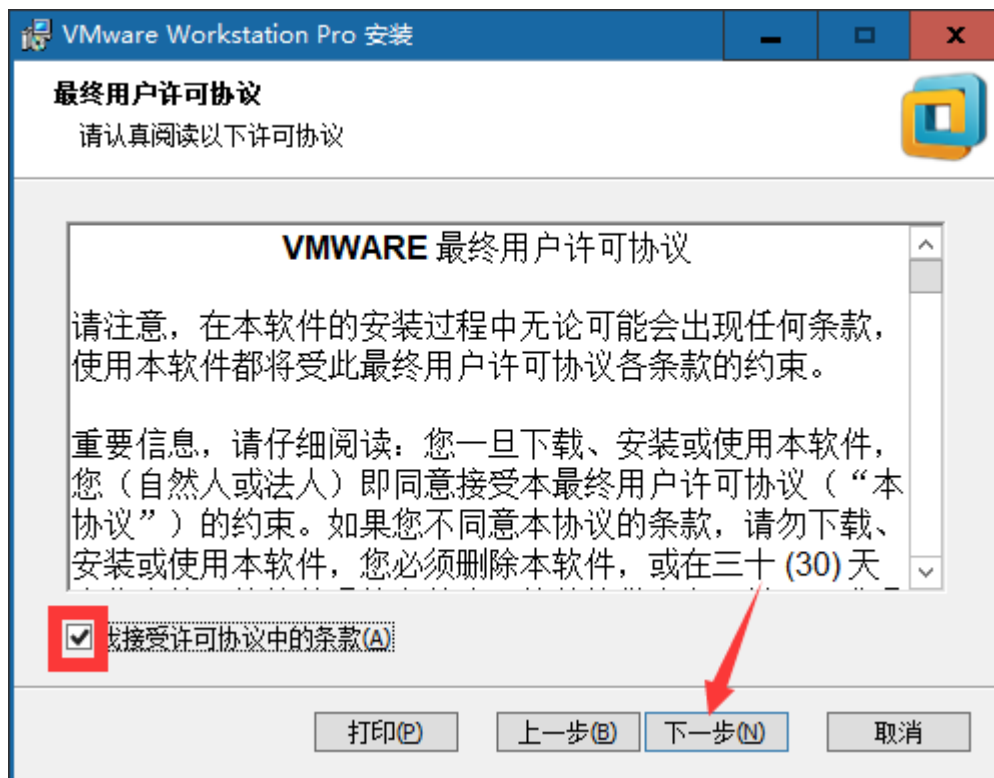
①打开安装程序



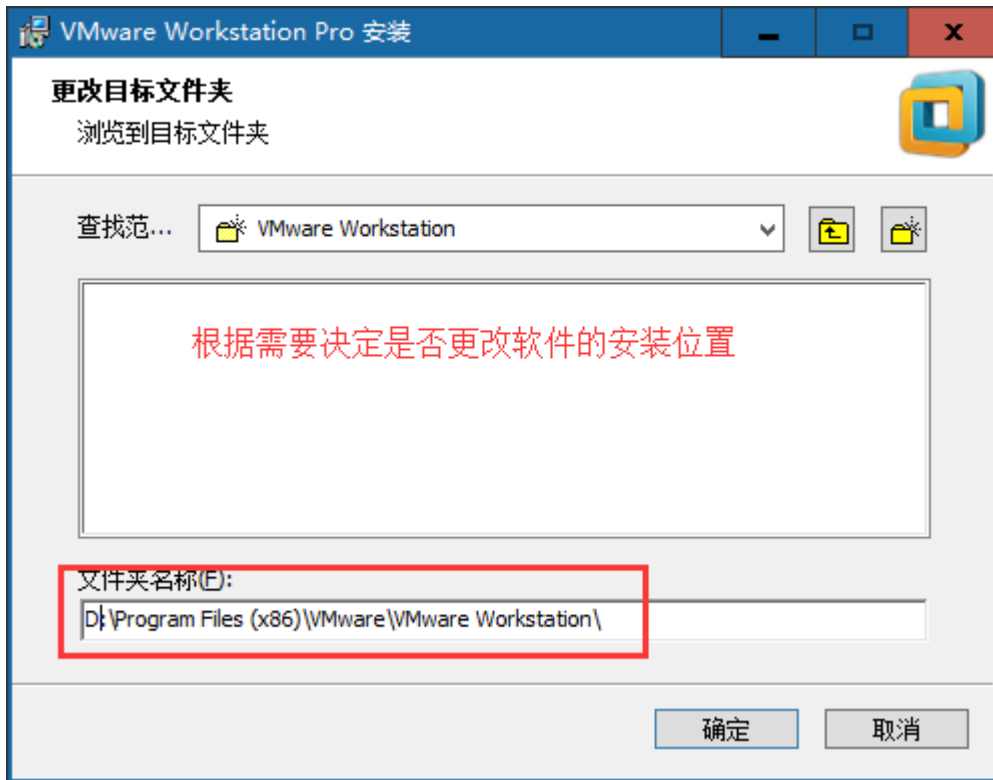
②进行下一步开始安装



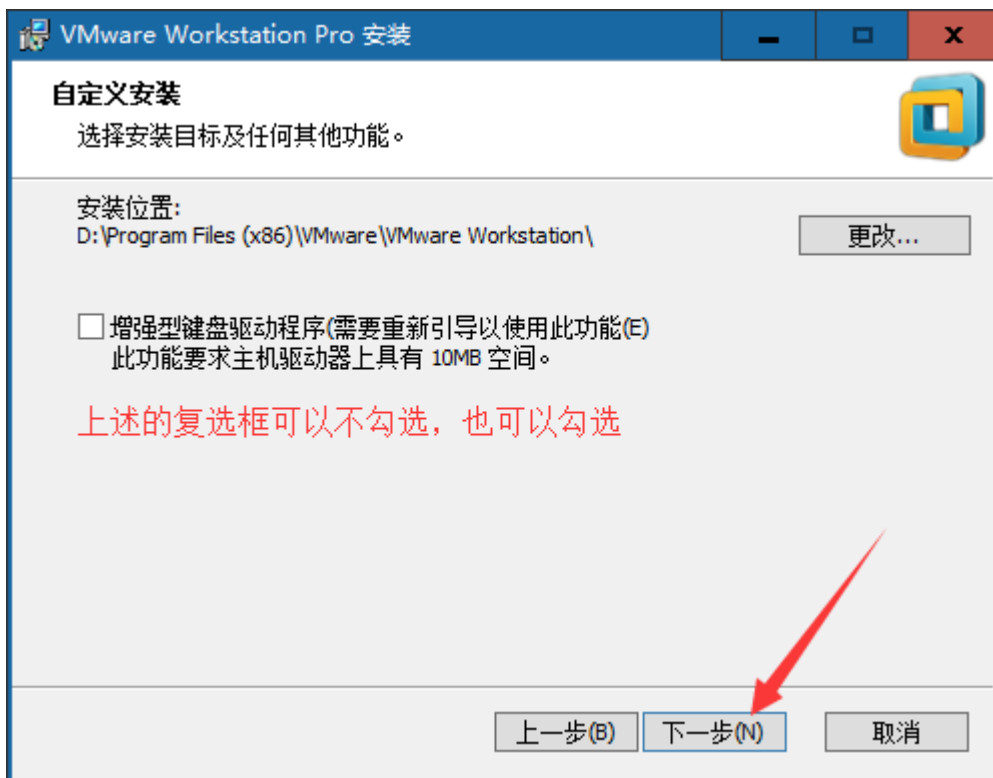
③同意许可协议



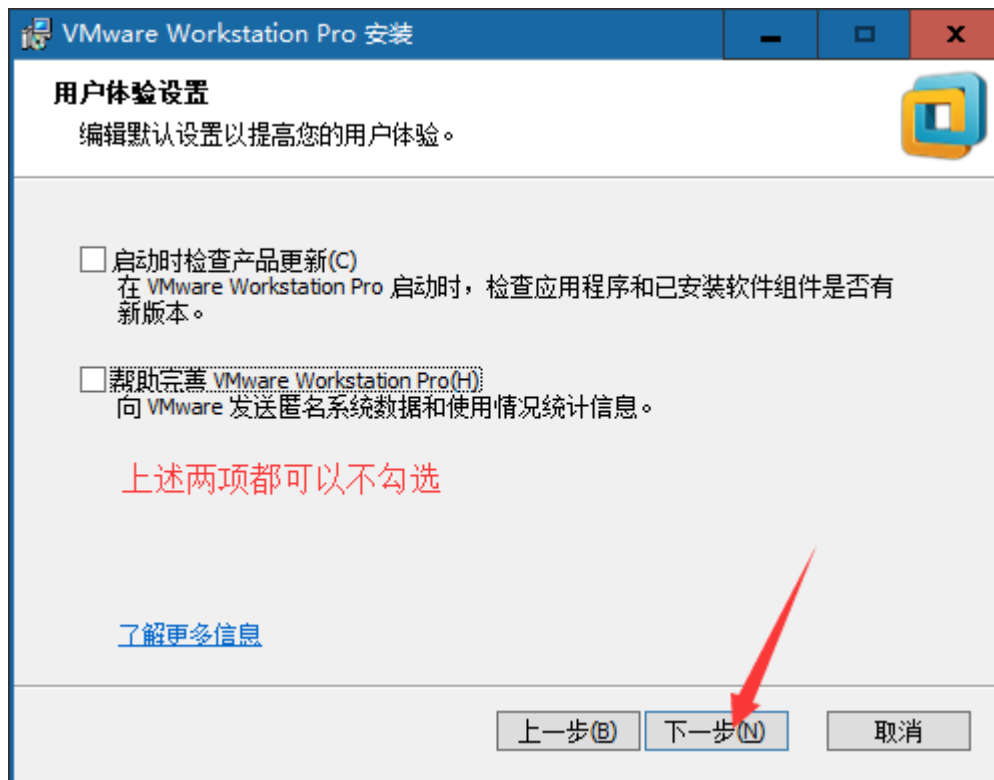
④根据需要决定是否需要更改软件的安装位置



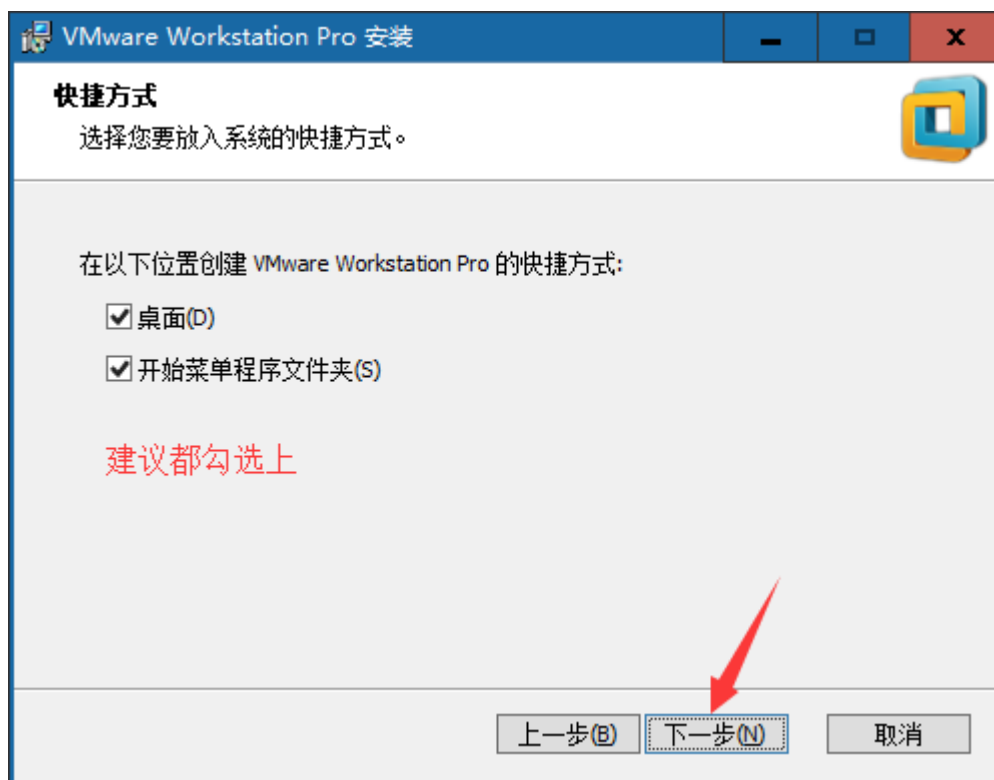
点击下一步：



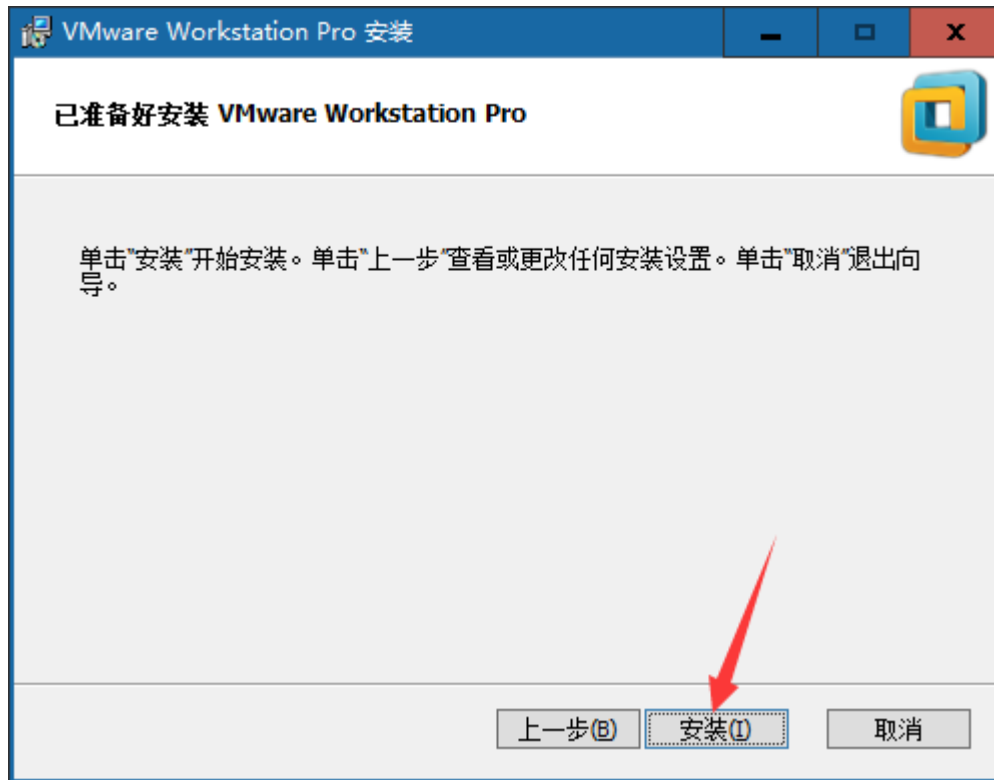
⑤用户体验设置



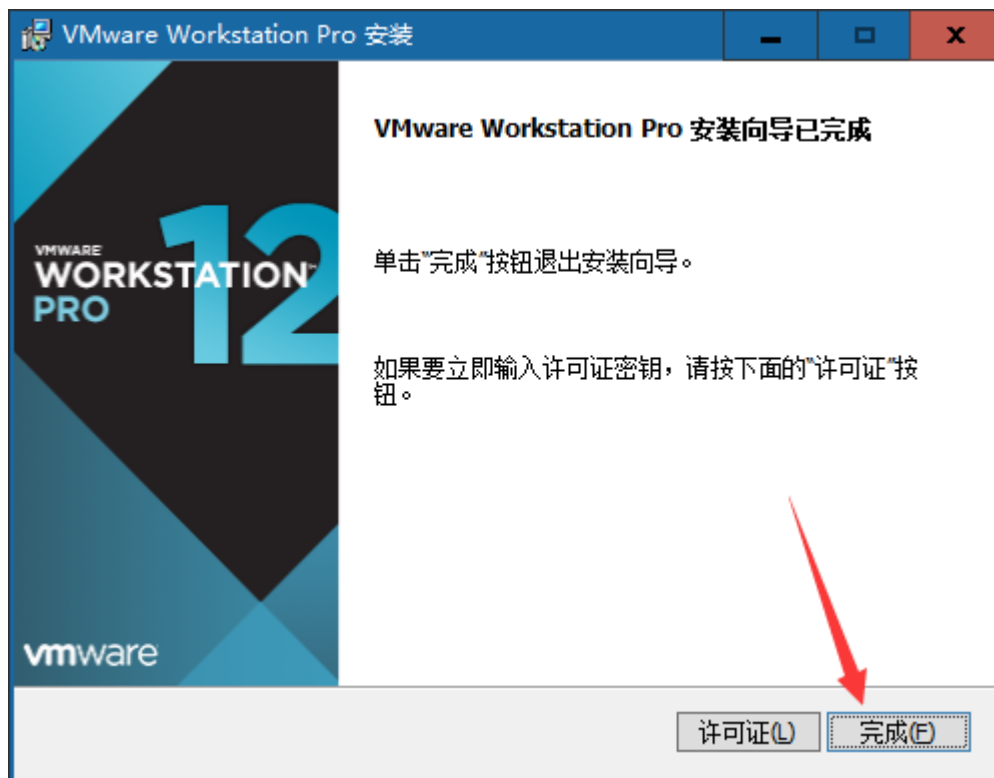
⑥快捷方式创建的步骤



⑦点击安装按钮



⑧ 点击完成



⑨ 最重要的地方，在安装完之后需要检查，检查虚拟机软件是否有安装 2 个虚拟网卡



如果没有这 2 个网卡的话，则会影响后期 windows 系统与虚拟机其中操作系统之间的相互通信（比如共享网络、文件传输等）。

3.2、Virtual Box 的安装

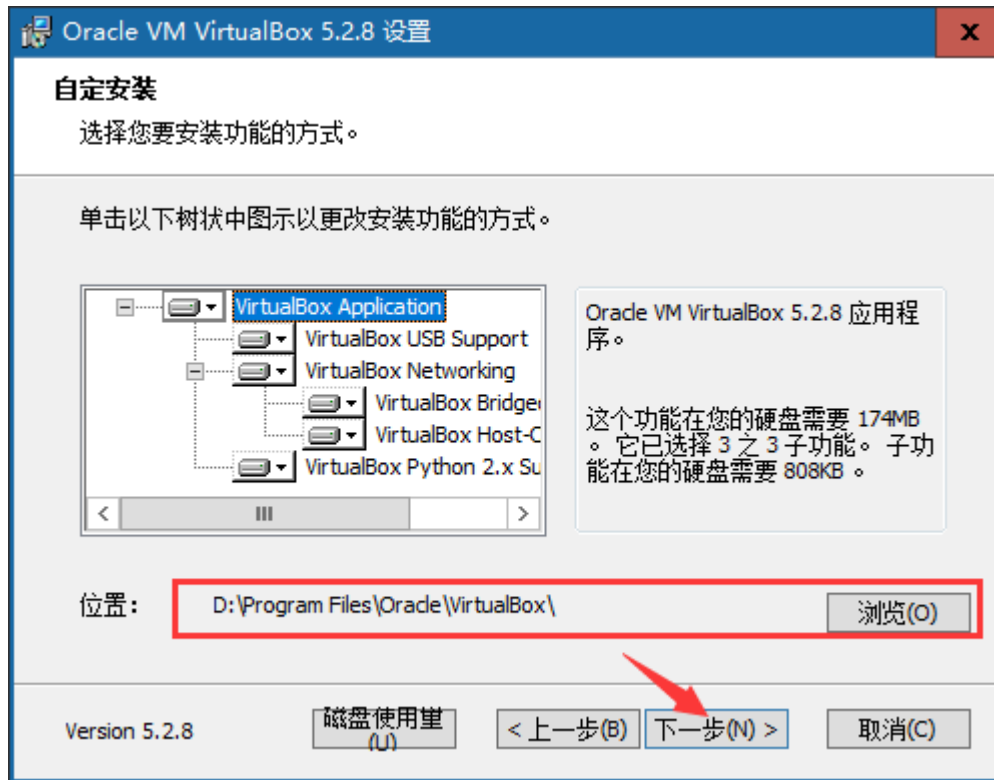
①运行安装程序



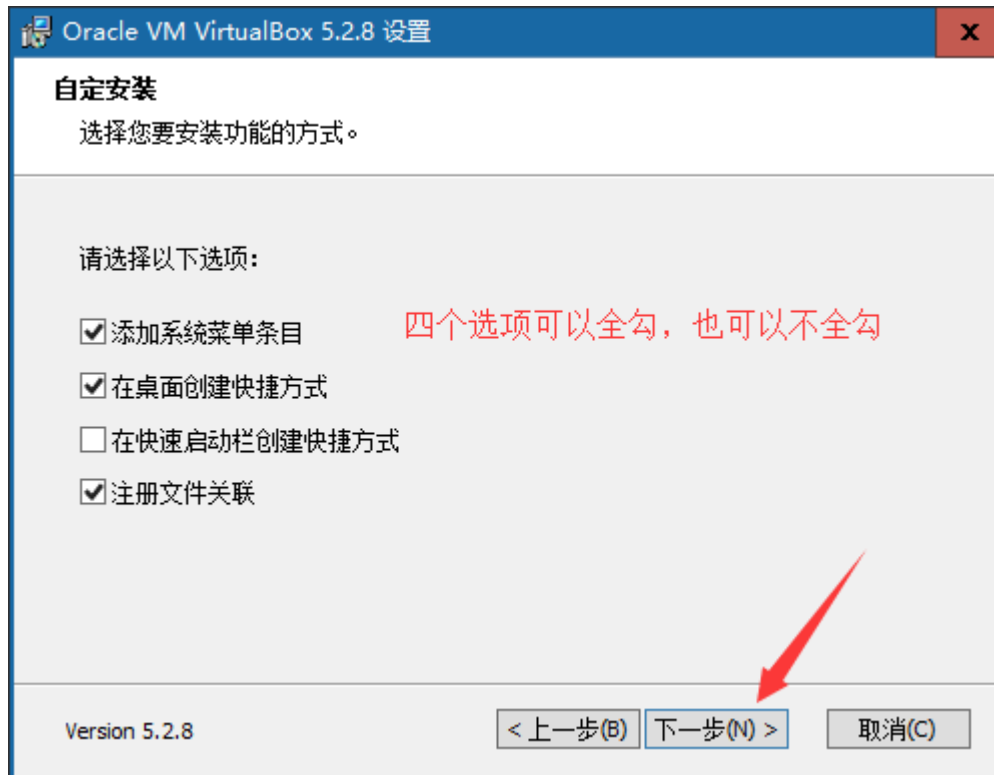
②点击下一步



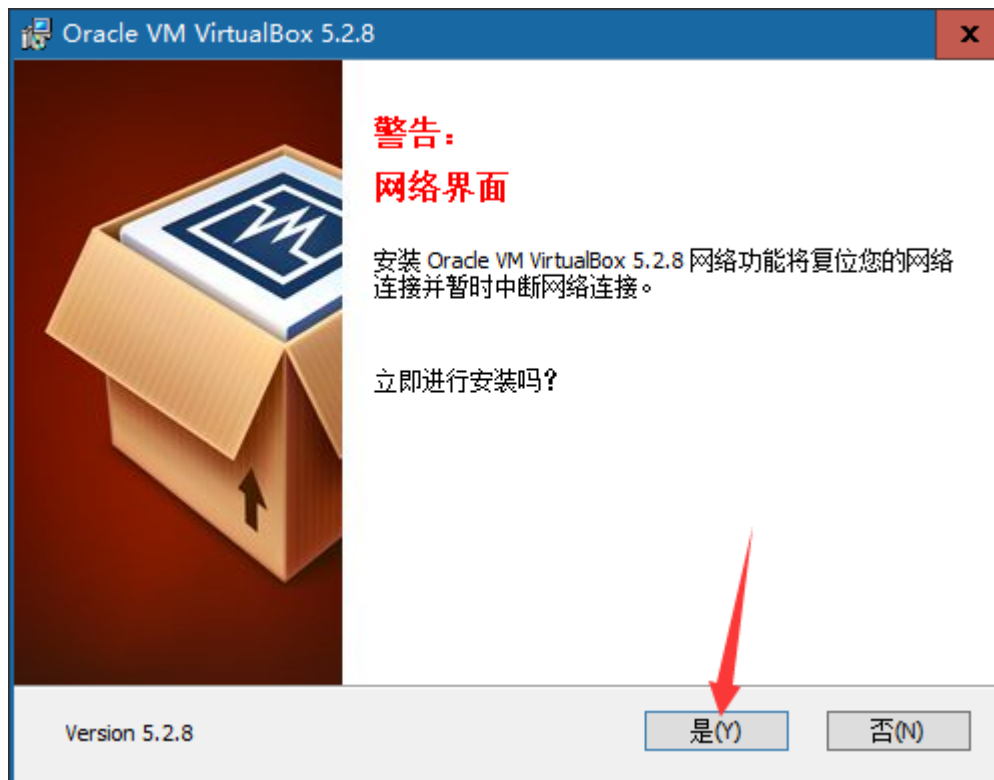
③选择性更改安装的位置



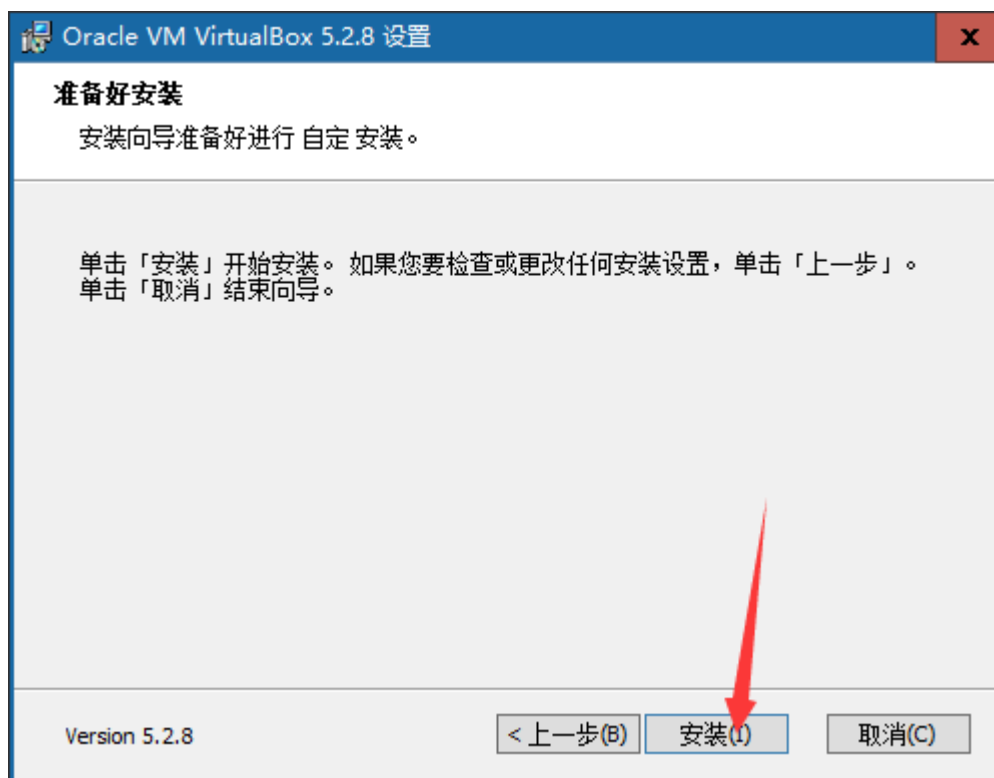
④ 安装的选项设置



⑤ 断网警告，点击是即可



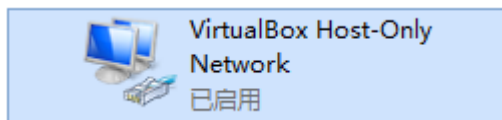
⑤ 点击安装按钮



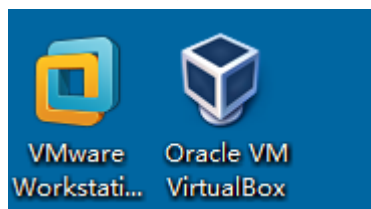
⑥ 完成



⑦检查是否有对应的虚拟网卡存在



两个软件安装完成之后在桌面上都有对应的快捷方式



4、Linux 版本的选择

版本选择：CentOS 6.5 【镜像一般都是 xxx.iso 文件】

问题：为什么不选择最新版的 7.x 版本？

- 6.x 目前依然是主流
- 6.x 的各种系统操作模式是基础
- 7.x 实际上也支持大多数 6.x 的操作形式

官网：<https://www.centos.org/>

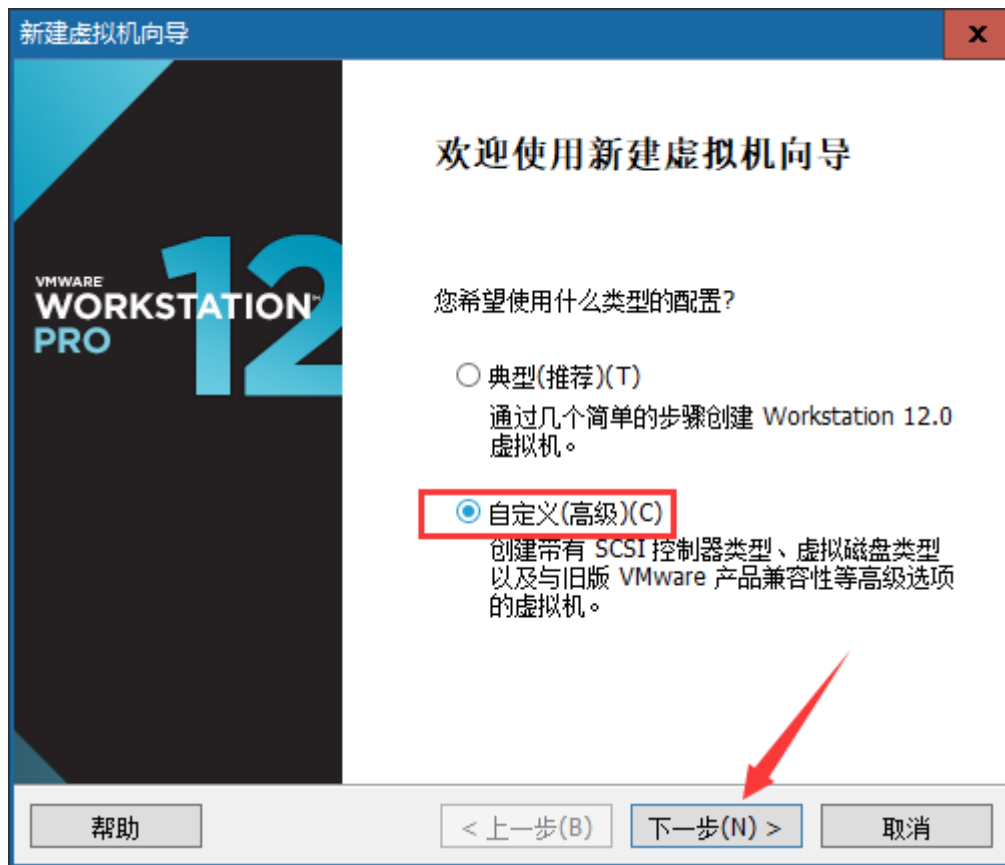
从官网下载得到的镜像文件

脑 > 本地磁盘 (E:) > VM+镜像 > CentOS-6.5		
名称	修改日期	类型
 CentOS-6.5-i386-bin-DVD1.iso	2018-03-21 18:50	WinRAR

5、新建虚拟机

5.1、使用 VMware Workstation（重点）

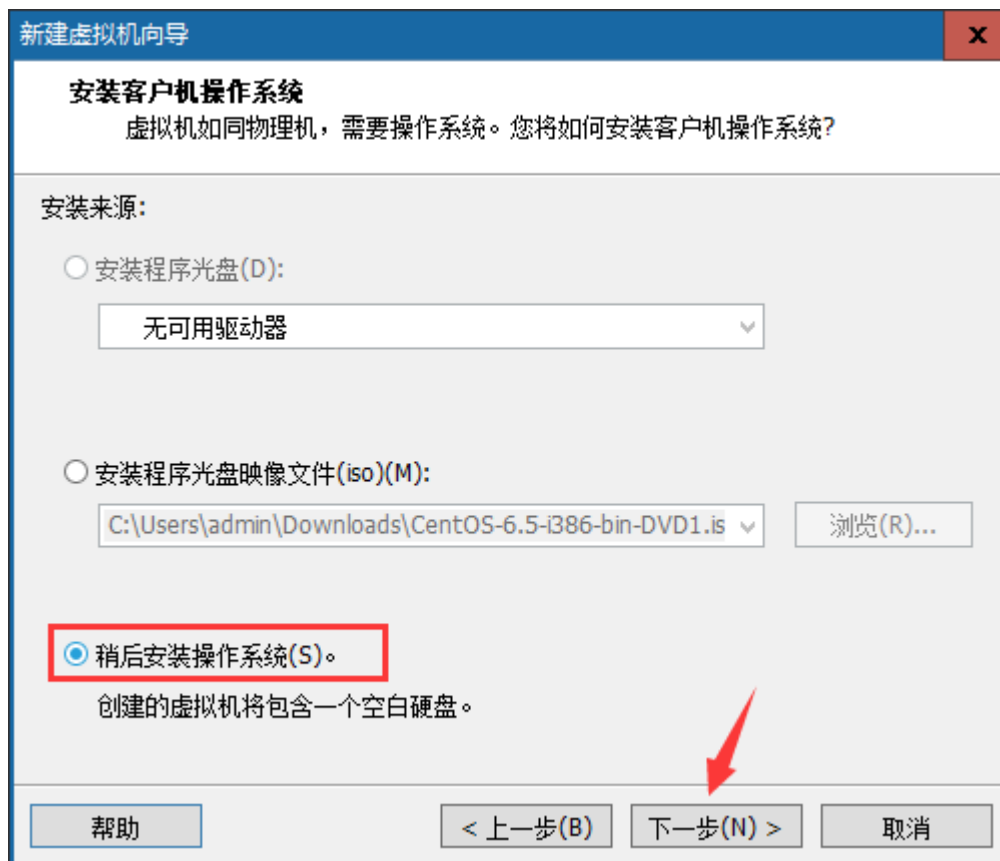
①点击“文件”菜单，选择“新建虚拟机...”选项，选择“自定义”点击下一步



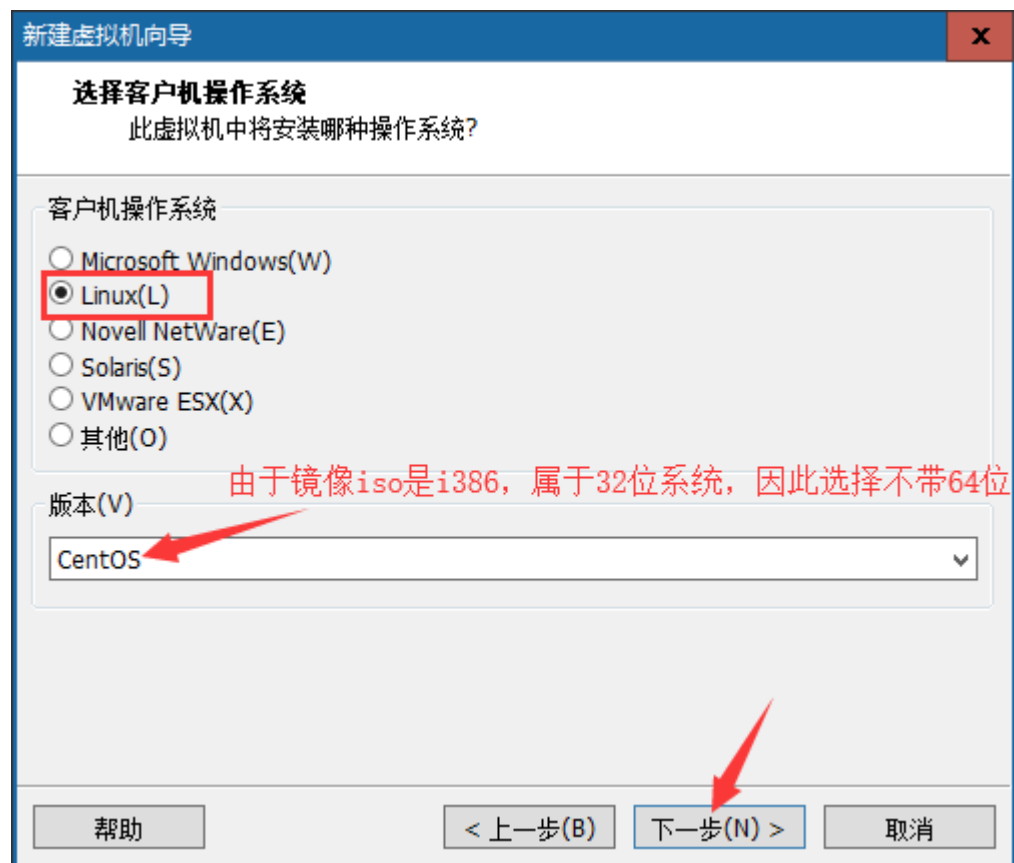
②选择兼容性，默认即可，直接下一步



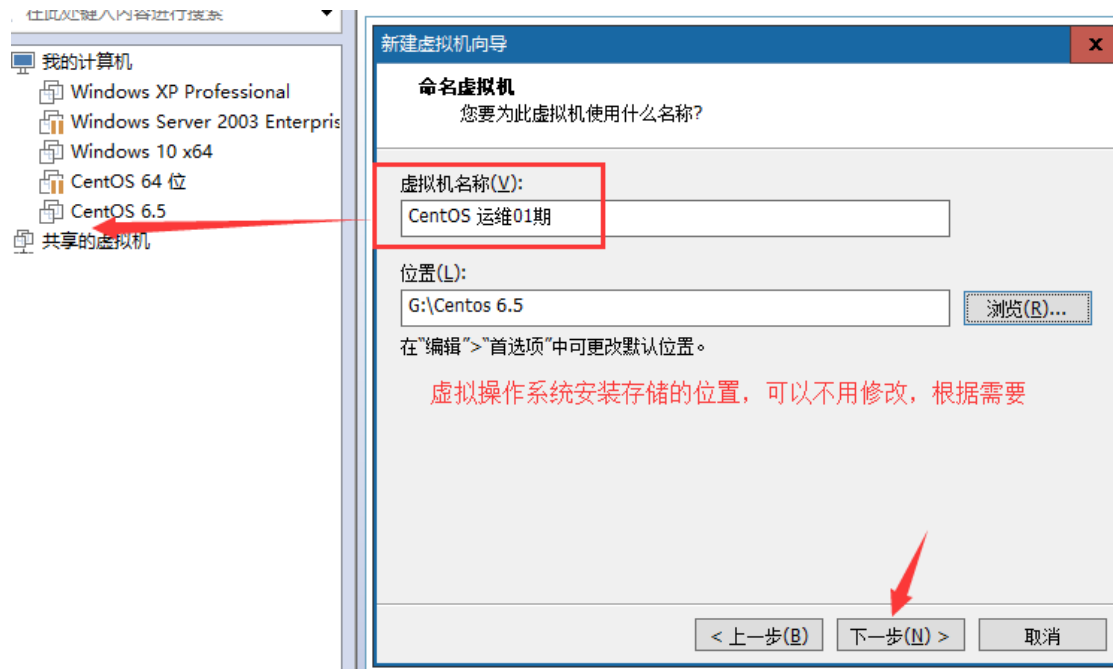
③选择镜像文件的时候选择“稍后...”，点击下一步



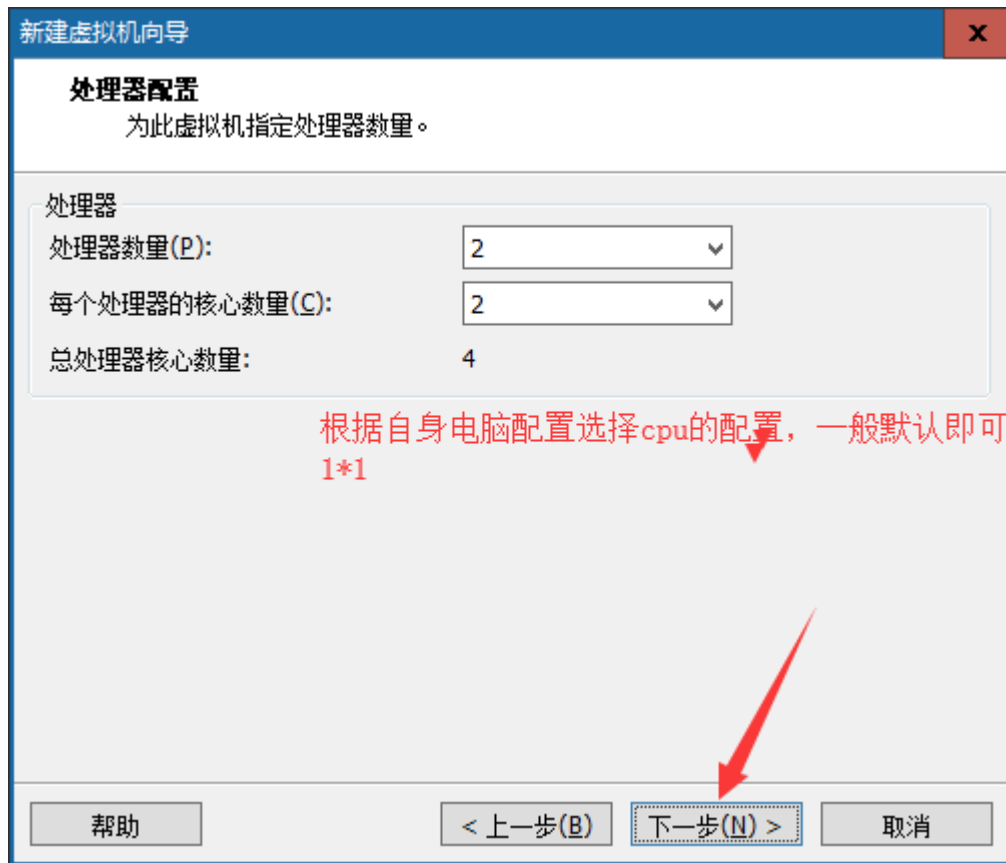
④选择需要安装的操作系统



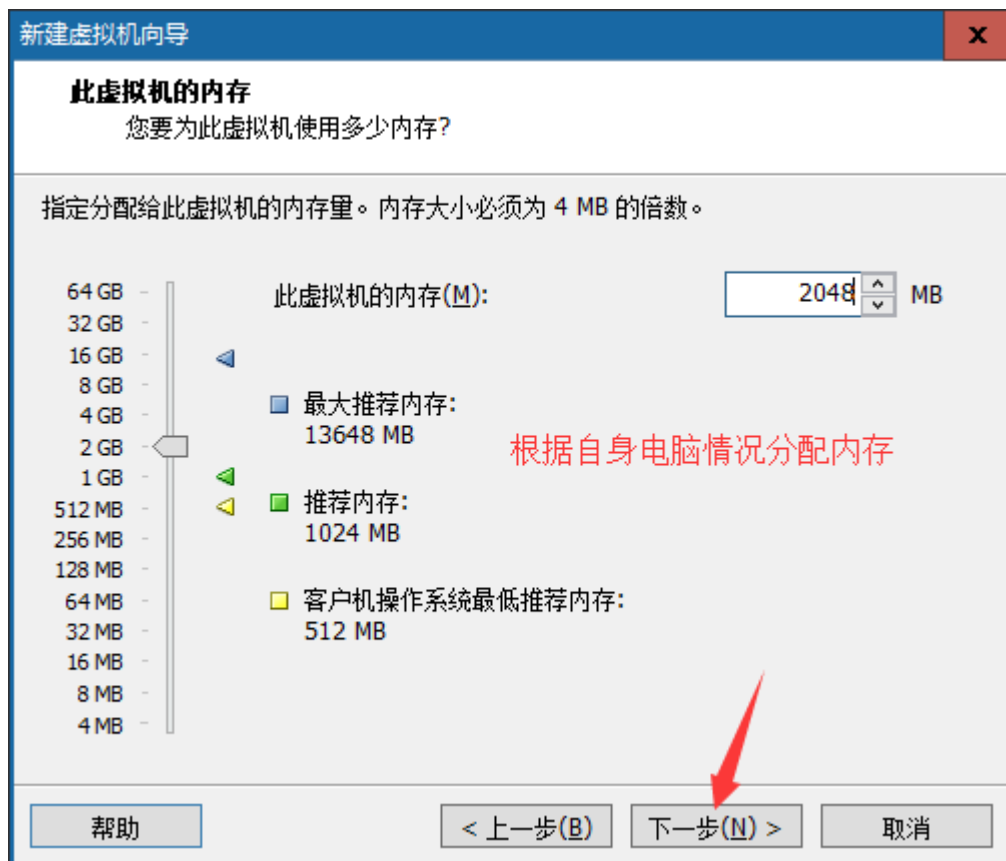
⑤选择虚拟机的名称（名称将会后期出现在左侧）和设置虚拟系统的安装位置



⑥cpu 设置



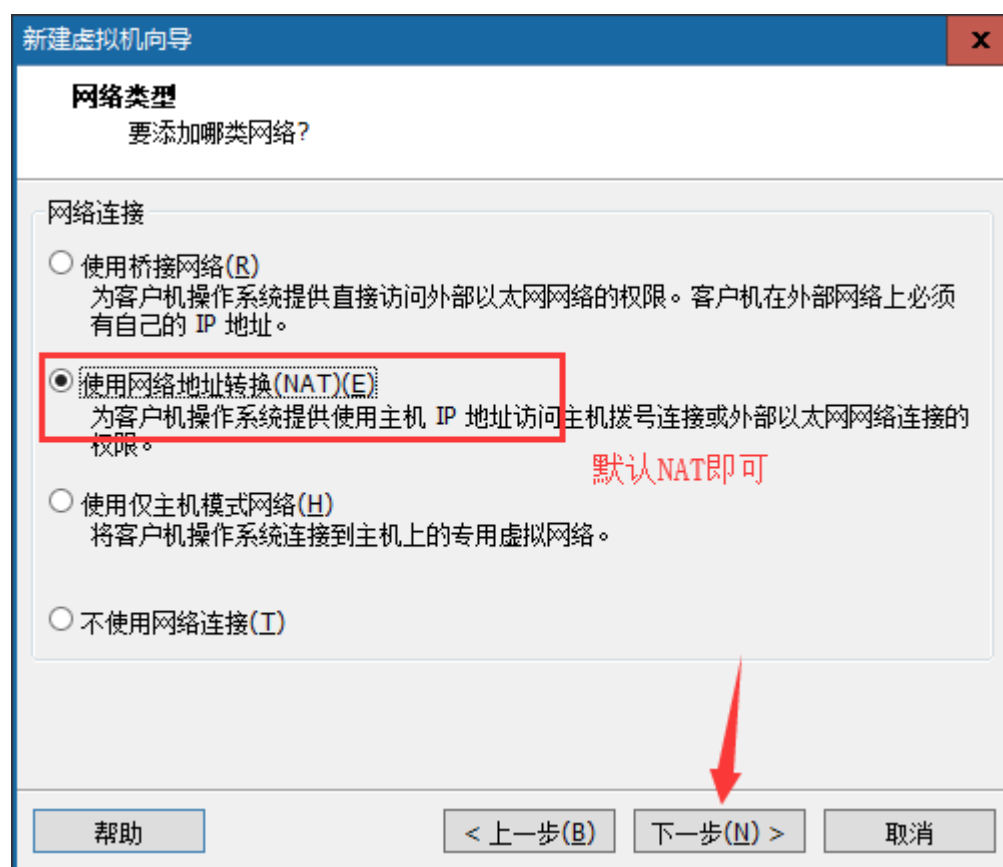
⑦分配内存



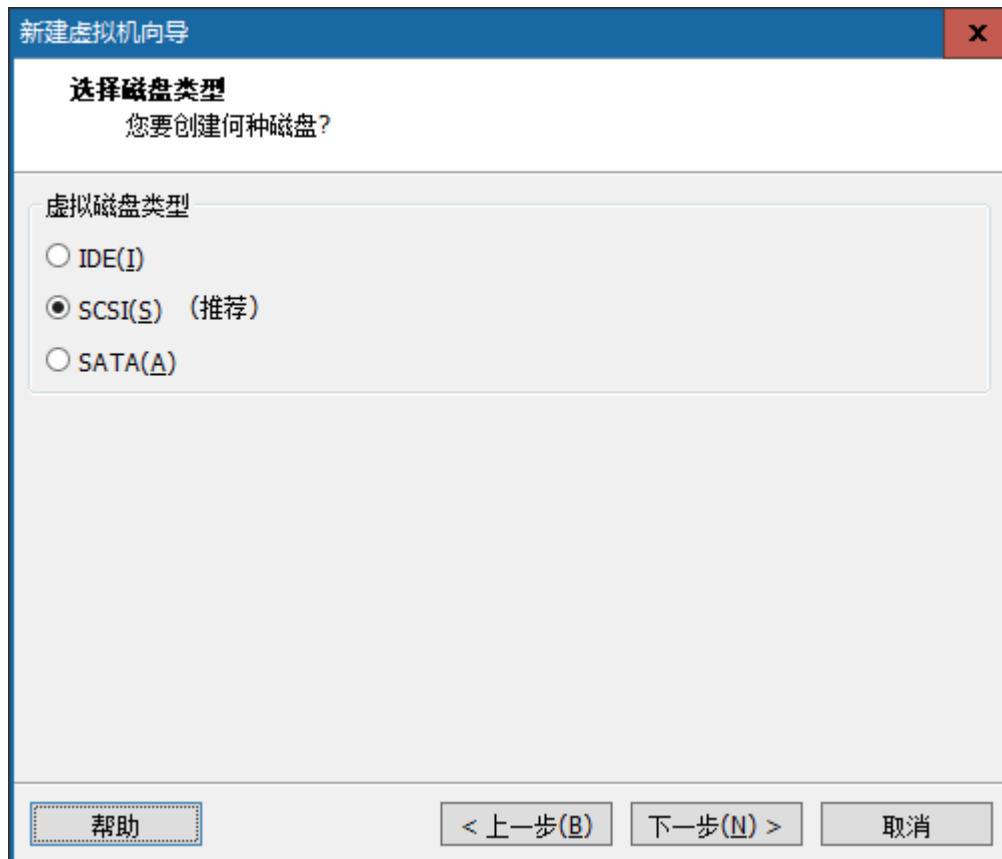
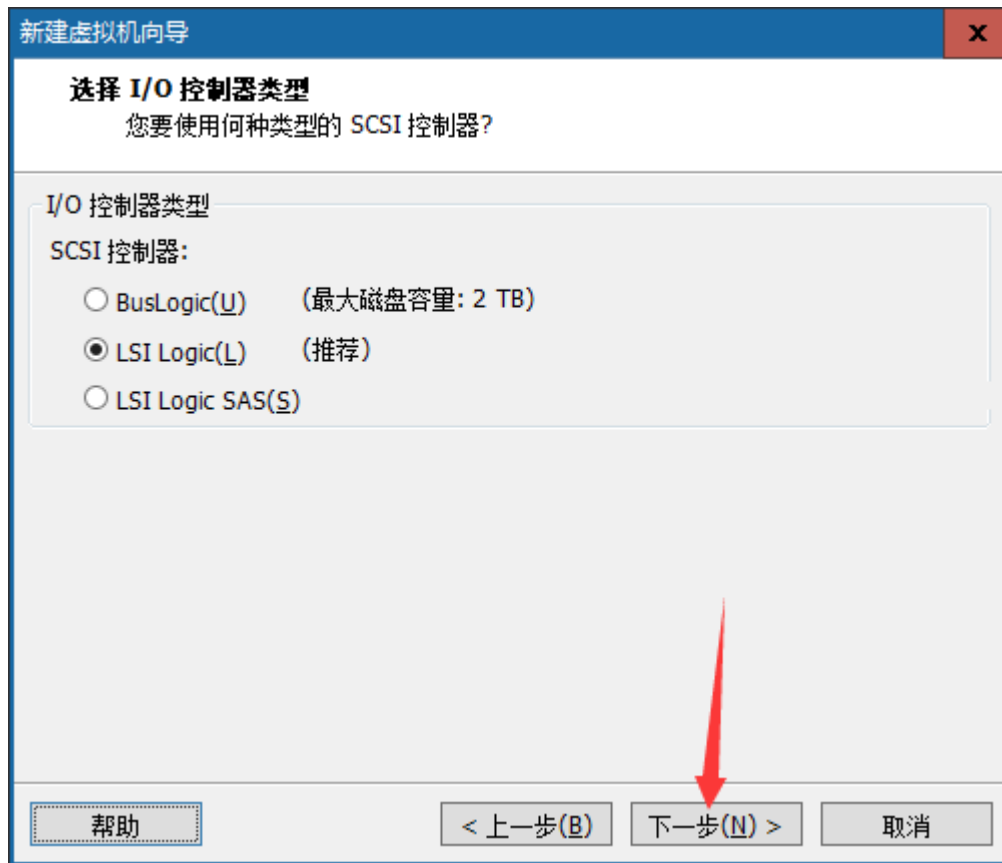
⑧选择网络类型，选择 nat 即可

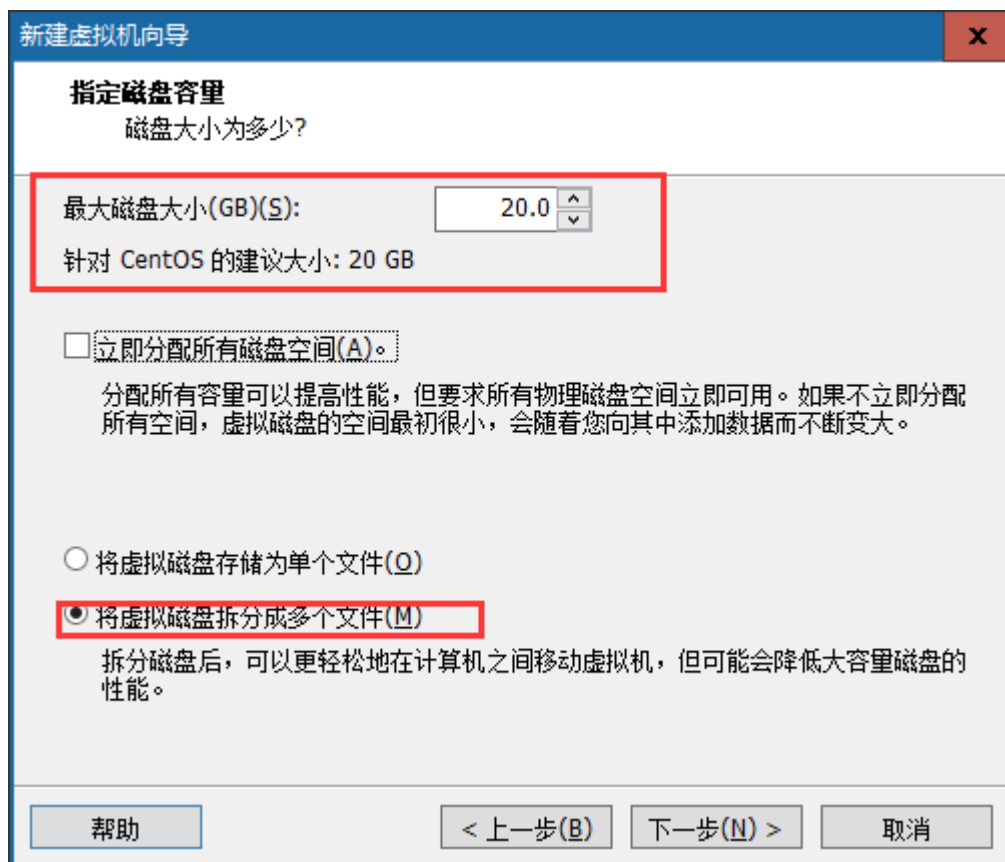
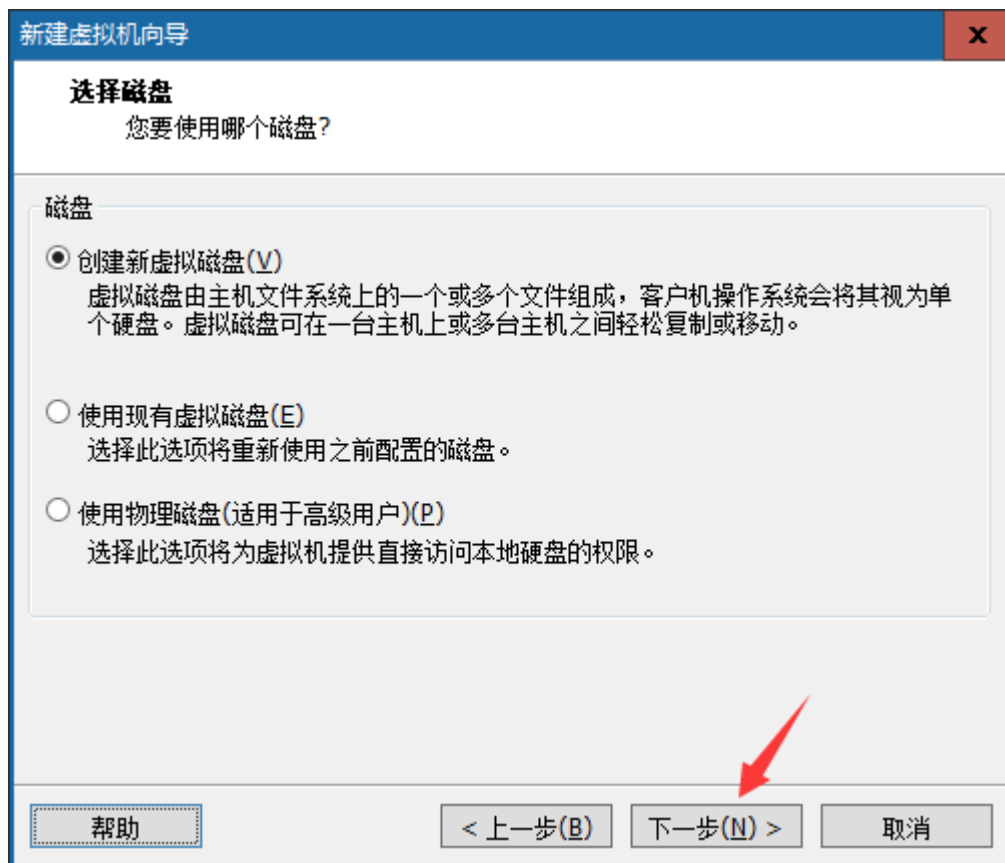
NAT：配置好之后 windows 即可和虚拟机进行互相通信，但是教室内的其他同学是访问不了的，只能自己可以访问虚拟机中的操作系统；

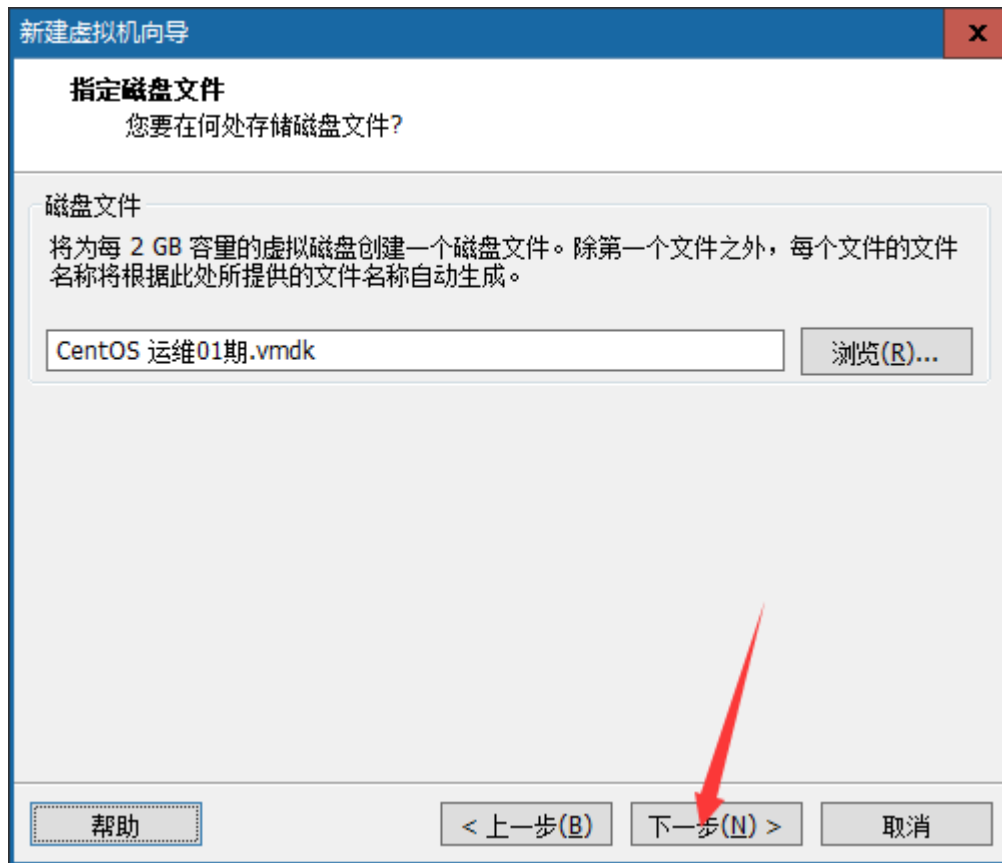
桥接：配置好之后其他同学也可以访问你的虚拟机操作系统；



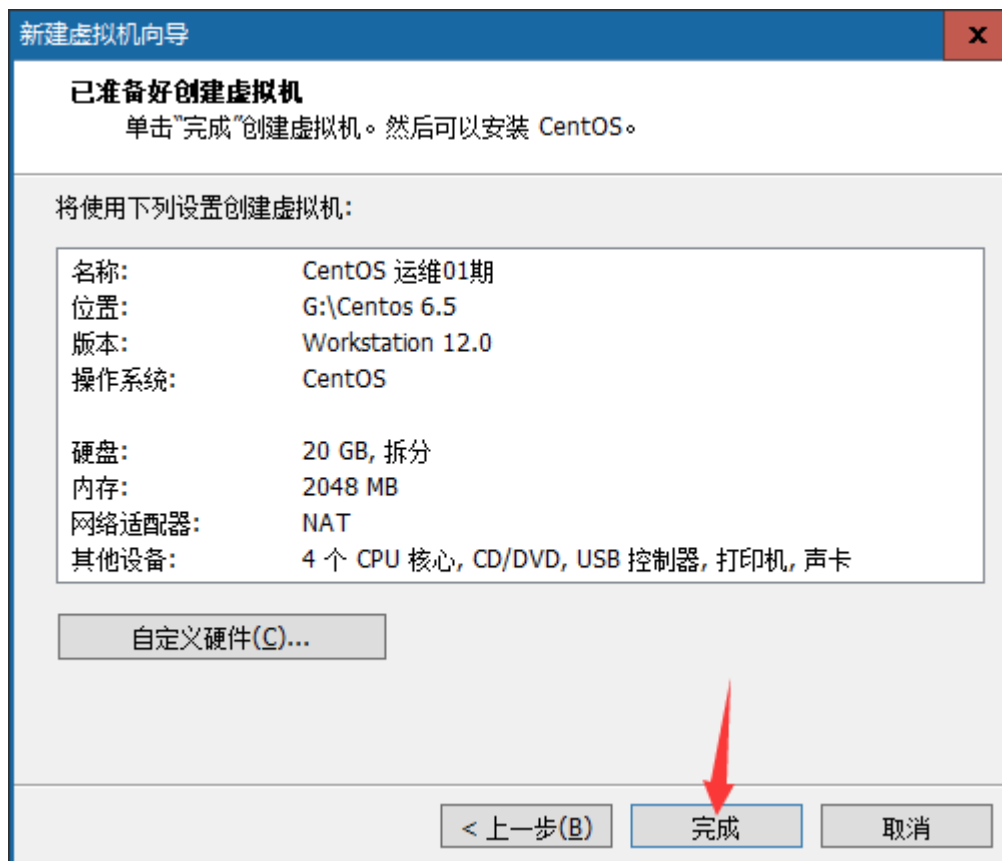
⑨后续默认的步骤，直接下一步





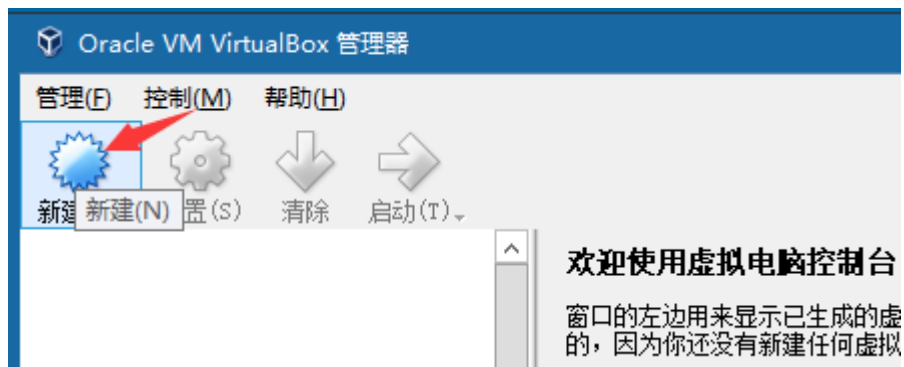


⑩ 点击完成

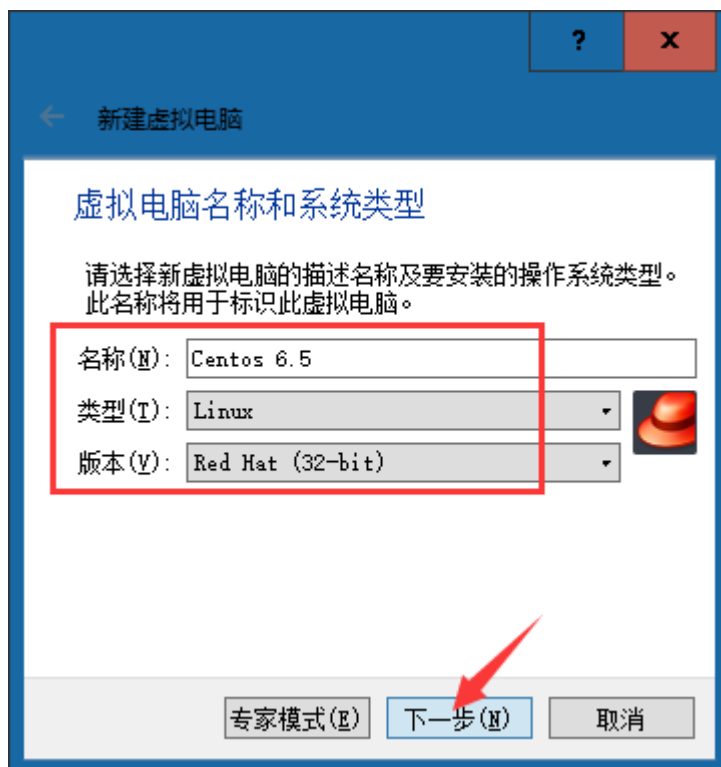


5.2、使用 Virtual Box （了解）

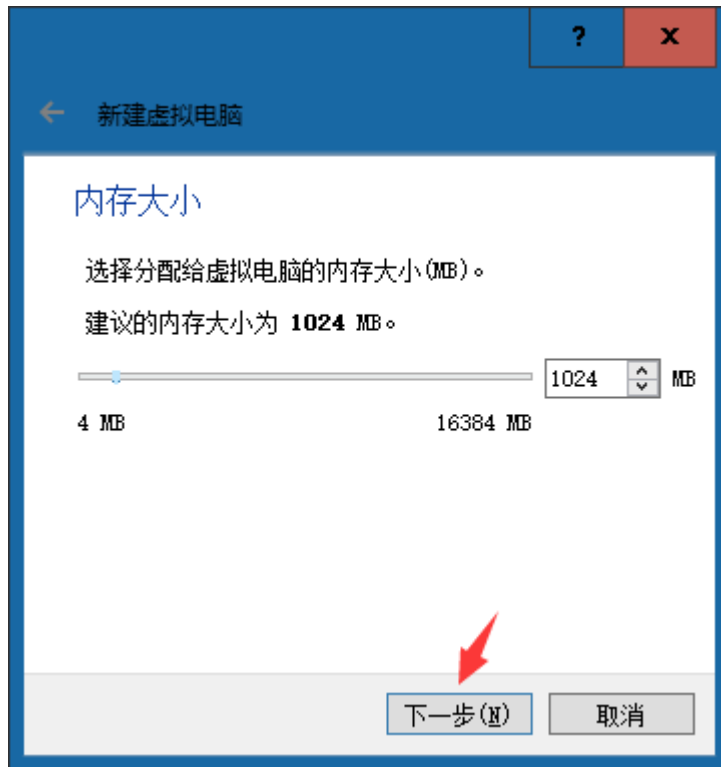
①点击左上角的新建按钮



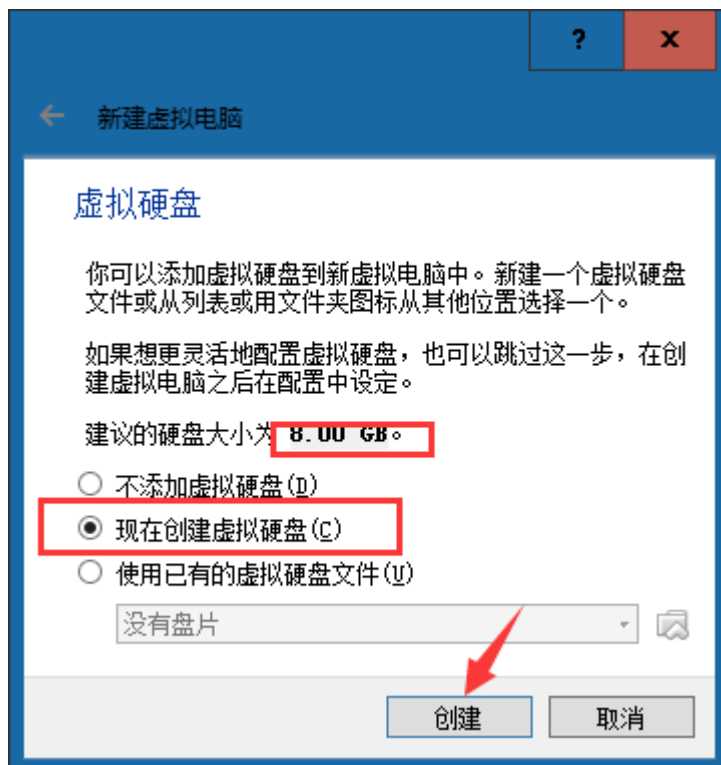
②选择操作系统，由于 centos 在列表中是没有的，所以选择 centos 对应的主线 redhat 32 位



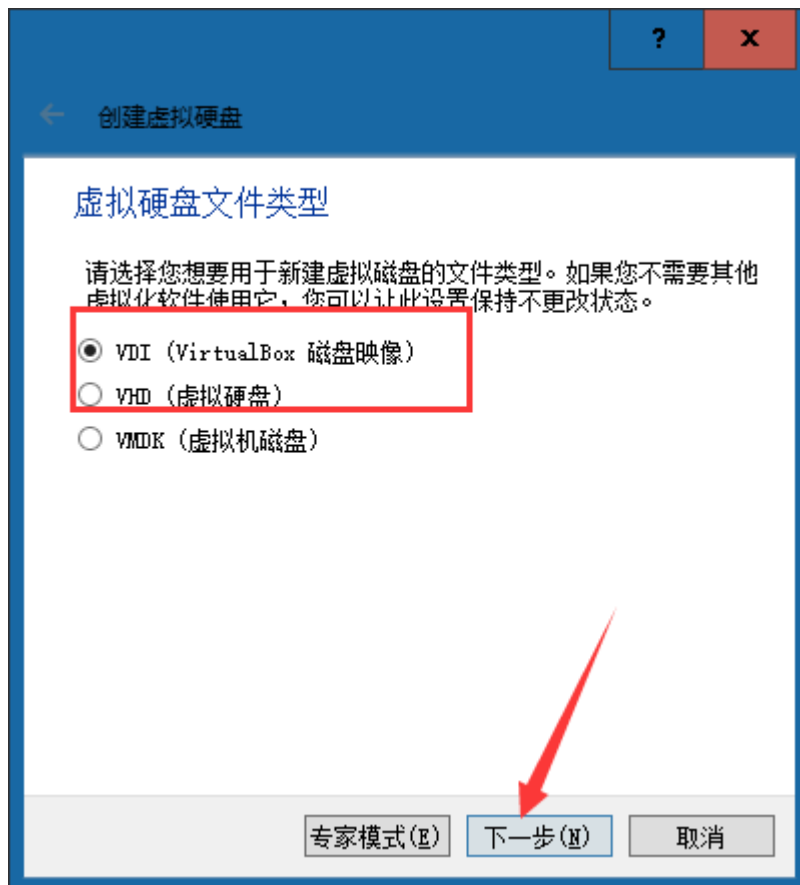
③选择内存大小



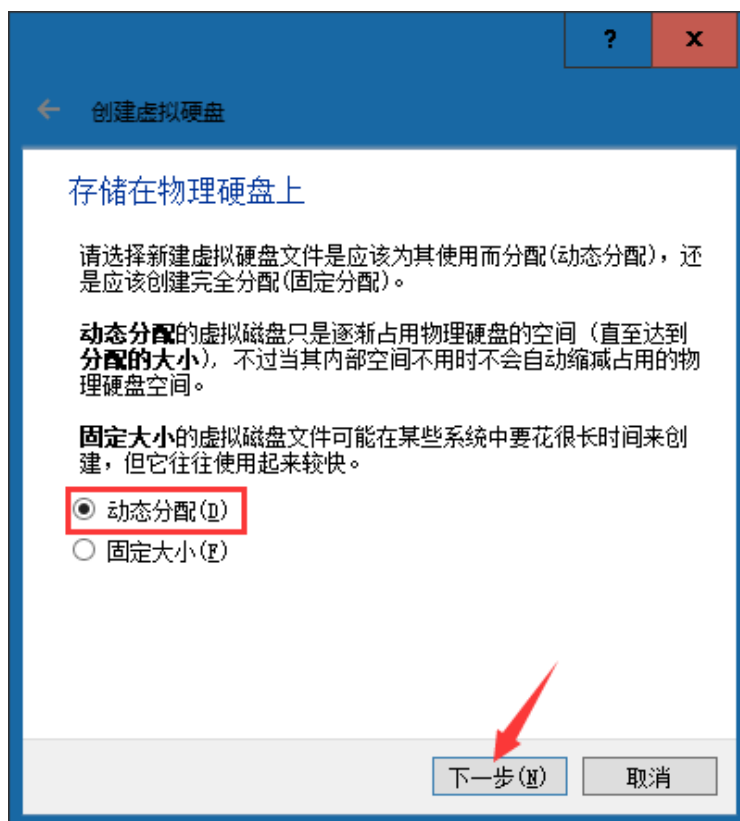
④创建磁盘，默认即可



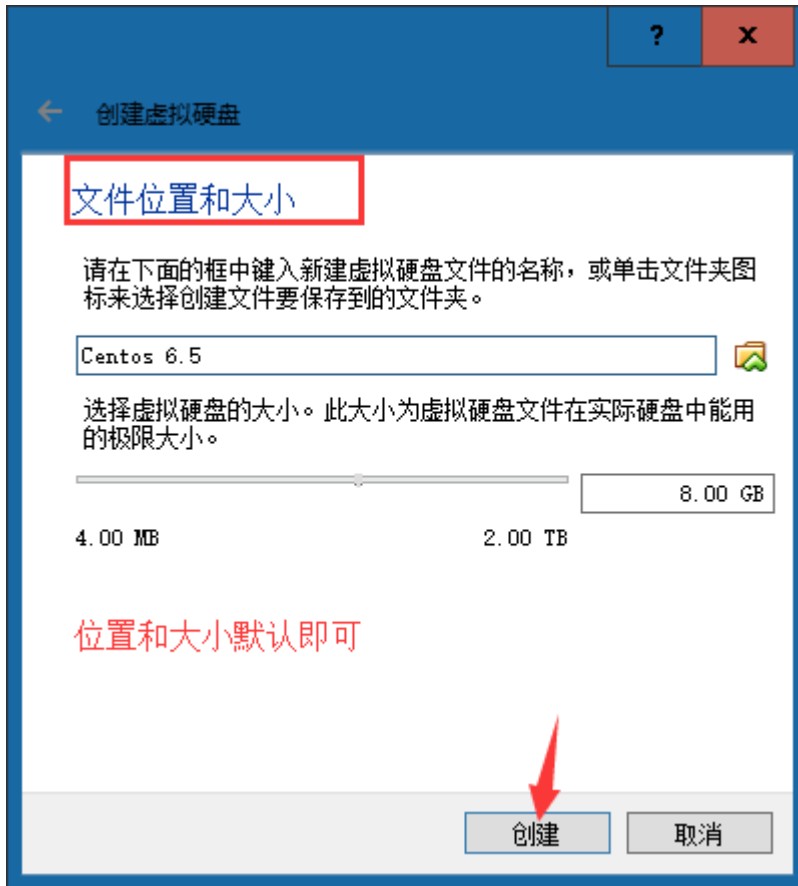
⑤选择磁盘类型，默认即可



⑥选择磁盘大小的分配方式，方式为动态分配即可



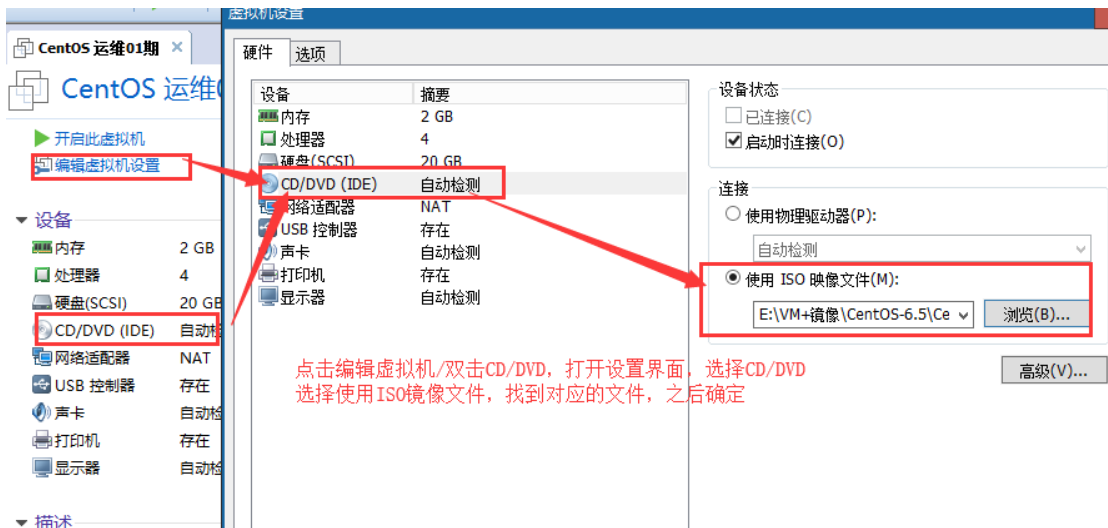
⑦设置磁盘的位置和大小确认



6、Linux 操作系统安装

6.1、使用 VMware workstation 进行安装

a. 由于之前没有指定 iso 镜像文件，因此此处需要先指定系统镜像文件



b. 运行此虚拟机



注意：如果开机之后鼠标点进去虚拟机出不来，则可以按下组合快捷键“ctrl+alt”。

如果启动之后出现类似提示框（不是错误框）则勾选不再提示，并且确定即可：

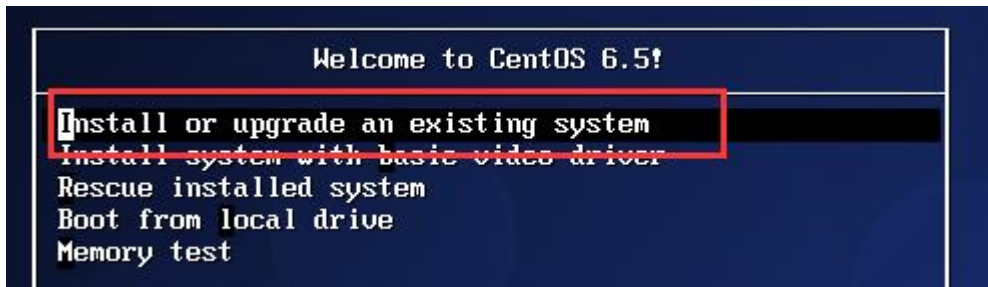


如果在启动时候出现下述错误，则说明电脑没有开启 cpu 的虚拟化，如果需要开启，则需要重启计算机，并且在开启的时候进入主板的 BIOS 设置开启虚拟化，然后保存设置重启电脑：





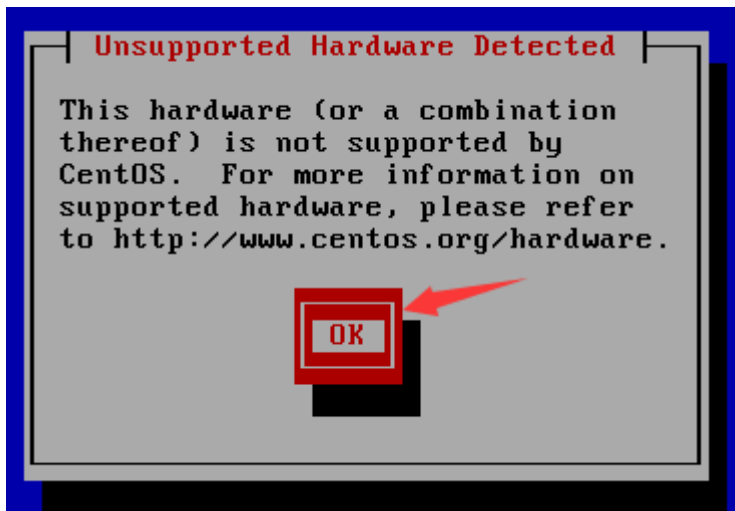
- c. 选择升级/安装已经存在的系统（通过↑/↓方向键）按下回车



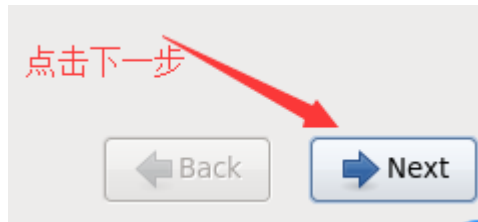
- d. 在检测到光盘（disc）之后选择跳过完整性检测直接进行安装



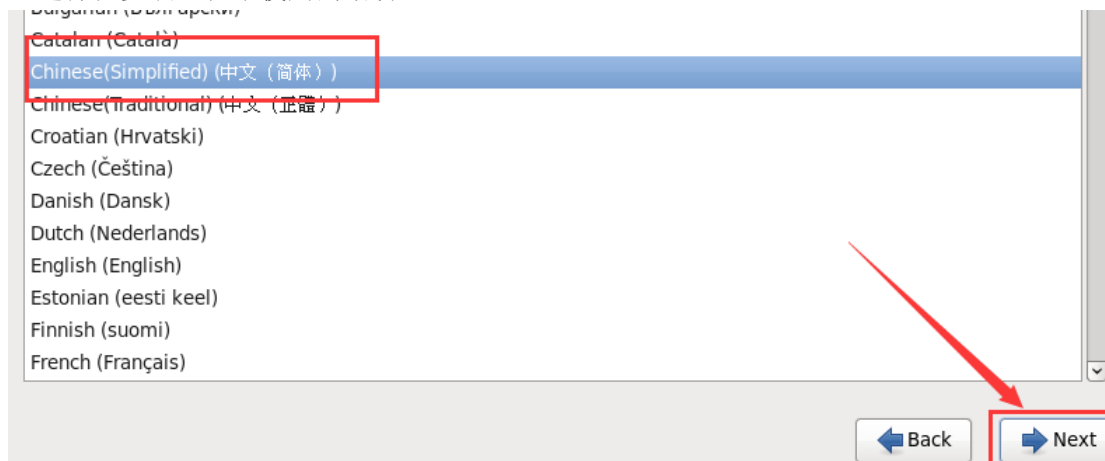
随后提示不支持的硬件，忽略直接下一步



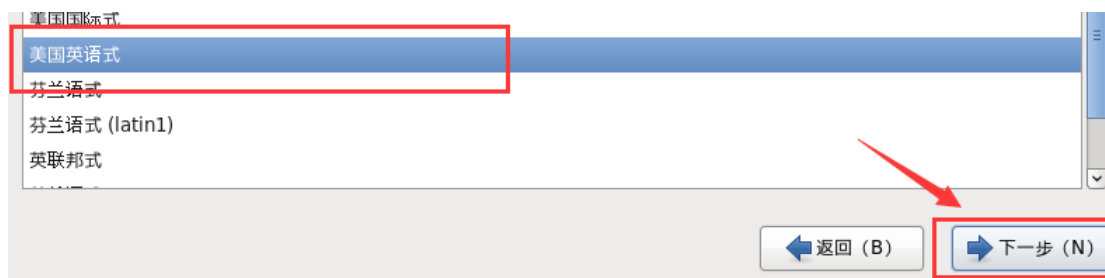
- e. 点击下一步



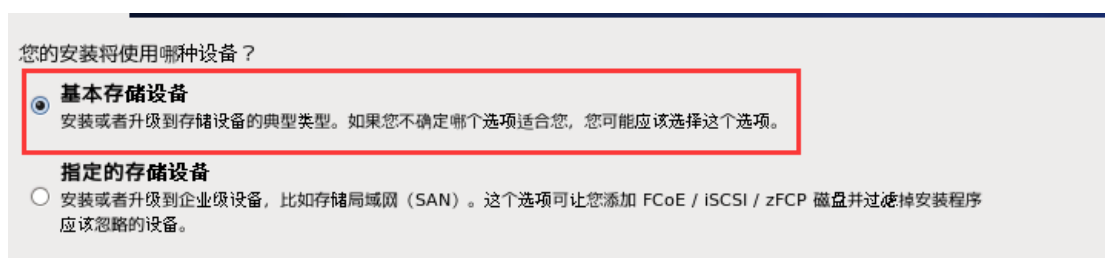
f. 选择在安装过程中使用的语言



g. 选择键盘类型，美国式英语



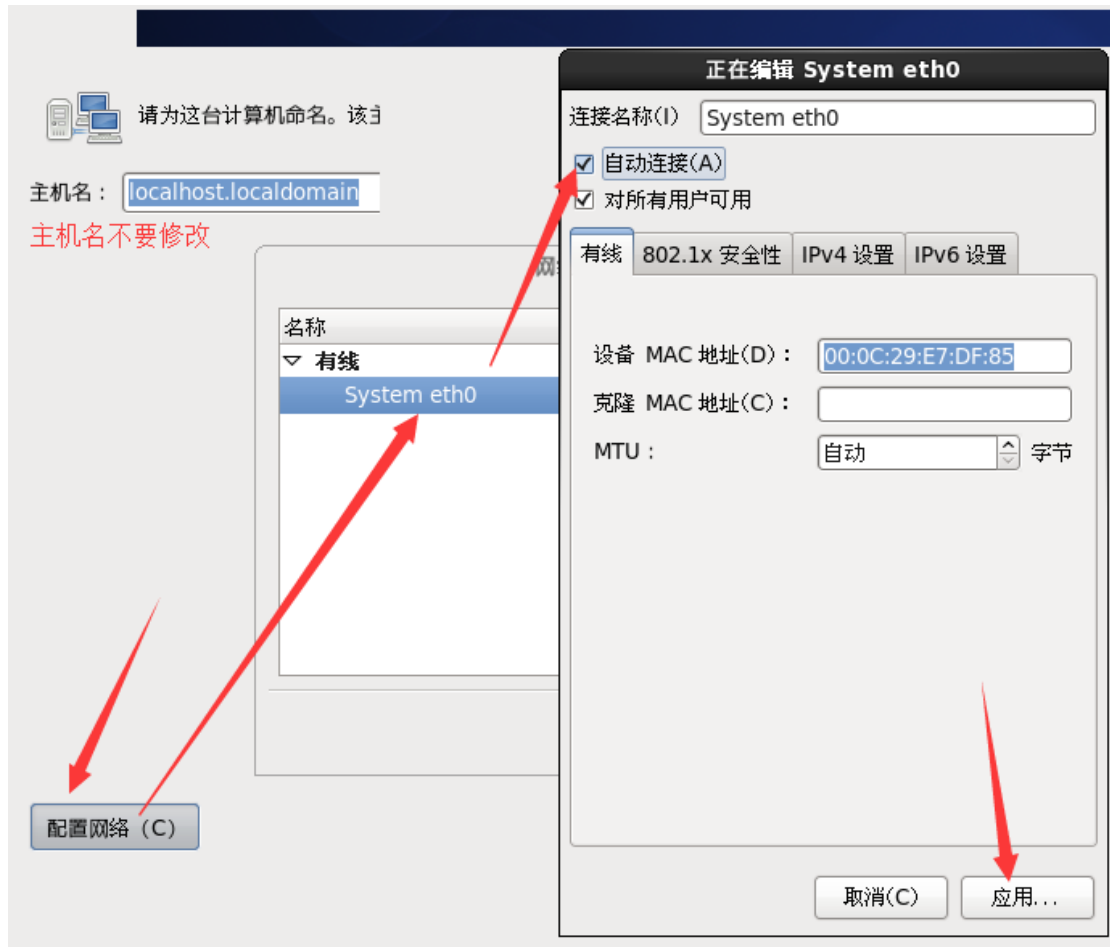
h. 选择存储设备类型



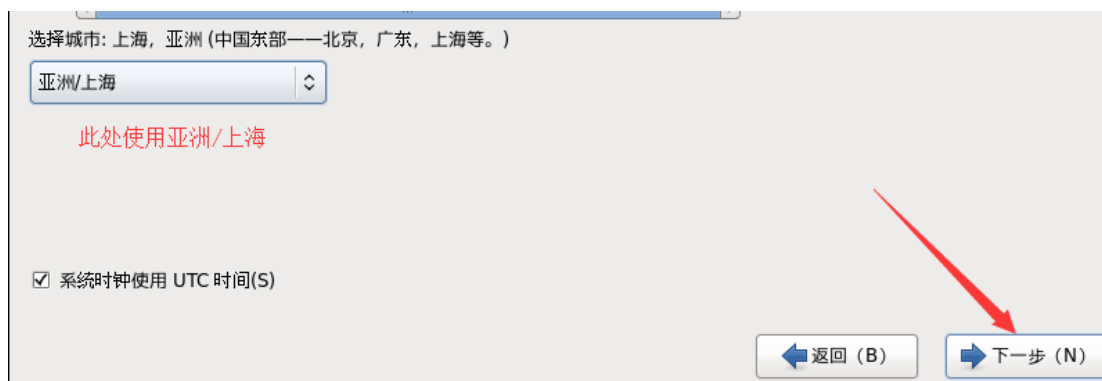
i. 对磁盘进行空白盘的初始化操作，选择“是，忽略所有的数据”



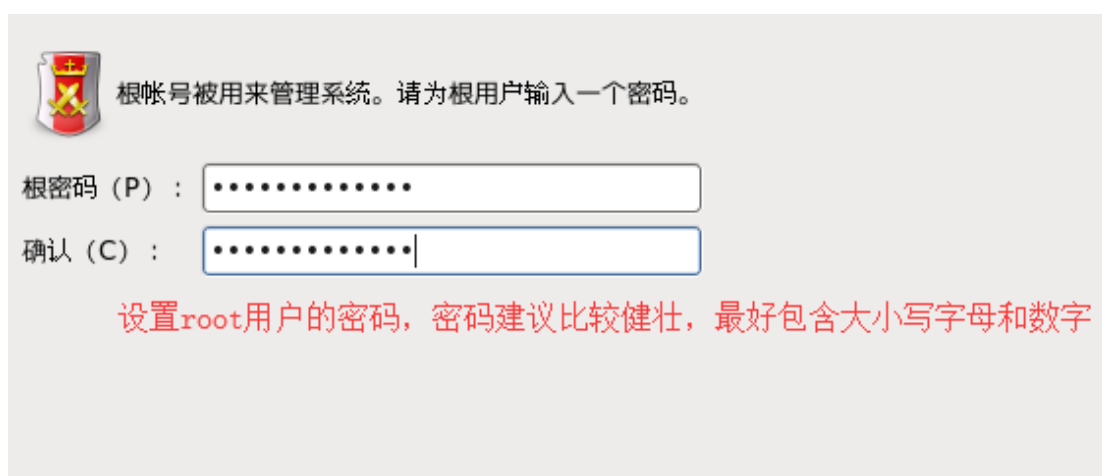
j. 设置网卡自动连接，依次应用 – 关闭 – 下一步



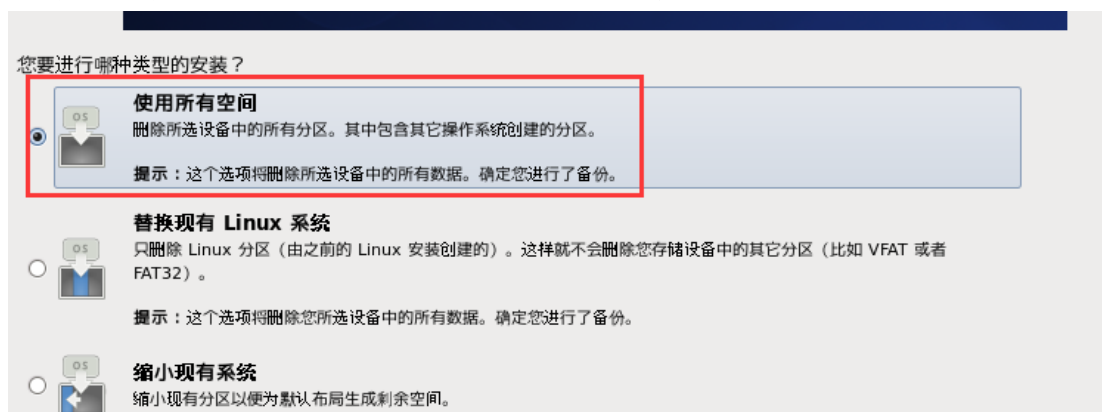
k. 设置时区，默认亚洲/上海



l. 设置密码，设置好了之后下一步

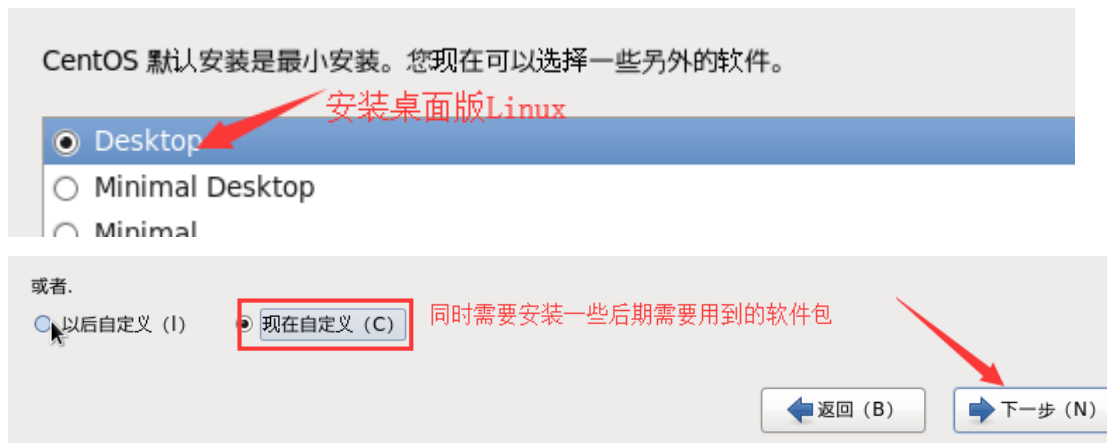


m. 使用全部的磁盘空间来安装 Linux 系统，点击下一步





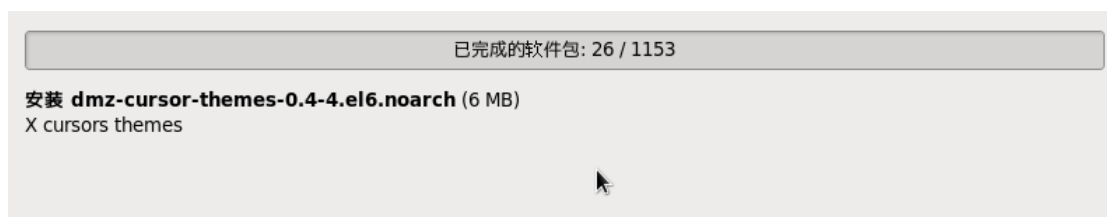
n. 选择安装的 Linux 类型



o. 选择开发 – 开发工具，前面复选框√，点击下一步



p. 等待软件包的安装



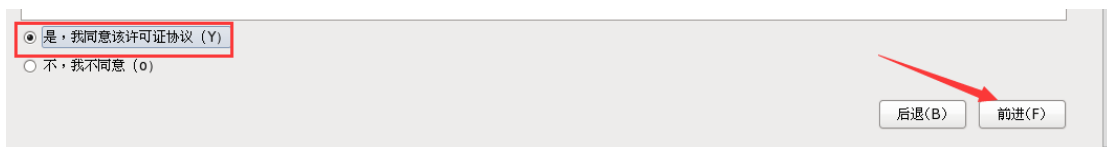
等待完成，点击重新引导



q. 重新引导之后点击“前进”



r. 在协议许可界面选择同意，然后点击前进



s. 创建普通用户帐号（可选），然后点击前进

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) :

创建的是普通用户，需要与root用户区分开

t. 时间设置，设置好之后前进

手动设置您系统的日期和时间

日期 (D)

< 2018 > < 三月 >

日	一	二	三	四	五	六
25	26	27	28	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

时间

时 (H) :

分 (M) :

秒 (S) :

后退(B)

前进(F)

u. 关于 kdump，之后点击完成

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) :

2022

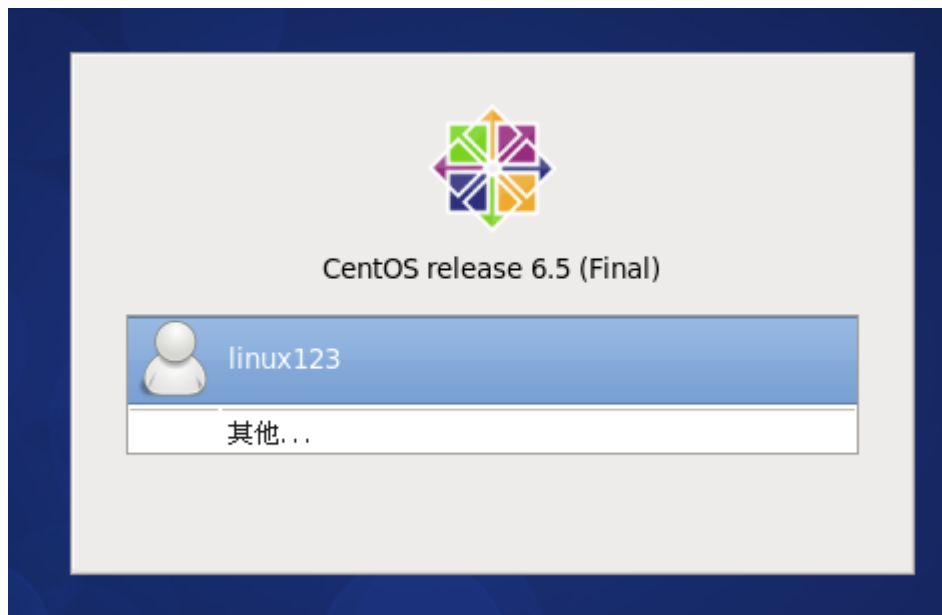
Kdump 内存 (MB) :

可用系统内存 (MB) :

1884

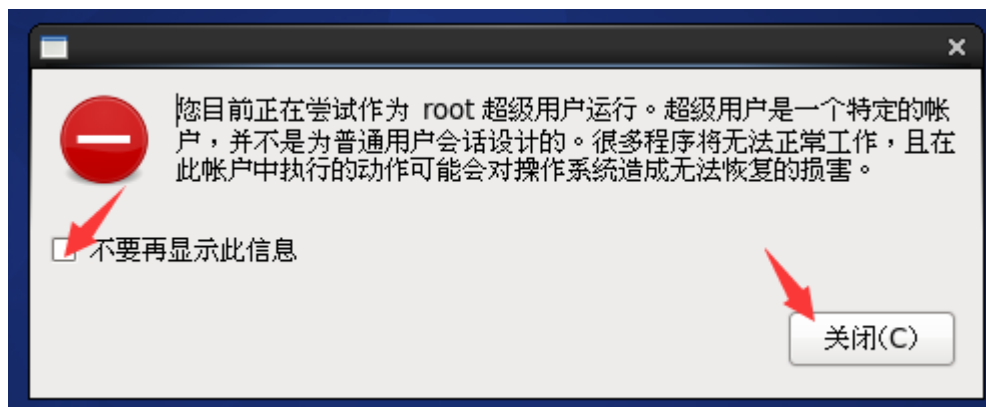
kdump是崩溃记录日志的工具，需要1G以上内存支持，如果内存不达标则会报错，告知无法启用

v. 登录界面



如果需要使用非列出的用户进行登录则点击其他，否则双击列出的用户名即可，随后输入密码。

w. 使用 root 帐号登录之后的提示

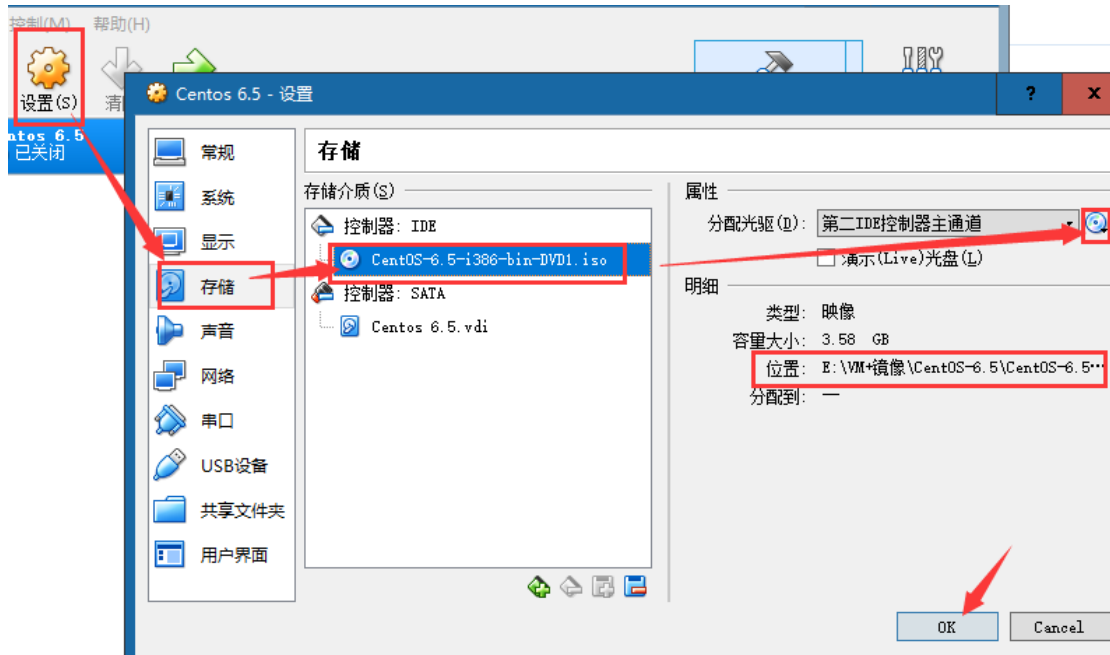


x. 看到的桌面



6.2、使用 virtual Box 安装 Linux（了解）

a. 选择需要安装的系统镜像



b. 启动虚拟机

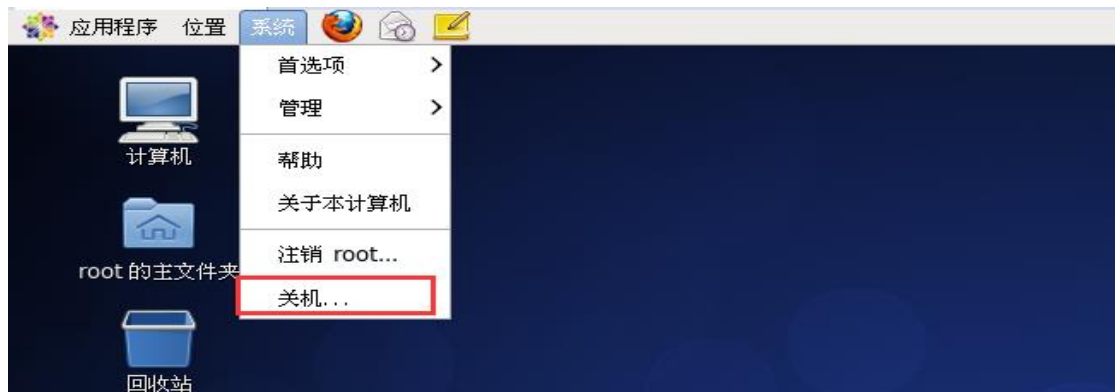


注意：如果鼠标在虚拟机中想退出到 windows，则需要按下 **ctrl+alt** 组合键（空格右侧的）

c. 后续全部操作按照上面 6.1 章节中的步骤继续安装即可。

7、终端

问题：在目前的桌面系统中，如果需要关机可以通过“系统”“关机”进行关机，那么后期服务器都是命令行模式的，届时这种方式将不好用，那要怎么关机呢？



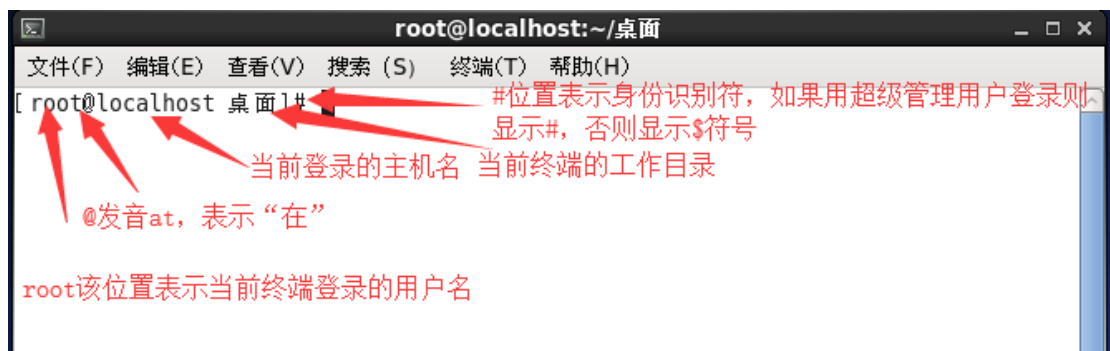
答：可以通过命令行方式进行关机。命令的输入需要在**终端**中进行输入。

所谓终端，其实类似于 windows 下 cmd 命令行模式。在终端中可以输入需要执行的一些指令，同样可以通过终端进行关机（注意：以后在工作中很少会去使用关机命令，会使用重启比较多）。

终端的形式：



终端组成部分：



如何使用终端命令进行关机？

在 Linux 中关机命令 有以下几个：shutdown -h now（正常关机）、halt（关闭内存）、init 0

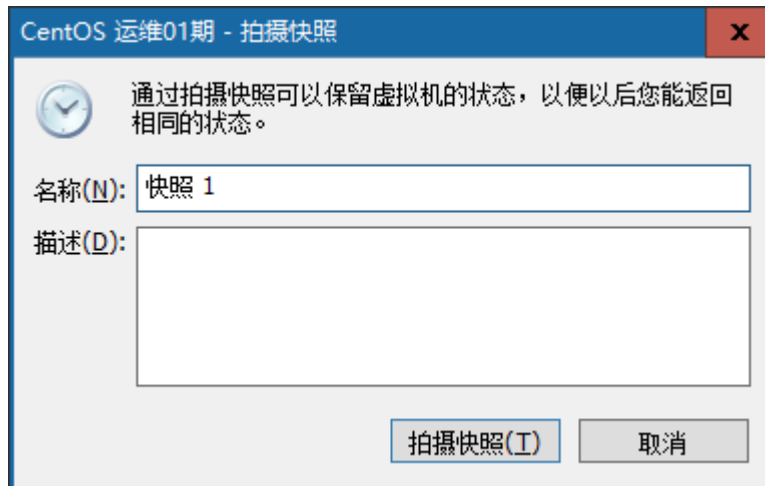


8、使用 VMware 备份操作系统

在 vm 中备份方式有 2 种：快照、克隆。

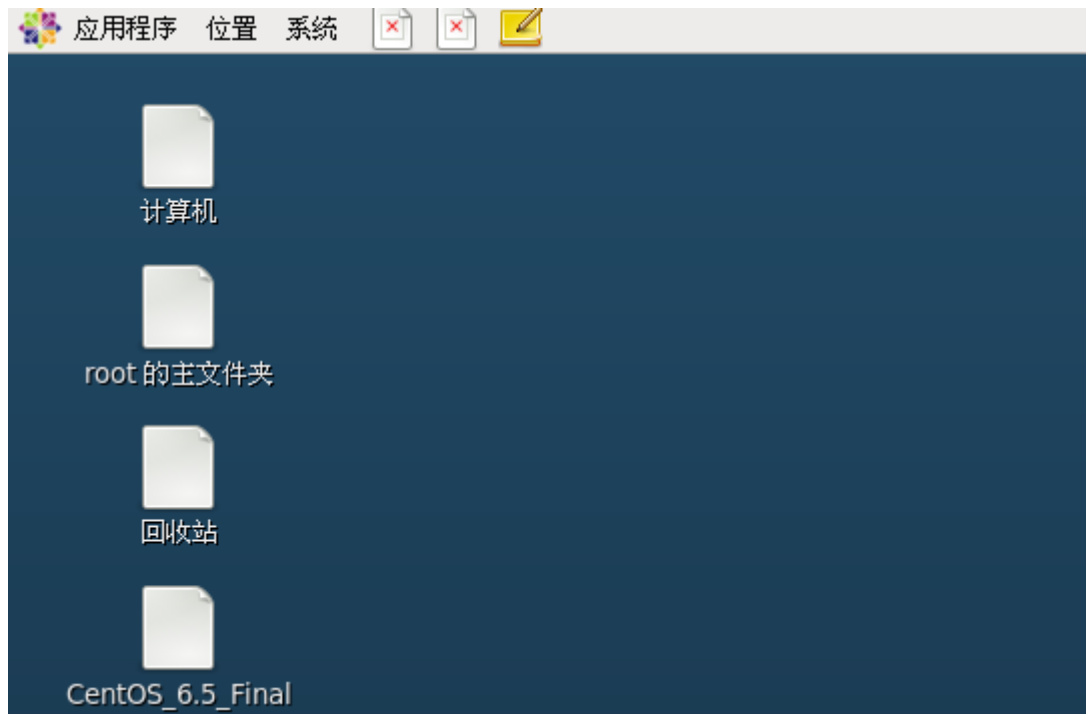
快照：又称还原点，就是保存在拍快照时候的系统的状态（包含了所有的内容），在后期的时候随时可以恢复。**【侧重于短期备份，需要频繁备份的时候可以使用快照，做快照的时候虚拟的操作系统一般处于开启状态】**

①在菜单“虚拟机” - “快照” - “拍摄快照”



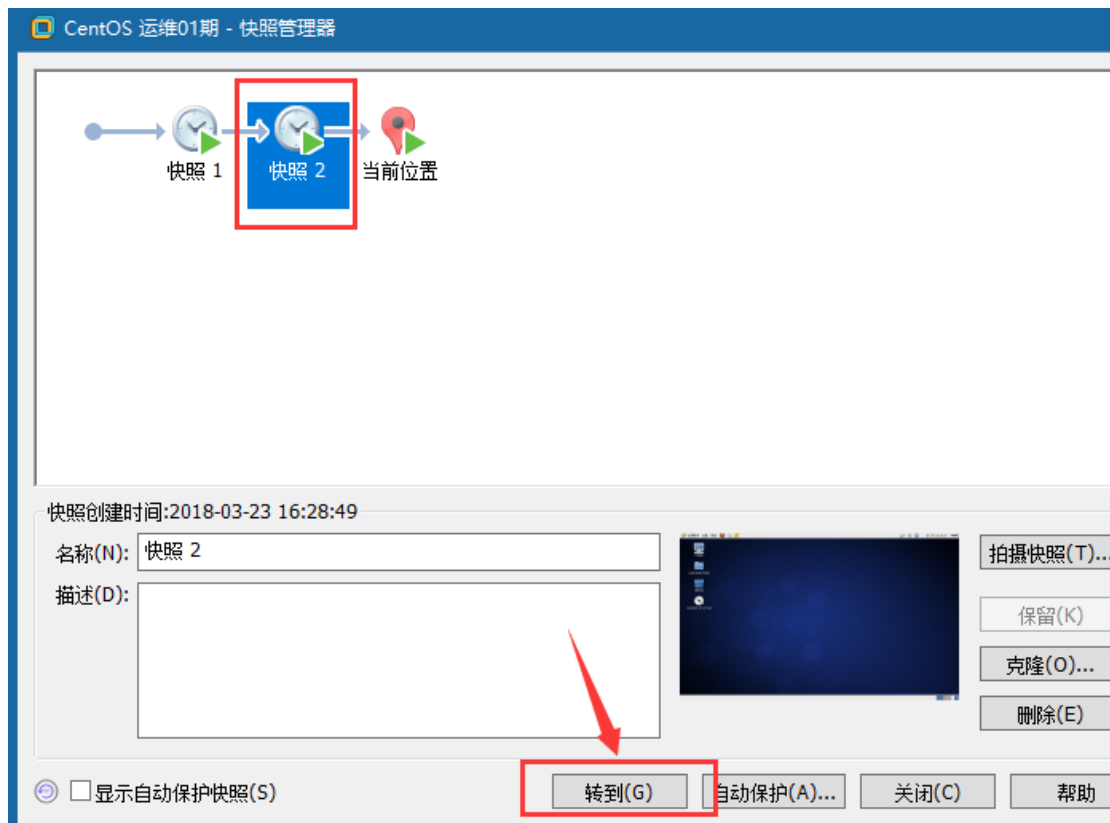
输入相关信息，点击拍摄快照

②搞事情

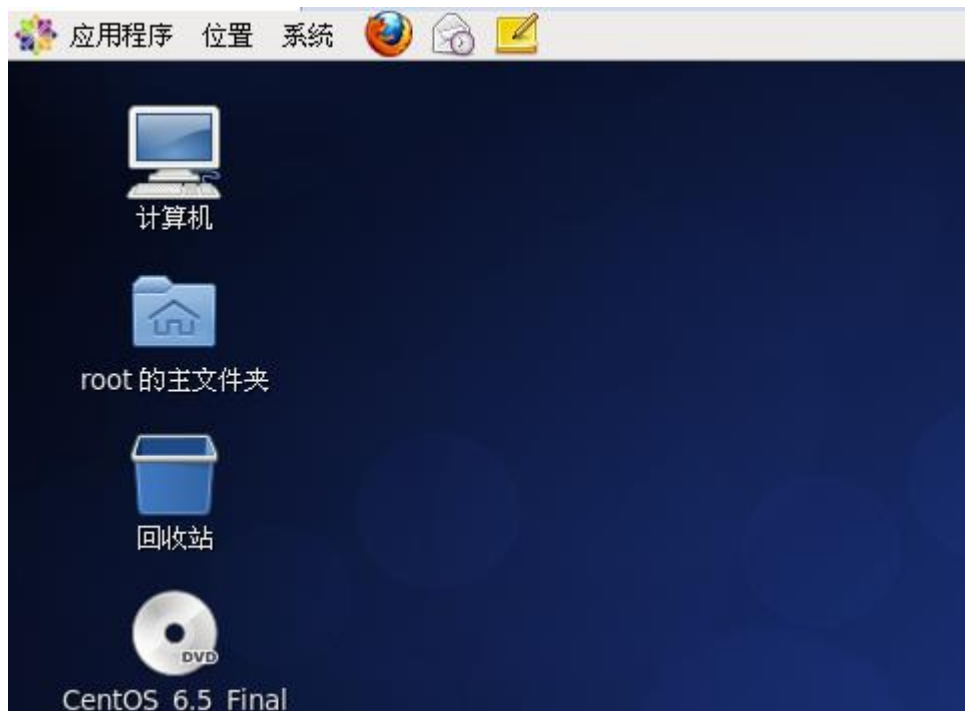


③使用快照恢复搞事情之前的状态

路径：虚拟机 - 快照 - 快照管理器

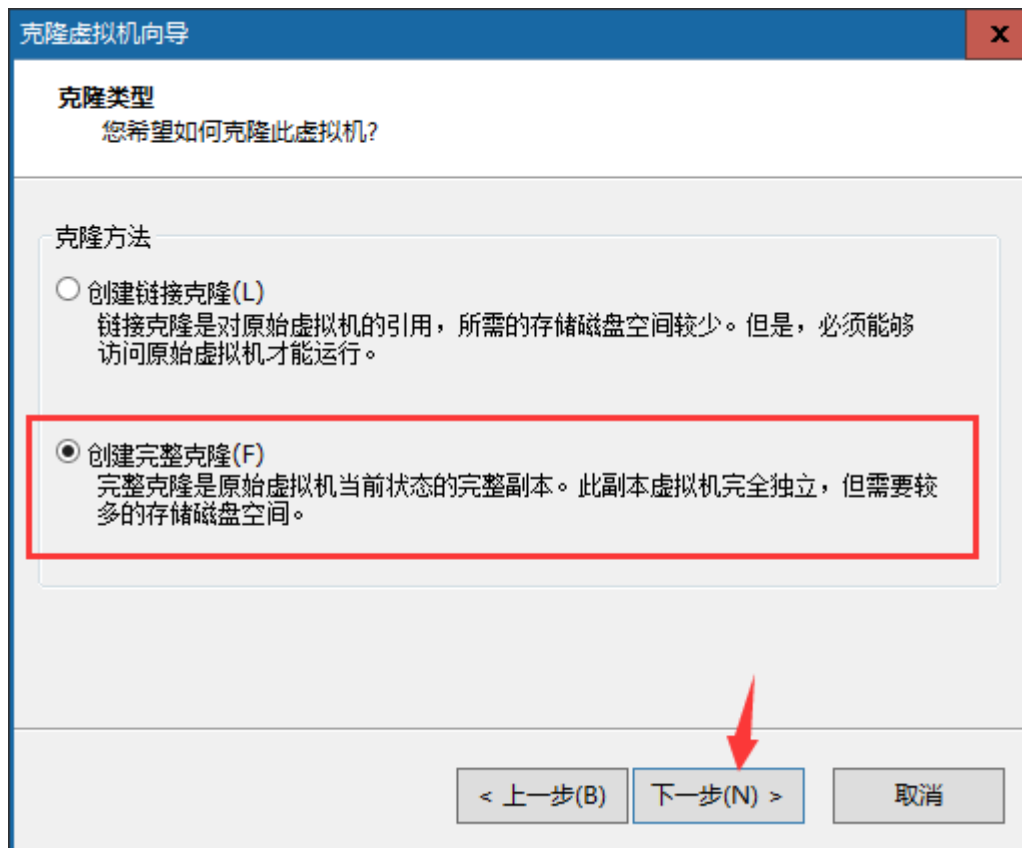
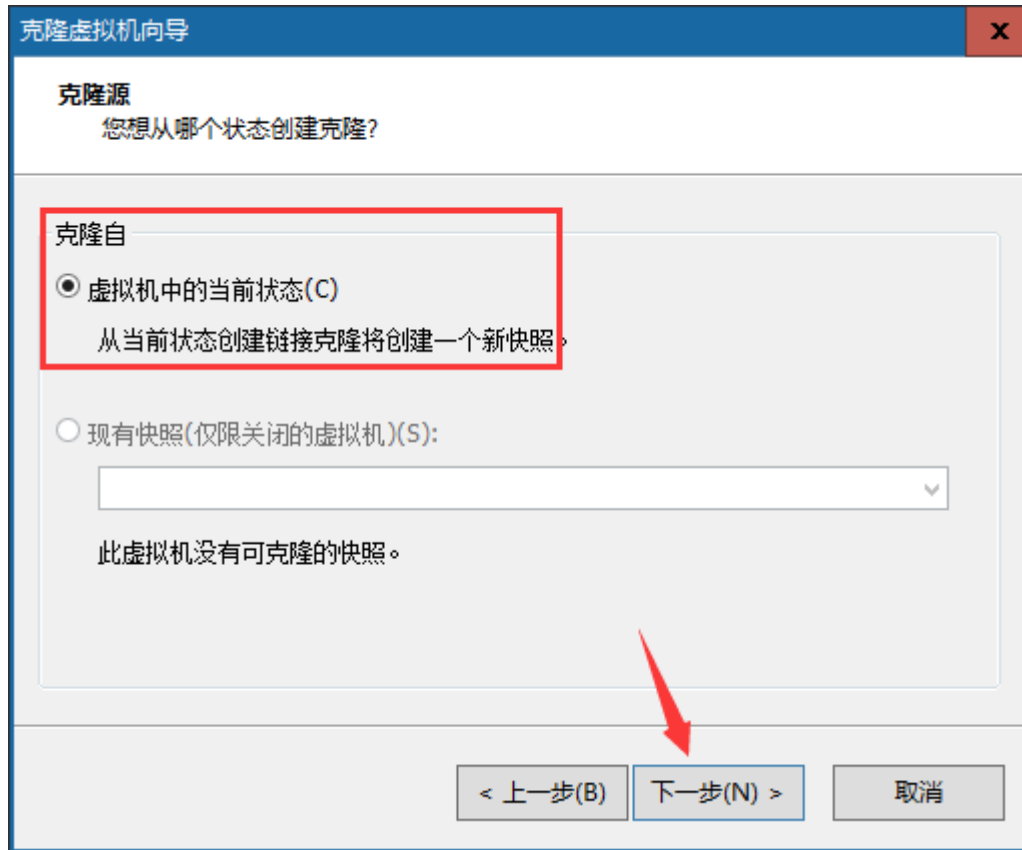


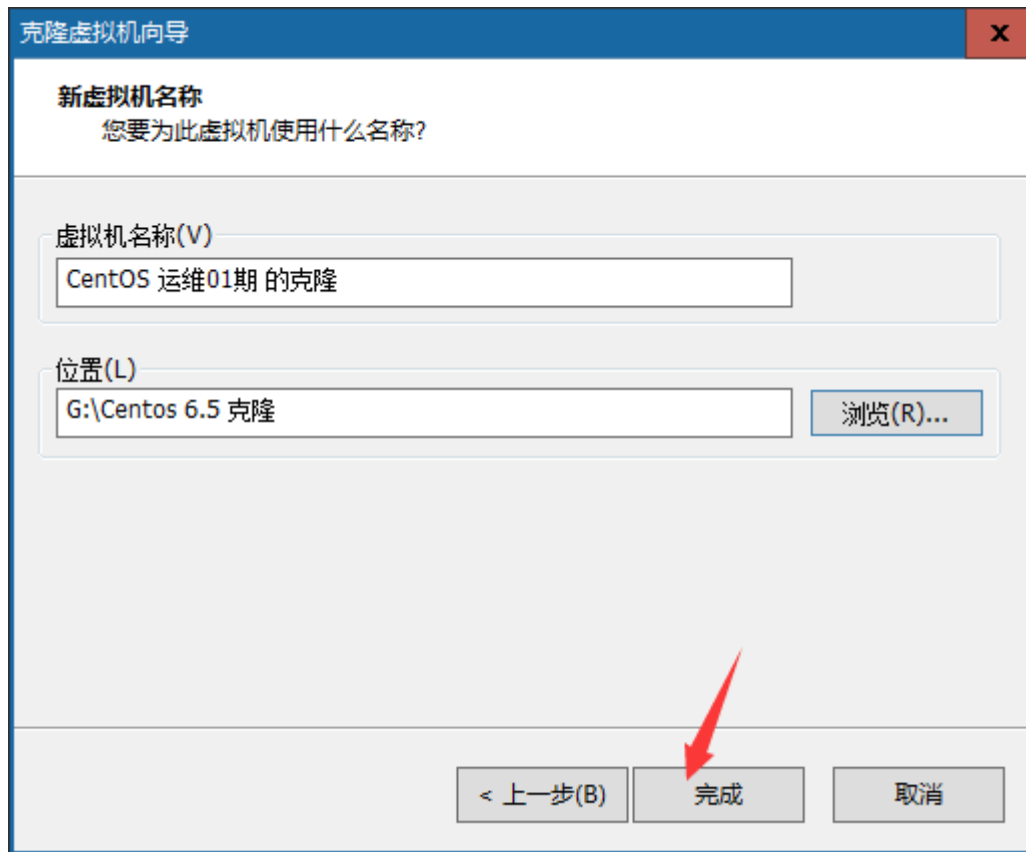
恢复好之后的状态：



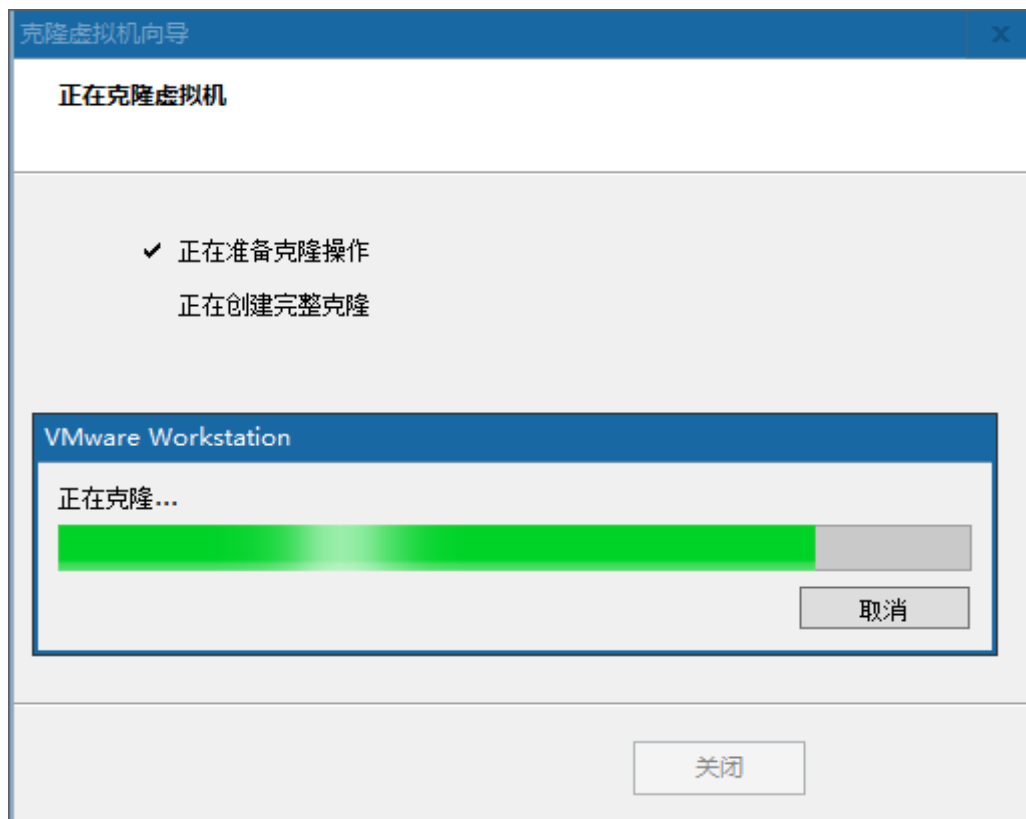
克隆：就是复制的意思。【侧重长期备份，做克隆的时候是必须得关闭】

路径：先关机 - 右键需要克隆的虚拟机 - 管理 - 克隆





上述的名称和位置与之前新建虚拟机的时候是一样的含义。
等待克隆完成





克隆好的服务器相关密码帐号等信息与被克隆的系统一致。

三、Linux 系统的文件

1、文件与文件夹（目录）


什么是文件？

一般都是一个独立的东西，可以通过一些特定的工具进行打开，并且其中不能包含除了文字以外的东西。例如：

 0x0404.ini	2013-08-19 14:55	配置
 0x0409.ini	2013-08-19 16:11	配置
 0x0804.ini	2013-08-19 14:55	配置

什么是文件夹？

可以在其中包含其他文件的东西。

名称	修改日期
 新建文件夹	2018-03-23 16

为什么先讲文件？

- 1:日常运维工作中，有近一半以上的工作内容 精力 其实都是对文件的操作。
- 2: Linux 本身也是一个基于文件形式表示的操作系统。

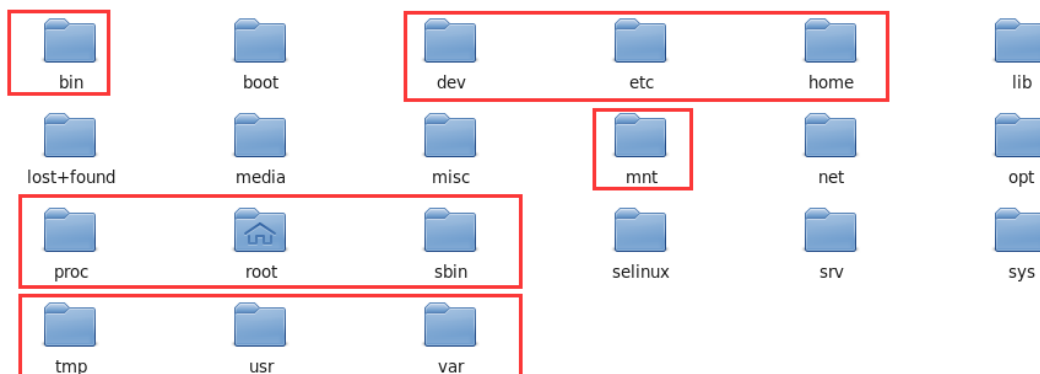
Linux 一切皆文件。

- ①在 windows 是文件的，在 Linux 下同样也是文件；
- ②在 windows 不是文件的，在 Linux 下也是以文件的形式存储的；

日常学习中和日常工作中，对于文件的操作的都有哪些种类？

创建文件、编辑文件、保存文件、关闭文件、重命名文件、删除文件、恢复文件。

2、Linux 系统的文件目录结构



目录结构:

Bin: 全称 **binary**, 含义是二进制。该目录中存储的都是一些二进制文件, 文件都是可以运行的。

Dev: 该目录中主要存放的是外接设备, 例如盘、其他的光盘等。在其中的外接设备是不能直接被使用的, 需要**挂载 (类似 windows 下的分配盘符)**。

Etc: 该目录主要存储一些配置文件。

Home: 表示“家”, 表示**除了 root 用户以外其他用户的家目录**, 类似于 windows 下的 User/ 用户目录。

Proc: **process**, 表示进程, 该目录中存储的是 Linux 运行时候的进程。

Root: 该目录是 root 用户自己的家目录。

Sbin: 全称 **super binary**, 该目录也是存储一些可以被执行的二进制文件, 但是必须得有 **super** 权限的用户才能执行。

Tmp: 表示“临时”的, 当系统运行时候产生的临时文件会在这个目录存着。

Usr: 存放的是用户自己安装的软件。类似于 windows 下的 **program files**。

Var: 存放的程序/系统的日志文件的目录。

Mnt: 当外接设备需要挂载的时候, 就需要挂载到 **mnt** 目录下。

Linux 的基本指令

一、指令与选项

什么是 Linux 的指令?

指在 Linux 终端 (命令行) 中输入的内容就称之为指令。

```
[root@localhost 桌面]# ggvhbjnkllkjh
bash: ggvhbjnkllkjh: command not found
[root@localhost 桌面]# █
```

一个完整的指令的标准格式: Linux 通用的格式

#指令主体 (空格) [选项] (空格) [操作对象]

一个指令可以包含多个选项

操作对象也可以是多个

例如: 需要让张三同学帮忙去楼下小卖铺买一瓶农夫山泉水和清风餐巾纸, 在这个指令中“买东西”是指令的主体, 买的水和餐巾纸是操作的对象, 农夫山泉、清风是操作的选项。



二、基础指令（重点）

1、ls 指令

含义：ls （list）

用法 1: #ls

含义：列出当前工作目录下的所有文件/文件夹的名称

```
[root@localhost ~]# ls
bin  dev  home  lost+found  misc  net  proc  sbin  srv  tmp  var
boot  etc  lib  media  mnt  opt  root  selinux  sys  usr  新文件
```

用法 2: #ls 路径

含义：列出指定路径下的所有文件/文件夹的名称

关于路径（重要）：

路径可以分为两种：相对路径、绝对路径。

相对路径：相对首先得有一个参照物（一般就是当前的工作路径）；

相对路径的写法：在相对路径中通常会用到 2 个符号“.”【表示当前目录下】、“..”【上一级目录下】。

绝对路径：绝对路径不需要参照物，直接从根“/”开始寻找对应路径；

```
[root@localhost ~]# ls /root
anaconda-ks.cfg  install.log.syslog  模板  图片  下载  桌面
install.log      公共的              视频  文档  音乐
```

用法 3: #ls 选项 路径

含义：在列出指定路径下的文件/文件夹的名称，并以指定的格式进行显示。

常见的语法：

#ls -l 路径

#ls -la 路径

选项解释：

-l：表示 list，表示以详细列表的形式进行展示

-a：表示显示所有的文件/文件夹（包含了隐藏文件/文件夹）

```
[root@localhost ~]# ls -l /root
总用量 100
-rw-----. 1 root root 1641 3月 23 23:17 anaconda-ks.cfg
-rw-r--r--. 1 root root 45503 3月 23 23:17 install.log
-rw-r--r--. 1 root root 10033 3月 23 23:16 install.log.syslog
drwxr-xr-x. 2 root root 4096 3月 23 15:32 公共的
drwxr-xr-x. 2 root root 4096 3月 23 15:32 模板
drwxr-xr-x. 2 root root 4096 3月 23 15:32 视频
drwxr-xr-x. 2 root root 4096 3月 23 15:32 图片
drwxr-xr-x. 2 root root 4096 3月 23 15:32 文档
drwxr-xr-x. 2 root root 4096 3月 23 15:32 下载
drwxr-xr-x. 2 root root 4096 3月 23 15:32 音乐
drwxr-xr-x. 2 root root 4096 3月 23 19:21 桌面
```

上述列表中的第一列字符表示文档的类型，其中“-”表示改行对应的文档类型为文件，“d”表示文档类型为文件夹。

```
[root@localhost ~]# ls -la /root
总用量 216
dr-xr-x---. 25 root root 4096 3月 24 08:58 .
dr-xr-xr-x. 25 root root 4096 3月 24 09:27 ..
drwxr-xr-x. 2 root root 4096 3月 23 15:33 .abrt
-rw-----. 1 root root 1641 3月 23 23:17 anaconda-ks.cfg
-rw-----. 1 root root 1995 3月 24 09:36 .bash_history
-rw-r--r--. 1 root root 18 5月 20 2009 .bash_logout
-rw-r--r--. 1 root root 176 5月 20 2009 .bash_profile
-rw-r--r--. 1 root root 176 9月 23 2004 .bashrc
drwxr-xr-x. 4 root root 4096 3月 23 17:03 .cache
```

在 Linux 中隐藏文档一般都是以“.”开头。

用法 4: #ls -lh 路径

含义：列出指定路径下的所有文件/文件夹的名称，以列表的形式并且在显示文档大小的时候以可读性较高的形式显示

参数含义：

```
[root@localhost ~]# ls -lh /root
总用量 100K
-rw-----. 1 root root 1.7K 3月 23 23:17 anaconda-ks.cfg
-rw-r--r--. 1 root root 45K 3月 23 23:17 install.log
-rw-r--r--. 1 root root 9.8K 3月 23 23:16 install.log.sysl
drwxr-xr-x. 2 root root 4.0K 3月 23 15:32 公共的
drwxr-xr-x. 2 root root 4.0K 3月 23 15:32 模板
drwxr-xr-x. 2 root root 4.0K 3月 23 15:32 视频
```

2、pwd 指令

用法：#pwd (print working directory, 打印当前工作目录)



```

[ root@localhost / ]# pwd
/
[ root@localhost / ]# █

[ root@localhost ~ ]# pwd
/root
[ root@localhost ~ ]# █

```

3、cd 指令

命令：#cd (change directory, 改变目录)

作用：用于切换当前的工作目录的

语法：**#cd 路径**

案例：当前在“/”下，需要使用绝对路径切换到/usr/local。

```

[ root@localhost / ]# cd /usr/local
[ root@localhost local ]# pwd
/usr/local
[ root@localhost local ]# █

```

案例：当前在/usr/local下，需要使用相对路径切换目录到home目录下的Linux123用户家目录中去。

```

[ root@localhost local ]# cd ../../home/linux123
[ root@localhost linux123 ]# pwd
/home/linux123
[ root@localhost linux123 ]# █

```

补充：

在Linux中有一个特殊的符号“~”，表示当前用户的家目录。

切换的方式：#cd ~

```

[ root@localhost linux123 ]# cd ~
[ root@localhost ~ ]# pwd
/root
[ root@localhost ~ ]# █

```

4、mkdir 指令

指令：mkdir (make directory, 创建目录)

语法 1: **#mkdir 路径** 【路径，可以是文件夹名称也可以是包含名称的一个完整路径】

案例：在当前路径下创建出目录“yunweihenniu”

```
[ root@localhost /]# mkdir yunweihenniu
[ root@localhost /]# ls
bin    etc    lost+found  mnt  proc  selinux  tmp    yunweihenniu
boot  home  media       net  root  srv      usr    新文件
dev    lib    misc       opt  sbin  sys      var
```

注意：ls 列出的结果颜色说明，其中蓝色的名称表示文件夹，黑色的表示文件，绿色的其权限为拥有所有权限。

案例：在指定路径下创建一个文件夹“yunweihenniu”

```
[ root@localhost /]# mkdir /root/yunweihenniu
[ root@localhost /]# ls ~
anaconda-ks.cfg  install.log.syslog  公共的  视频  文档  音乐
install.log      yunweihenniu       模板   图片  下载  桌面
```

语法 2: #mkdir -p 路径

含义：当一次性创建多层不存在的目录的时候，添加-p 参数，否则会报错

```
[ root@localhost /]# mkdir /root/yunweihenniu/a/b/c/d
mkdir: 无法创建目录"/root/yunweihenniu/a/b/c/d": 没有那个文件或目录
[ root@localhost /]# mkdir -p /root/yunweihenniu/a/b/c/d
[ root@localhost /]# ls ~/yunweihenniu/
a
[ root@localhost /]# ls ~/yunweihenniu/a
b
[ root@localhost /]# ls ~/yunweihenniu/a/b
c
[ root@localhost /]# ls ~/yunweihenniu/a/b/c
d
```

语法 3: #mkdir 路径 1 路径 2 路径 3.... 【表示一次性创建多个目录】

```
[ root@localhost ~]# mkdir a b c
[ root@localhost ~]# ls
a    b  install.log  yunweihenniu  模板  图片  下载  桌面
anaconda-ks.cfg  c  install.log.syslog  公共的  视频  文档  音乐
```

5、touch 指令

指令：touch

作用：创建文件

语法：#touch 文件路径 【路径可以是直接的文件名也可以是路径】

案例：使用 touch 来在当前路径下创建一个文件，命名为 Linux.txt

```
[ root@localhost ~]# touch linux.txt
[ root@localhost ~]# ls
a    b  install.log  linux.txt  公共的  视频  文档  音乐
anaconda-ks.cfg  c  install.log.syslog  yunweihenniu  模板  图片  下载  桌面
[ root@localhost ~]#
```



案例：使用 touch 来同时创建多个文件

```
[root@localhost ~]# touch linux1.txt linux2.txt
[root@localhost ~]# ls
a          c          linux1.txt  yunweihenniu  视频  下载
anaconda-ks.cfg  install.log  linux2.txt  公共的        图片  音乐
b          install.log.syslog  linux.txt   模板          文档  桌面
[root@localhost ~]#
```

案例：使用 touch 来在“Linux123”用户的家目录中创建文件，Linux.txt

```
[root@localhost ~]# touch /home/linux123/linux.txt
[root@localhost ~]# ls /home/linux123/
linux.txt  公共的  模板  视频  图片  文档  下载  音乐  桌面
[root@localhost ~]#
```

6、cp 指令

指令：cp (copy, 复制)

作用：复制文件/文件夹到指定的位置

语法：**#cp 被复制的文档路径 文档被复制到的路径**

案例：使用 cp 命令来复制一个文件

```
[root@localhost ~]# cp linux1.txt /home/linux123/linux1.txt
[root@localhost ~]# cp linux1.txt /home/linux123/linux10.txt
[root@localhost ~]# ls /home/linux123/
linux10.txt  linux.txt  模板  图片  下载  桌面
linux1.txt   公共的    视频  文档  音乐
[root@localhost ~]#
```

注意：Linux 在复制过程中是可以重新对新位置的文件进行重命名的，但是如果不是必须的需要，则建议保持前后名称一致。

案例：使用 cp 命令来复制一个文件夹

注意：当使用 cp 命令进行文件夹复制操作的时候需要添加选项“-r”【-r 表示递归复制】，否则目录将被忽略

```
[root@localhost ~]# cp ./yunweihenniu/ /home/linux123/yunweihenniu
cp: 略过目录"./yunweihenniu/"
[root@localhost ~]# cp -r ./yunweihenniu/ /home/linux123/yunweihenniu
[root@localhost ~]#
```

7、mv 指令

指令：mv (move, 移动, 剪切)

作用：移动文档到新的位置

语法：**#mv 需要移动的文档路径 需要保存的位置路径**

确认：移动之后原始的文件还在不在原来的位置？原始文件是不在原始位置的

案例：使用 mv 命令移动一个文件

```
[root@localhost ~]# mv linux1.txt /linux1.txt
[root@localhost ~]# ls /
bin  etc  linux1.txt  misc  opt  sbin  sys  var
boot home lost+found mnt  proc selinux tmp  yunweihenniu
dev  lib  media      net   root  srv   usr  新文件
```

案例：使用 mv 命令移动一个文件夹

```
[root@localhost ~]# mv /home/linux123/yunweihenniu/ /
[root@localhost ~]# ls /
bin  etc  linux1.txt  misc  opt  sbin  sys  var
boot home lost+found mnt  proc selinux tmp  yunweihenniu
dev  lib  media      net   root  srv   usr  新文件
```

补充：在 Linux 中重命名的命令也是 mv，语法和移动语法一样。

```
[root@localhost ~]# mv /yunweihenniu /henniu
[root@localhost ~]# ls /
bin  etc  lib  media  net  root  srv  usr
boot henniu linux1.txt misc  opt  sbin  sys  var
dev  home  lost+found mnt  proc selinux tmp  新文件
```

8、rm 指令

指令：rm (remove, 移除、删除)

作用：移除/删除文档

语法：#rm 选项 需要移除的文档路径

选项：

- f: force, 强制删除, 不提示是否删除
- r: 表示递归

案例：删除一个文件

```
[root@localhost ~]# rm /新文件
rm: 是否删除普通空文件 "/新文件"? y
[root@localhost ~]# ls /
bin  etc  lib  media  net  root  srv  usr
boot henniu linux1.txt misc  opt  sbin  sys  var
dev  home  lost+found mnt  proc selinux tmp
```

在删除的时候如果不带选项，会提示是否删除，如果需要确认则输入“y/yes”，否则输入



“n/no” 按下回车。

注意：如果在删除的时候不想频繁地确认，则可以在指令中添加选项“-f”，表示 force（强制）。

```
[root@localhost ~]# rm -f /linux1.txt
[root@localhost ~]# ls /
bin  dev  hennix  lib  media  mnt  opt  root  selinux  sys  usr
boot  etc  home  lost+found  misc  net  proc  sbin  srv  tmp  var
```

案例：删除一个文件夹

```
[root@localhost ~]# rm /hennix/
rm: 无法删除"/hennix/": 是一个目录
[root@localhost ~]# rm -r /hennix/
rm: 是否进入目录"/hennix"? y
rm: 是否删除普通空文件 "/hennix/linux1.txt"? n
rm: 是否进入目录"/hennix/a"? n
[root@localhost ~]# rm -rf /hennix/
```

注意：删除一个目录的时候需要做递归删除，并且一般也不需要进行删除确认询问，所以移除目录的时候一般需要使用-rf 选项。

案例：删除多个文档

```
[root@localhost ~]# rm -rf a linux.txt
[root@localhost ~]# ls
anaconda-ks.cfg  install.log  linux2.txt  模板  文档  桌面
b                install.log.syslog  yunweihennix  视频  下载
c                linux1.txt~  公共的  图片  音乐
```

案例：要删除一个目录下有公共特性的文档，例如都以 Linux 开头

```
[root@localhost ~]# rm -f linux*
[root@localhost ~]# ls
anaconda-ks.cfg  c  install.log.syslog  公共的  视频  文档  音乐
b                install.log  yunweihennix  模板  图片  下载  桌面
```

其中*称之为通配符，意思表示任意的字符，Linux*，则表示只要文件以 Linux 开头，后续字符则不管。

9、vim 指令

指令：vim （vim 是一款文本编辑器）

语法：**#vim 文件的路径**

作用：打开一个文件（可以不存在，也可以存在）

案例：使用 vim 来打开文件

退出打开的文件：在没有按下其他命令的时候，按下 shift+英文冒号，输入 q，按下回车即

```
可  
安装 libcap-ng-0.6.4-3.el6_0.1.i686  
:q
```

10、输出重定向

一般命令的输出都会显示在终端中，有些时候需要将一些命令的执行结果想要保存到文件中进行分析/统计，则这时候需要使用到的输出重定向技术。

>: **覆盖**输出，会覆盖掉原先的文件内容

>>: **追加**输出，不会覆盖原始文件内容，会在原始内容末尾继续添加

语法: #正常执行的指令 >/>> 文件的路径

注意: 文件可以不存在，不存在则新建

案例: 使用覆盖重定向，保存 ls -la 的执行结果，保存到当前目录下的 ls.txt

```
[root@localhost ~]# ls -la > ls.txt
```

案例: 使用追加重定向，保存 ls -la 的执行结果到 ls.txt 中

```
[root@localhost ~]# ls -la >> ls.txt  
[root@localhost ~]# vim ls.txt
```

11、cat 指令

作用 1: cat 有直接打开一个文件的功能。

语法 1: #cat 文件的路径

```
安装 strace-4.5.19-1.17.el6.i686  
安装 b43-fwcutter-012-2.2.el6.i686  
安装 latrace-0.5.9-2.el6.i686  
安装 trace-cmd-1.0.5-10.el6.i686  
安装 crash-trace-command-1.0-4.el6.i686  
安装 byacc-1.9.20070509-7.el6.i686  
安装 diffstat-1.51-2.el6.i686  
安装 ctags-5.8-2.el6.i686  
*** FINISHED INSTALLING PACKAGES *** [root@localhost ~]#
```

作用 2: cat 还可以对文件进行合并

语法 2: #cat 待合并的文件路径 1 待合并的文件路径 2 文件路径 n > 合并之后的文件路径



例如，合并 3 个文件，并存到一个文件中【配合输出重定向使用】

```
[root@localhost ~]# cat /home/linux123/linux.txt /home/linux123/linux1.txt /home/linux123/linux10.txt > hebing.txt
[root@localhost ~]# ls
anaconda-ks.cfg  c          install.log      ls.txt          公共的  视频  文档  音乐
b                hebing.txt  install.log.syslog  yunweihenniu  模板  图片  下载  桌面
[root@localhost ~]# cat hebing.txt
linux.txt
linux1.txt
linux10.txt
```

三、进阶指令（重点）

1、df 指令

作用：查看磁盘的空间

语法：**#df -h** **-h** 表示以可读性较高的形式展示大小

```
[root@localhost ~]# df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/VolGroup-lv_root  18G    3.6G   13G   22% /
tmpfs                      948M    224K   947M   1% /dev/shm
/dev/sda1                   485M     37M   423M   9% /boot
/dev/sr0                     3.6G     3.6G     0 100% /media/CentOS_6.5_Final
```

2、free 指令

作用：查看内存使用情况

语法：**#free -m** **-m** 表示以 **mb** 为单位查看

```
[root@localhost ~]# free -m
              total        used         free     shared    buffers     cached
Mem:           1894          540         1353          0          58          253
-/+ buffers/cache:           228         1665
Swap:           2047           0         2047
```

剩余的真实可以用的内存为 1665mb。

Swap：用于临时内存，当系统真实内存不够用的时候可以临时使用磁盘空间来充当内存。

3、head 指令

作用：查看一个文件的前 n 行，如果不指定 n，则默认显示前 10 行。

语法：**#head -n 文件路径** **【n 表示数字】**

```
[root@localhost ~]# head -5 install.log
安装 fontpackages-filesystem-1.41-1.1.el6.noarch
warning: fontpackages-filesystem-1.41-1.1.el6.noarch: Header V3 RSA/SHA256 Signature, key ID c105b9de: NOKEY
安装 m17n-db-1.5.5-1.1.el6.noarch
安装 setup-2.8.14-20.el6_4.1.noarch
安装 liberation-fonts-common-1.05.1.20090721-5.el6.noarch
```

4、tail 指令

作用 1: 查看一个文件的未 n 行, 如果 n 不指定默认显示后 10 行

语法: #tail -n 文件的路径 n 同样表示数字

```
[root@localhost ~]# tail -1 install.log
*** FINISHED INSTALLING PACKAGES ***[root@localhost ~]#
```

作用 2: 可以通过 tail 指令来查看一个文件的动态变化内容【变化的内容不能是用户手动增加的】

语法: #tail -f 文件路径

该命令一般用于查看系统的日志比较多。

5、less 指令

作用: 查看文件, 以较少的内容进行输出, 按下辅助功能键 (数字+回车、空格键+上下方向键) 查看更多

语法: #less 需要查看的文件路径

```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
安装 fontpackages-filesystem-1.41-1.1.el6.noarch
warning: fontpackages-filesystem-1.41-1.1.el6.noarch: Header V3 RSA/SHA256 Si
ture, key ID c105b9de: NOKEY
安装 m17n-db-1.5.5-1.1.el6.noarch
安装 setup-2.8.14-20.el6_4.1.noarch
安装 liberation-fonts-common-1.05.1.20090721-5.el6.noarch
安装 xkeyboard-config-2.6-6.el6.noarch
安装 xml-common-0.6.3-32.el6.noarch
安装 iso-codes-3.16-2.el6.noarch
安装 filesystem-2.4.30-3.el6.i686
安装 dejavu-fonts-common-2.30-2.el6.noarch
安装 control-center-filesystem-2.28.1-39.el6.i686
安装 paktype-fonts-common-2.0-8.el6.noarch
安装 mesa-dri-filesystem-9.2-0.5.el6.i686
安装 autocorr-zh-4.0.4.2-9.el6.noarch
安装 autocorr-en-4.0.4.2-9.el6.noarch
安装 mailcap-2.1.31-2.el6.noarch
安装 tzdata-2013g-1.el6.noarch
安装 mozilla-filesystem-1.9-5.1.el6.i686
安装 tzdata-java-2013g-1.el6.noarch
安装 foomatic-db-filesystem-4.0-7.20091126.el6.noarch
安装 cjkuni-fonts-common-0.2.20080216.1-36.el6.noarch
安装 basesystem-10.0-4.el6.noarch
```

在退出的只需要按下 q 键即可。

6、wc 指令

作用: 统计文件内容信息 (包含行数、单词数、字节数)

语法: **#wc -lwc** 需要统计的文件路径

-l: 表示 lines, 行数

-w: 表示 words, 单词数 依照空格来判断单词数量

-c: 表示 bytes, 字节数

```
[root@localhost ~]# wc install.log
1155 2328 45503 install.log
[root@localhost ~]# wc -l install.log
1155 install.log
[root@localhost ~]# wc -w install.log
2328 install.log
[root@localhost ~]# wc -c install.log
45503 install.log
[root@localhost ~]# █
```

7、date 指令 (重点)

作用: 表示操作时间日期 (读取、设置)

语法 1: **#date** 输出的形式: 2018年3月24日 星期六 15:54:28

语法 2: **#date +%F** (等价于**#date +%Y-%m-%d**) 输出形式: 2018-03-24

语法 3: **#date +%F %T** 引号表示让“年月日与时分秒”成为一个不可分割的整体

等价操作#date +%Y-%m-%d %H:%M:%S

输出的形式: 2018-03-24 16:01:00

语法 4: 获取之前或者之后的某个时间 (备份)

#date -d "-1 day" +%Y-%m-%d %H:%M:%S

符号的可选值: + (之后) 或者 - (之前)

单位的可选值: **day** (天)、**month** (月份)、**year** (年)

%F: 表示完整的年月日

%T: 表示完整的时分秒

%Y: 表示四位年份

%m: 表示两位月份 (带前导 0)

%d: 表示日期 (带前导 0)

%H: 表示小时 (带前导 0)

%M: 表示分钟 (带前导 0)

%S: 表示秒数 (带前导 0)

8、cal 指令

作用: 用来操作日历的

语法 1: **#cal** 等价于 **#cal -1** 直接输出当前月份的日历

语法 2: **#cal -3** 表示输出上一个月+本月+下个月的日历

语法 3: **#cal -y 年份** 表示输出某一个年份的日历

9、clear/ctrl + L 指令

作用：清除终端中已经存在的命令和结果（信息）。

语法：clear 或者快捷键：ctrl + L

需要注意的是，该命令并不是真的清除了之前的信息，而是把之前的信息的隐藏到了最上面，通过滚动条继续查看以前的信息。

10、管道（重要）

管道符：|

作用：管道一般可以用于“过滤”，“特殊”，“扩展处理”。

语法：管道不能单独使用，必须需要配合前面所讲的一些指令来一起使用，其作用**主要是辅助作用**。

①过滤案例（**100%使用**）：需要通过管道查询出根目录下包含“y”字母的文档名称。

```
#ls | grep y
```

针对上面这个命令说明：

①以管道作为分界线，前面的命令有个输出，后面需要先输入，然后再过滤，最后再输出，通俗的讲就是管道前面的输出就是后面指令的输入；

②grep 指令：**主要用于过滤**

②特殊用法案例：通过管道的操作方法来实现 less 的等价效果（了解）

之前通过 less 查看一个文件，可以#less 路径

现在通过管道还可以这么：#cat 路径|less

③扩展处理：请使用学过的命令，来统计某个目录下的文档的总个数？

答：#ls | wc -l

Linux 的基本指令 (2)

一、高级指令

1、hostname 指令

作用：操作服务器的**主机名**（**读取**、设置）

语法 1: #hostname 含义：表示输出完整的主机名

语法 2: #hostname -f 含义：表示输出当前主机名中的 **FQDN**（全限定**域名**）

```
[root@localhost 桌面]# hostname
localhost.localdomain
[root@localhost 桌面]# hostname -f
localhost
[root@localhost 桌面]# █
```

2、id 指令

作用：查看一个用户的一些基本信息（包含用户 **id**，用户组 **id**，附加组 **id**...），该指令如果不指定用户则默认当前用户。

语法 1: #id 默认显示当前执行该命令的用户的基本信息

语法 2: #id 用户名 显示指定用户的基本信息

```
[root@localhost 桌面]# id
uid=0(root) gid=0(root) 组=0(root) 环境=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c
1023
[root@localhost 桌面]# id linux123
uid=500(linux123) gid=500(linux123) 组=500(linux123)
[root@localhost 桌面]# █
```

验证上述信息是否正确？

验证用户信息：通过文件/etc/passwd

验证用户组信息：通过文件/etc/group

```
[root@localhost 桌面]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

3、whoami 指令

作用：“我是谁？”显示当前登录的用户名，一般用于 shell 脚本，用于获取当前操作的用户名方便记录日志。

语法：#whoami

```
[root@localhost 桌面]# whoami
root
[root@localhost 桌面]# █
```

4、ps -ef 指令（重点）

指令：ps

作用：主要是查看服务器的进程信息

选项含义：

-e: 等价于“-A”，表示列出全部的进程

-f: 显示全部的列（显示全字段）

执行结果：

```
[root@localhost 桌面]# ps -ef
UID          PID    PPID  C  STIME TTY          TIME CMD
root          1      0  0  08:56 ?           00:00:01 /sbin/init
root          2      0  0  08:56 ?           00:00:00 [kthreadd]
root          3      2  0  08:56 ?           00:00:00 [migration/0]
root          4      2  0  08:56 ?           00:00:00 [ksoftirqd/0]
root          5      2  0  08:56 ?           00:00:00 [migration/0]
root          6      2  0  08:56 ?           00:00:03 [watchdog/0]
root          7      2  0  08:56 ?           00:00:00 [migration/1]
```

列的含义：

UID: 该进程执行的用户 id;

PID: 进程 id;

PPID: 该进程的父级进程 id，如果一个程序的父级进程找不到，该程序的进程称之为僵尸进程 (parent process ID);

C: Cpu 的占用率，其形式是百分数;

STIME: 进程的启动时间;

TTY: 终端设备，发起该进程的设备识别符号，如果显示“?”则表示该进程并不是由终端设备发起;

TIME: 进程的执行时间;

CMD: 该进程的名称或者对应的路径;

案例：（100%使用的命令）在 ps 的结果中过滤出想要查看的进程状态

#ps -ef|grep “进程名称”

```
[root@localhost 桌面]# ps -ef|grep gnome-panel
root      2316   2218   0 08:58 ?        00:00:01 gnome-panel
root      28727  28428   0 19:42 pts/1    00:00:00 grep gnome-panel
```

再例如查看火狐浏览器的进程:

```
[root@localhost 桌面]# ps -ef|grep firefox
root      28749  28428   0 19:45 pts/1    00:00:00 grep firefox
[root@localhost 桌面]# ps -ef|grep firefox
root      28751     1  31 19:45 ?        00:00:08 /usr/lib/firefox/firefox
root      28802  28428   2 19:46 pts/1    00:00:00 grep firefox
```

5、top 指令 (重点)

作用：查看服务器的进程占的资源（100%使用）

语法：

进入命令：`#top` (动态显示)

退出命令：按下 `q` 键

输出的结果：

```
top - 19:53:57 up 10:57, 3 users, load average: 0.00, 0.01, 0.00
Tasks: 190 total, 1 running, 189 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.1%us, 0.0%sy, 0.0%ni, 99.9%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1939652k total, 617504k used, 1322148k free, 66584k buffers
Swap: 2097144k total, 0k used, 2097144k free, 307364k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
    1 root        20   0  2900  1440 1216  S   0.0   0.1   0:01.80  init
    2 root        20   0     0     0     0  S   0.0   0.0   0:00.01  kthreadd
    3 root        RT   0     0     0     0  S   0.0   0.0   0:00.53  migration/0
    4 root        20   0     0     0     0  S   0.0   0.0   0:00.15  ksoftirqd/0
    5 root        RT   0     0     0     0  S   0.0   0.0   0:00.00  migration/0
```

表头含义：

PID：进程 id；

USER：该进程对应的用户；

PR：优先级；

VIRT：虚拟内存；

RES：常驻内存；

SHR：共享内存；

计算一个进程实际使用的内存 = 常驻内存 (RES) - 共享内存 (SHR)

S：表示进程的状态 status (sleeping, 其中 S 表示睡眠, R 表示运行)；

%CPU：表示 CPU 的占用百分比；

%MEM：表示内存的占用百分比；

TIME+：执行的时间；

COMMAND：进程的名称或者路径；

在运行 top 的时候，可以按下方便的快捷键：

M：表示将结果按照内存 (MEM) 从高到低进行降序排列；

P: 表示将结果按照 CPU 使用率从高到低进行降序排列;

1: 当服务器拥有多个 cpu 的时候可以使用“1”快捷键来切换是否展示显示各个 cpu 的详细信息;

6、du -sh 指令

作用: 查看目录的真实大小

语法: #du -sh 目录路径

选项含义:

-s: summaries, 只显示汇总的大小

-h: 表示以高可读性的形式进行显示

案例: 统计“/root/yunweihennix”目录的实际大小

```
[root@localhost ~]# du -sh /root/yunweihennix
20K    /root/yunweihennix
[root@localhost ~]# du -sh yunweihennix/
20K    yunweihennix/
[root@localhost ~]# █
```

案例: 统计“/etc”目录实际大小

```
[root@localhost ~]# du -sh /etc/
37M    /etc/
[root@localhost ~]# █
```

7、find 指令

作用: 用于查找文件 (其参数有 55 个之多)

语法: #find 路径范围 选项 选项的值

选项:

-name: 按照文档名称进行搜索 (支持模糊搜索)

-type: 按照文档的类型进行搜索

文档类型: “-”表示文件 (在使用 find 的时候需要用 f 来替换), “d”表示文件夹

案例: 使用 find 来搜索 httpd.conf

#find / -name httpd.conf

```
[root@localhost ~]# find / -name httpd.conf
/etc/httpd/conf/httpd.conf
[root@localhost ~]# █
```

案例: 搜索 etc 目录下所有的 conf 后缀文件

#find /etc -name *.conf



```
[root@localhost ~]# find /etc -name *.conf
/etc/sane.d/microtek2.conf
/etc/sane.d/rts8891.conf
/etc/sane.d/dc210.conf
/etc/sane.d/saned.conf
/etc/sane.d/nec.conf
.....
```

案例：使用 find 来搜索/etc/sane.d/目录下所有的文件

```
#find /etc/sane.d/ -type f
[root@localhost ~]# find /etc/sane.d/ -type f
/etc/sane.d/microtek2.conf
/etc/sane.d/rts8891.conf
/etc/sane.d/dc210.conf
/etc/sane.d/saned.conf
/etc/sane.d/nec.conf
/etc/sane.d/hp3900.conf
.....
```

案例：使用 find 来搜索/etc/目录下所有的文件夹

```
#find /etc -type d
[root@localhost ~]# find /etc -type d
/etc
/etc/gtk-2.0
/etc/gtk-2.0/i386-redhat-linux-gnu
/etc/sane.d
/etc/sane.d/dll.d
/etc/chkconfig.d
```

8、service 指令（重点）

作用：用于控制一些软件的服务启动/停止/重启

语法：**#service 服务名 start/stop/restart**

例如：需要启动本机安装的 Apache（网站服务器软件），其服务名 httpd

```
#service httpd start
.....
[root@localhost ~]# service httpd start
正在启动 httpd: httpd: Could not reliably determine the server's fully qualified domain name,
ocaldomain for ServerName
..... [确定]
```

通过 ps 命令来检查 httpd 服务是否启动:

```
[root@localhost ~]# ps -ef | grep httpd
root      29867      1   0  21:04 ?        00:00:00 /usr/sbin/httpd
apache    29870  29867   0  21:04 ?        00:00:00 /usr/sbin/httpd
apache    29871  29867   0  21:04 ?        00:00:00 /usr/sbin/httpd
apache    29872  29867   0  21:04 ?        00:00:00 /usr/sbin/httpd
apache    29873  29867   0  21:04 ?        00:00:00 /usr/sbin/httpd
apache    29874  29867   0  21:04 ?        00:00:00 /usr/sbin/httpd
apache    29875  29867   0  21:04 ?        00:00:00 /usr/sbin/httpd
apache    29876  29867   0  21:04 ?        00:00:00 /usr/sbin/httpd
apache    29877  29867   0  21:04 ?        00:00:00 /usr/sbin/httpd
root      29881  28428   0  21:06 pts/1    00:00:00 grep httpd
```

9、kill 指令（重点）

作用：表示杀死进程（当遇到僵尸进程或者出于某些原因需要关闭进程的时候）

语法：**#kill 进程 PID**（语法需要配合 ps 一起使用）

案例：需要 kill 掉 Apache 的进程

```
[root@localhost ~]# ps -ef|grep httpd
root    29867      1  0 21:04 ?        00:00:00 /usr/sbin/httpd
apache  29870 29867  0 21:04 ?        00:00:00 /usr/sbin/httpd
apache  29871 29867  0 21:04 ?        00:00:00 /usr/sbin/httpd
apache  29872 29867  0 21:04 ?        00:00:00 /usr/sbin/httpd
apache  29873 29867  0 21:04 ?        00:00:00 /usr/sbin/httpd
apache  29874 29867  0 21:04 ?        00:00:00 /usr/sbin/httpd
apache  29875 29867  0 21:04 ?        00:00:00 /usr/sbin/httpd
apache  29876 29867  0 21:04 ?        00:00:00 /usr/sbin/httpd
apache  29877 29867  0 21:04 ?        00:00:00 /usr/sbin/httpd
root    30518 30023  0 23:54 pts/1    00:00:00 grep httpd
[root@localhost ~]# kill 29867
[root@localhost ~]# ps -ef|grep httpd
root    30521 30023  0 23:55 pts/1    00:00:00 grep httpd
[root@localhost ~]# kill 29867
bash: kill: (29867) - 没有那个进程
```

与 kill 命令作用相似但是比 kill 更加好用的杀死进程的命令：killall

语法：**#killall 进程名称**

```
[root@localhost ~]# killall httpd
[root@localhost ~]# ps -ef|grep httpd
root    30722 30023  0 00:01 pts/1    00:00:00 grep httpd
```

10、ifconfig 指令（重点）

作用：用于操作网卡相关的指令。

简单语法：**#ifconfig**（获取网卡信息）

```
[root@localhost ~]# ifconfig
eth0    Link encap: Ethernet  HWaddr 00:0C:29:E7:DF:85
        inet addr:192.168.21.128  Bcast:192.168.21.255  Mask:255.255.255.0
        inet6 addr: fe80::20c:29ff:fee7:df85/64  Scope: Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:2068 errors:1 dropped:1 overruns:0 frame:0
        TX packets:1293 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:884983 (864.2 KiB)  TX bytes:126047 (123.0 KiB)
        Interrupt:19  Base address:0x2000

lo      Link encap: Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128  Scope: Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:114 errors:0 dropped:0 overruns:0 frame:0
        TX packets:114 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:27257 (26.6 KiB)  TX bytes:27257 (26.6 KiB)
```


Eth0 表示 Linux 中的一个网卡，eth0 是其名称。Lo (loop, 本地回还网卡, 其 ip 地址一般都是 127.0.0.1) 也是一个网卡名称。

```
[root@localhost ~]# ifconfig
eth0      Link encap: Ethernet  HWaddr 00:0C:29:E7:DF:85
          inet addr: 192.168.21.128  Bcast: 192.168.21.255  Mask: 255.255.255.0
          inet6 addr: fe80::20c:29ff:fee7:df85/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU: 1500  Metric: 1
          RX packets: 2068  errors: 1  dropped: 1  overruns: 0  frame: 0
          TX packets: 1293  errors: 0  dropped: 0  overruns: 0  carrier: 0
          collisions: 0  txqueuelen: 1000
          RX bytes: 884983 (864.2 KiB)  TX bytes: 126047 (123.0 KiB)
          Interrupt: 19  Base address: 0x2000

lo        Link encap: Local Loopback
          inet addr: 127.0.0.1  Mask: 255.0.0.0
          inet6 addr: ::1/128 Scope: Host
          UP LOOPBACK RUNNING  MTU: 16436  Metric: 1
          RX packets: 114  errors: 0  dropped: 0  overruns: 0  frame: 0
          TX packets: 114  errors: 0  dropped: 0  overruns: 0  carrier: 0
          collisions: 0  txqueuelen: 0
          RX bytes: 27257 (26.6 KiB)  TX bytes: 27257 (26.6 KiB)
```

注意: inet addr 就是网卡的 ip 地址。

11、reboot 指令

作用: 重新启动计算机

语法 1: #reboot 重启

语法 2: #reboot -w 模拟重启, 但是不重启 (只写关机与开机的日志信息)

12、shutdown 指令

作用: 关机 (慎用)

语法 1: #shutdown -h now “关机提示” 或者 #shutdown -h 15:25 “关机提示”

案例: 设置 Linux 系统关机时间在 12:00

```
[root@localhost ~]# reboot -w [确定]
[root@localhost ~]# shutdown -h 12:00 "系统将在 12:00 关机, 请保存好你的文档!"

Broadcast message from root@localhost.localdomain
(/dev/pts/1) at 0:28 ...

The system is going down for halt in 692 minutes!
系统将在 12:00 关机, 请保存好你的文档!
^Cshutdown: Shutdown cancelled
[root@localhost ~]# █
```

如果想要取消关机计划的话, 则可以按照以下方式去尝试:

① 针对于 centos7.x 之前的版本: ctrl+c

② 针对于 centos7.x (包含) 之后的版本: #shutdown -c

除了 shutdown 关机以外, 还有以下几个关机命令:

```
#init 0
#halt
#poweroff
```

13、uptime 指令

作用：输出计算机的持续在线时间（计算机从开机到现在运行的时间）

语法：**#uptime**

```
[root@localhost ~]# uptime
 00:55:42 up 15:59,  5 users,  load average: 0.00, 0.00, 0.00
[root@localhost ~]#
```

14、uname 指令

作用：获取计算机操作系统相关信息

语法 1：**#uname** 获取操作系统的类型

语法 2：**#uname -a** all, 表示获取全部的系统信息（类型、全部主机名、内核版本、发布时间、开源计划）

```
[root@localhost ~]# uname
Linux
[root@localhost ~]# uname -a
Linux localhost.localdomain 2.6.32-431.el6.i686 #1 SMP Fri Nov 22 00:26:36 UTC 2013 i686 i686 i386 GNU/Linux
[root@localhost ~]#
```

15、netstat -tnlp 指令

作用：查看网络连接状态

语法：**#netstat -tnlp**

```
[root@localhost ~]# netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111             0.0.0.0:*                 LISTEN      1424/rpcbind
tcp        0      0 0.0.0.0:36656          0.0.0.0:*                 LISTEN      1573/rpc.statd
tcp        0      0 0.0.0.0:22             0.0.0.0:*                 LISTEN      1770/sshd
tcp        0      0 127.0.0.1:631          0.0.0.0:*                 LISTEN      1607/cupsd
tcp        0      0 127.0.0.1:25           0.0.0.0:*                 LISTEN      1883/master
tcp        0      0 :::111                 :::*                     LISTEN      1424/rpcbind
tcp        0      0 :::80                  :::*                     LISTEN      30790/httpd
tcp        0      0 :::22                  :::*                     LISTEN      1770/sshd
tcp        0      0 :::1:631               :::*                     LISTEN      1607/cupsd
tcp        0      0 :::1:25                 :::*                     LISTEN      1883/master
tcp        0      0 :::34521                :::*                     LISTEN      1573/rpc.statd
```

选项说明：

- t: 表示只列出 tcp 协议的连接；
- n: 表示将地址从字母组合转化成 ip 地址，将协议转化成端口号来显示；
- l: 表示过滤出“state（状态）”列中其值为 LISTEN（监听）的连接；
- p: 表示显示发起连接的进程 pid 和进程名称；

16、man 指令

作用：manual，手册（包含了 Linux 中全部命令手册，英文）

语法：#man 命令 （退出按下 q 键）

案例：通过 man 命令查询 cp 指令的用法

```
#man cp
```

二、练习题

- 1、如何通过命令行重启 linux 操作系统？ #reboot
- 2、如何在命令行中快速删除光标前/后的内容？ 前：ctrl+u 后：ctrl+k
- 3、如何删除/tmp 下所有 A 开头的文件？ #rm -f /tmp/A*
- 4、系统重要文件需要备份，如何把/etc/passwd 备份到/tmp 目录下？
#cp /etc/passwd /tmp/
- 5、如何查看系统最后创建的 3 个用户？
#tail -3 /etc/passwd
- 6、什么命令可以统计当前系统中一共有多少账户？
#wc -l /etc/passwd #cat /etc/passwd|wc -l
- 7、如何创建/tmp/test.conf 文件？
#touch /tmp/test.conf
- 8、如何通过 vim 编辑打开/tmp/test.conf？
#vim /tmp/test.conf
- 9、如何查看/etc/passwd 的头 3 行和尾 3 行？
#head -3 /etc/passwd
#tail -3 /etc/passwd
- 10、如何一次性创建目录/text/1/2/3/4？
#mkdir -p /text/1/2/3/4
- 11、如何最快的返回到当前账户的家目录？
#cd ~ #cd
- 12、如何查看/etc 所占的磁盘空间？
#du -sh /etc
- 13、如何删除/tmp 下所有的文件？
#rm -rf /tmp/*
- 14、尝试启动 Apache 的服务，并且检查是否启动成功。
#service httpd start
#ps -ef|grep httpd
- 15、使用已学命令杀死 Apache 的进程。
#killall httpd

编辑器之神——vim 编辑器

一、vi 介绍

Vi 编辑器是所有 Unix 及 Linux 系统下标准的编辑器，类似于 windows 系统下的 notepad（记事本）编辑器，由于在 Unix 及 Linux 系统的任何版本，Vi 编辑器是完全相同的，因此可以在其他任何介绍 vi 的地方都能进一步了解它，Vi 也是 Linux 中最基本的文本编辑器，学会它后，我们将在 Linux 的世界里畅行无阻，**尤其是在终端中。**

关于 vim:

vi 和 vim 都是 Linux 中的编辑器，不同的是，**vim 比较高级**，可以视为 vi 的升级版。vi 用于文本编辑，但是 vim 更适用于 coding（写代码的）。

Vim 重点是光标的移动，模式切换，删除，查找，替换，复制，粘贴，撤销命令的使用。

二、vim 三种模式（重点）

Vim 中存在三种模式（大众的认知）：**命令模式、编辑模式（输入模式）、末行模式（尾行模式）。**

命令模式：在该模式下是不能对文件**直接**编辑，可以输入快捷键进行一些操作（删除行，复制行，移动光标，粘贴等等）**【打开文件之后默认进入的模式】**；

编辑模式：在该模式下可以对文件的内容进行编辑；

末行模式：可以在末行输入命令来对文件进行操作（搜索、替换、保存、退出、撤销、高亮等等）；

Vim 的打开文件的方式（4 种，要求掌握的就前三种）：

#vim 文件路径 作用：打开指定的文件

#vim +数字 文件的路径 作用：打开指定的文件，并且将光标移动到指定行

#vim +/关键词 文件的路径 作用：打开指定的文件，并且高亮显示关键词

#vim 文件路径 1 文件路径 2 文件路径 3 作用：同时打开多个文件

重点：先复制出一个/etc/passwd 文件，复制当前家目录下（千万不要在 etc 下直接修改!!!）

```
[ root@localhost 桌面]# cd
[ root@localhost ~]# cp /etc/passwd ~/
[ root@localhost ~]# ls
anaconda-ks.cfg  install.log          passwd              公共的  视频  文档  音乐
hebing.txt       install.log.syslog  yunweihenniu      模板    图片  下载  桌面
[ root@localhost ~]# █
```

后续一切 vim 命令都是基于/root/passwd 文件进行操作。



退出方式：输入:q 按下回车即可

```
[root@localhost ~]# vim passwd
[root@localhost ~]# vim +10 passwd
[root@localhost ~]# vim +/login passwd
[root@localhost ~]# vim passwd hebing.txt install.log
还有 3 个文件等待编辑
[root@localhost ~]# █
```

三、命令模式

注意：该模式是打开文件的第一个看到的模式（打开文件即可进入）

1、光标移动

①光标移动到行首

按键：shift + 6 或 ^（T 字母上面的 6，不要按小键盘的 6）

②光标移动到行尾

按键：shift + 4 或 \$（R 字母的左上角的 4，不是小键盘的 4）

③光标移动到首行

按键：gg

④光标移动到末行

按键：G

⑤翻页

向上翻页：按键 ctrl + b （before） 或 PgUp

向下翻页：按键 ctrl + f （after） 或 PgDn

2、复制操作

①复制光标所在行

按键：yy

粘贴：在想要粘贴的地方按下 p 键

②以光标所在行为准（包含当前行），向下复制指定的行数

按键：数字 yy

③可视化复制

按键：ctrl+v（可视块）或 V（可视行）或 v（可视），然后按下 ↑ ↓ ← → 方向键来选中需要复制的区块，按下 y 键进行复制，最后按下 p 键粘贴

3、剪切/删除

①剪切/删除光标所在行

按键：dd （删除之后下一行上移）

注意：dd 严格意义上说是剪切命令，但是如果剪切了不粘贴就是删除的效果。

②剪切/删除光标所在行为准（包含当前行），向下删除/剪切指定的行

按键：数字 dd （删除之后下一行上移）

③剪切/删除光标所在的当前行之后的内容，但是删除之后下一行不上移

按键：D （删除之后当前行会变成空白行）

④可视化删除

按键：ctrl+v（可视块）或 V（可视行）或 v（可视），上下左右移动，按下 D 表示删除选中行，d 表示删选中块

4、撤销/恢复

撤销：输入 :u （不属于命令模式） 或者 u （undo）

恢复：ctrl+r 恢复（取消）之前的撤销操作

5、扩展 1：光标的快速移动

①快速将光标移动到指定的行

按键：数字 G

②以当前光标为准向上/向下移动 n 行

按键：数字 ↑，数字 ↓

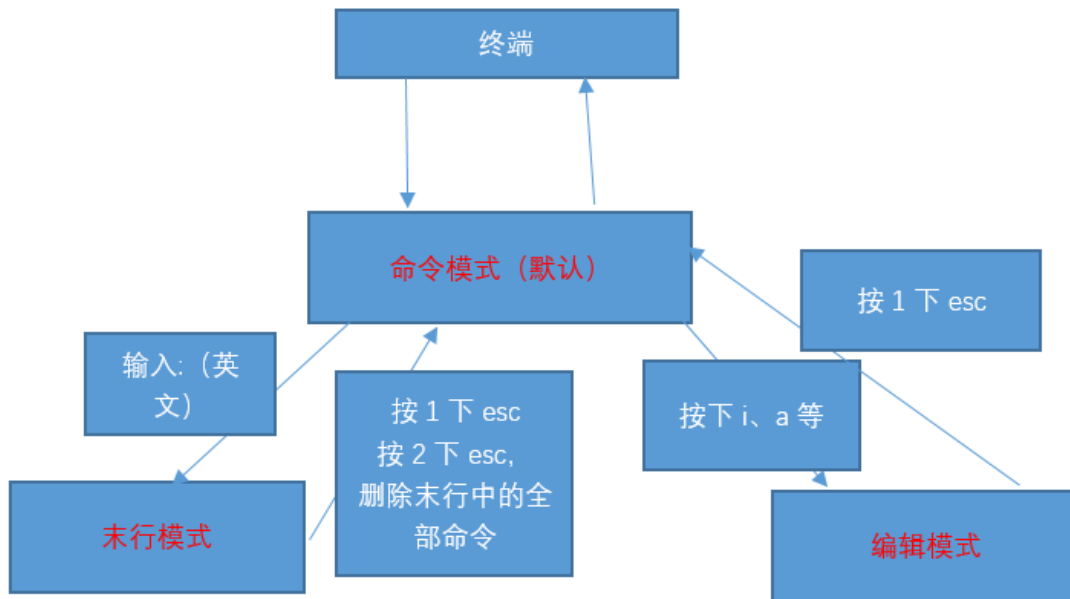
③以当前光标为准向左/向右移动 n 字符

按键：数字 ←，数字 →

④末行模式下的快速移动方式：移动到指定的行

按键：输入英文 “:”，其后输入行数数字，按下回车

四、模式间的切换（重点）



五、末行模式

进入方式：由命令模式进入，按下 “:” 或者 “/（表示查找）” 即可进入
退出方式：

- a. 按下 esc
- b. 连接 2 次 esc 键
- c. 删除末行全部输入字符

①保存操作（write）

输入：“:w” 保存文件
输入：“:w 路径” 另存为

②退出（quit）

输入：“:q” 退出文件

③保存并退出

输入：“:wq” 保存并且退出

④强制（!）

输入：“:q!” 表示强制退出，刚才做的修改操作不做保存

⑤调用外部命令（了解）

输入：“:!外部命令”

例如：

```
linux123: x: 500: 500: linux123: /home/lin  
:! ls -la /
```

当外部命令执行结束之后按下任意键回到 vim 编辑器打开的内容

```
drwxr-xr-x.  28 root root  4096 3月  23 23:05 tmp  
drwxr-xr-x.  12 root root  4096 3月  23 23:05 usr  
drwxr-xr-x.  22 root root  4096 3月  23 23:16 var
```

请按 ENTER 或其它命令继续

⑥搜索/查找

输入：“/关键词”

例如：我想在 passwd 文件中搜索 “sbin” 关键词

```
tcpdump: x: /2: /2: /:/ /sbin/ no log1  
linux123: x: 500: 500: linux123: /ho  
/sbin
```

在搜索结果中切换上/下一个结果：N/n (next)

如果需要取消高亮，则需要输入：“:nohl”【no highlight】

⑦替换

:s/搜索的关键词/新的内容

替换光标所在行的第一处符合条件的内容

:s/搜索的关键词/新的内容/g

替换光标所在行的全部符合条件的内容

:%s/搜索的关键词/新的内容

替换整个文档中每行第一个符合条件的内容

:%s/搜索的关键词/新的内容/g

替换整个文档的符合条件的内容

%表示整个文件

g 表示全局 (global)

⑧显示行号 (临时)

输入：“:set nu” [number]

如果想取消显示，则输入：“:set nonu”

⑨扩展 2：使用 vim 同时打开多个文件，在末行模式下进行切换文件

查看当前已经打开的文件名称：“:files”

```
: files  
 1 %a "passwd"          第 35 行  
 2    "group"          第 0 行  
 3    "install.log"    第 0 行
```

在%a 的位置有 2 种显示可能

%a: a=active, 表示当前正在打开的文件;

#: 表示上一个打开的文件

切换文件的方式:

a. 如果需要指定切换文件的名称，则可以输入：“:open 已经打开的文件名”



```

: files
 1 #      "passwd"           第 35 行
 2 %a    "group"           第 1 行
 3      "install.log"      第 0 行
请按 ENTER 或其它命令继续

```

b. 可以通过其他命令来切换上一个文件/下一个文件

输入：“:bn” 切换到下一个文件（back next）

输入：“:bp” 切换到上一个文件（back prev）

六、编辑模式

	i: 在光标所在字符前开始插入
	a: 在光标所在字符后开始插入
	o: 在光标所在行的下面另起一新行插入
进入方式	I: 在光标所在行的行首开始插入 如果行首有空格则在空格之后插入
	A: 在光标所在行的行尾开始插入
	O: 在光标所在行的上面另起一行开始插入
	S: 删除光标所在行并开始插入

重点看前 2 个进入方式：i（insert）、a（after）。

退出方式：按下 esc 键

七、实用功能

1、代码着色

```

root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt

```

案例：首先创建简单的 c 语言程序

```
#include <stdio.h>
void main(){
    printf('hello world! \n' );
}
~
```

如何控制着色显示与否？

显示：“:syntax on” syn

tax: 语法

关闭显示：“:syntax off”

2、vim 中计算器的使用

当在编辑文件的时候突然需要使用计算器去计算一些公式，则此时需要用计算器，但是需要退出，vim 自身集成了一个简易的计算器。

- 进入编辑模式
- 按下按键“ctrl + R”，然后输入“=”，此时光标会变到最后一行
- 输入需要计算的内容，按下回车

```
pulse: x: 498: 496: PulseAud1c
=2+2
```

八、扩展（3）

1、vim 的配置（重点）

Vim 是一款编辑器，编辑器也是有配置文件的。

Vim 配置有三种情况：

- 在文件打开的时候在末行模式下输入的配置（临时的）
- 个人配置文件（`~/.vimrc`，如果没有可以自行新建）
- 全局配置文件（vim 自带，`/etc/vimrc`）

①新建好个人配置文件之后进入编辑

②在配置文件中配置

比如显示行号：set nu

```
1 set nu
~
~
```

配置好之后 vim 打开文件就会永远显示行号

问题：如果某个配置项，在个人配置文件与全局配置文件产生冲突的时候应该以谁为准？

测试步骤：在两个配置文件中针对同一个配置项设置不同的值

①先在全局的配置中设置不显示行号，在个人的配置文件中设置显示行号，观察结果最后显示行号：说明以个人为准

②先在全局中配置显示行号，在个人中设置不显示行号，观察结果最后的显示是不显示行号，说明以个人为准

结论：如果针对同一个配置项，个人配置文件中存在，则以个人配置文件为准，如果个人配置文件中不存在这一项，则以全局配置文件为准。

2、异常退出

什么是异常退出：在编辑文件之后并没有正常的去 `wq`（保存退出），而是遇到突然关闭终端或者断电的情况，则会显示下面的效果，这个情况称之为异常退出：

```
E325: 注意
发现交换文件 ".passwd.swp"
    所有者: root    日期: Sun Mar 25 13:21:19 2018
    文件名: ~root/passwd
    修改过: 是
    用户名: root    主机名: localhost.localdomain
    进程 ID: 1997
正在打开文件 "passwd"
    日期: Sun Mar 25 13:02:34 2018
```

(1) 另一个程序可能也在编辑同一个文件。
如果是这样，修改时请注意避免同一个文件产生两个不同的版本。

退出，或小心地继续。

(2) 上次编辑此文件时崩溃。
如果是这样，请用 `":recover"` 或 `"vim -r passwd"`
恢复修改的内容（请见 `":help recovery"`）。
如果你已经进行了恢复，请删除交换文件 `".passwd.swp"`
以避免再看到此消息。

```
交换文件 ".passwd.swp" 已存在！
以只读方式打开([O])，直接编辑((E))，恢复((R))，删除交换文件((D))，退出((Q))，中>
-- 更多 --
```

解决办法：将交换文件（在编程过程中产生的临时文件）删除掉即可

```
#rm -f .passwd.swp
```

```
[root@localhost ~]# rm -f .passwd.swp
[root@localhost ~]#
```

3、别名机制（实用）

作用：相当于创建一些属于自己的自定义命令

例如：在 windows 下有 cls 命令，在 Linux 下可能因为没有这个命令而不习惯清屏。现在可以通过别名机制来解决这个问题，可以自己创造出 cls 命令

别名机制依靠一个别名映射文件：~/.bashrc

```
#vim ~/.bashrc
1 # .bashrc
2
3 # User specific aliases and functions
4
5 alias rm='rm -i'
6 alias cp='cp -i'
7 alias mv='mv -i'
8 alias cls='clear'
9 # Source global definitions
10 if [ -f /etc/bashrc ]; then
11     . /etc/bashrc
12 fi
~
~
```

注意：如果想新创造的命令生效，必须要重新登录当前用户。

4、退出方式

回顾：之前 vim 中退出编辑的文件可以使用“:q”或者“:wq”。

除了上面的这个语法之外，vim 还支持另外一个保存退出方法“:x”。

说明：

① “:x” 在文件没有修改的情况下，表示直接退出，在文件修改的情况下表示保存并退出；

②如果文件没有被修改，但是使用 wq 进行退出的话，则文件的修改时间会被更新；但是如果文件没有被修改，使用 x 进行退出的话，则文件修改时间不会被更新的；主要是会混淆用户对文件的修改时间的认定。

因此建议以后使用“:x”来进行对文件的保存退出。

但是：不要使用 x，不要使用 x，不要使用 x，x 表示对文件进行加密操作。



九、作业

- 1、参考作业文件“httpd-vhosts.conf”的描述；
- 2、使用别名机制，创建一个快捷命令“kj”，要求实现按下“kj”回车之后能够实现：统计出 Apache 的服务进程数量。

Linux 自有服务 (1)

自有服务，即不需要用户独立去安装的软件的服务，而是当系统安装好之后就可以直接使用的服务（内置）。

一、运行模式

运行模式也可以称之为**运行级别**。

在 linux 中存在一个进程：init（initialize，初始化），进程 id 是 1。

查看进程：#ps -ef|grep init

```
[root@localhost 桌面]# ps -ef|grep init
root      1      0  0 15:20 ?        00:00:01 /sbin/init
root     3142  3131  0 16:40 pts/0    00:00:00 grep init
[root@localhost 桌面]# █
```

该进程存在一个对应的配置文件：inittab（**系统运行级别配置文件**，位置/etc/inittab）

文件的主要内容：

```
''
# Default runlevel. The runlevels used are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:
```

根据上述的描述，可以得知，Centos6.5 中存在 7 中运行级别/模式。

- 0 — 表示关机级别（不要将默认的运行级别设置成这个值）
- 1 — 单用户模式
- 2 — 多用户模式，不带 NFS（Network File System）
- 3 — 多用户模式，完全的多用户模式（不带桌面的，纯命令行模式）

- 4 — 没有被使用的模式（被保留模式）
- 5 — X11, 完整的图形化界面模式
- 6 — 表示重启级别（不要将默认的运行级别设置成这个值）

与该级别相关的几个命令：

- #init 0 表示关机
- #init 3 表示切换到不带桌面的模式
- #init 5 切换到图形界面
- #init 6 重启电脑

注意：init 指令需要超级管理员的权限，普通用户无法执行。

这些命令其实都是调用的 **init** 进程，将数字（运行级别）传递给进程，进程去读配置文件执行对应的操作。

①切换到纯命令行模式下（临时切换，重启之后又恢复）

#init 3

```
localhost login: root
Password:
Last login: Sun Mar 25 15:20:28 on tty1
[root@localhost ~]# _
```

切换之后需要输入用户名和密码，在**输入密码的时候没有“*”提示输入，只要自己确认输入的密码没有错误，按下回车即可。**

②回到桌面模式

#init 5

③设置模式永久为命令行模式

```
25 #
26 id: 3: initdefault:
```

将/etc/inittab 文件中的 initdefault 值设置成 3，然后重启操作系统。

二、用户与用户组管理（重点）

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

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

每个用户账号都拥有一个惟一的用户名和各自的密码。

用户在登录时键入正确的用户名和密码后，就能够进入系统和自己的主目录。

要想实现用户账号的管理，要完成的工作主要有如下几个方面：



用户账号的添加、删除、修改以及用户密码的管理。
用户组的管理。

注意三个文件：

/etc/passwd	存储用户的关键信息
/etc/group	存储用户组的关键信息
/etc/shadow	存储用户的密码信息

1、用户管理

①添加用户

常用语法：#useradd 选项 用户名

常用选项：

- g: 表示指定用户的用户主组，选项的值可以是用户组的 id，也可以是组名
- G: 表示指定用户的用户附加组，选项的值可以是用户组的 id，也可以是组名
- u: uid，用户的 id（用户的标识符），系统默认会从 500 之后按顺序分配 uid，如果不想使用系统分配的，可以通过该选项自定义【类似于腾讯 QQ 的自选靓号情况】
- c comment: 添加注释

案例：创建用户 zhangsan，不带任何选项

```
[root@localhost ~]# useradd zhangsan
[root@localhost ~]# _
```

验证是否成功：

- a. 验证/etc/passwd 的最后一行，查看是否有 zhangsan 的信息；
- b. 验证是否存在家目录（在 Centos 下创建好用户之后随之产生一个同名家目录）；

扩展：认识 passwd 文件

```
abrt:x:173:173:/:etc/abrt:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
gdm:x:42:42:/:var/lib/gdm:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
tcpdump:x:72:72:/:/sbin/nologin
linux123:x:500:500:linux123:/home/linux123:/bin/bash
zhangsan:x:501:501:/:home/zhangsan:/bin/bash
```

用户名:密码:用户 ID:用户组 ID:注释:家目录:解释器 shell

- 用户名：创建新用户名称，后期登录的时候需要输入；
- 密码：此密码位置一般情况都是“x”，表示密码的占位；
- 用户 ID：用户的识别符；
- 用户组 ID：该用户所属的主组 ID；
- 注释：解释该用户是做什么用的；
- 家目录：用户登录进入系统之后默认的位置；

解释器 shell：等待用户进入系统之后，用户输入指令之后，该解释器会收集用户输入的指令，传递给内核处理；

注意：在不添加选项的时候，执行 useradd 之后会执行一系列的操作

- a. 创建同名的家目录；
- b. 创建同名的用户组；

案例：添加选项，创建用户 lisi，让 lisi 属于 501 主组，附加组 500，自选靓号 666。

```
[root@localhost ~]# useradd -g 501 -G 500 -u 666 lisi
[root@localhost ~]# _
```

```
linux123:x:500:500:linux123:/home/linux123:/bin/bash
zhangsan:x:501:501:~/home/zhangsan:/bin/bash
lisi:x:666:501:~/home/lisi:/bin/bash
[root@localhost ~]# _
```

```
tcpdump:x:72:
slocate:x:31:
linux123:x:500:lisi
zhangsan:x:501:
[root@localhost ~]# _
```

附加组

注意：查看用户的主组可以查看 passwd 文件，查看附加组可以查看 group 文件。

②修改用户

常用语法：#usermod 选项 用户名

Usermod：user modify，用户修改

常用选项：

- g：表示指定用户的用户主组，选项的值可以是用户组的 id，也可以是组名
- G：表示指定用户的用户附加组，选项的值可以是用户组的 id，也可以是组名
- u：uid，用户的 id（用户的标识符），系统默认会从 500 之后按顺序分配 uid，如果不想使用系统分配的，可以通过该选项自定义【类似于腾讯 QQ 的自选靓号情况】
- l：修改用户名

案例：修改 zhangsan 用户主组为 500，附加组改为 501

```
#usermod -g 500 -G 501 zhangsan
```

案例：修改 zhangsan 用户用户名，改为 wangerma

#usermod -l 新的用户名 旧的用户名

```
#usermod -l wangerma zhangsan
```

③设置密码

Linux 不允许没有密码的用户登录到系统，因此前面创建的用户目前都处于锁定状态，需要设置密码之后才能登录计算机。



常用语法: #passwd 用户名

案例: 设置 wangerma 用户的密码

```
[root@localhost ~]# passwd wangerma
Changing password for user wangerma.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]# _
```

在设置密码的时候也是没有任何输入提示的，放心输入，确保两次输入的密码一致，按下回车即可。

也可以使用弱密码，但是不建议，否则会看到以下的提示：

```
[root@localhost ~]# passwd wangerma
Changing password for user wangerma.
New password:
BAD PASSWORD: it is too simplistic/systematic
BAD PASSWORD: is too simple
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]#
```

设置密码之后 shadow 文件中的体现：能够看出 lisi 用户没有密码的。

```
[root@localhost ~]# tail -3 /etc/shadow
linux123:$6$k/UBhsmSYWsNqZ11$aHsJY2UJyU840JiWctB6hXo2P2ghh5e jZtkG1GpuU9tAizLMmeu
rgeHAzusanIVCc7E.Fyk0JpXJEZA7Q0r1F0:17613:0:99999:7:::
lisi:!!:17615:0:99999:7:::
wangerma:$6$19U/1NvS$qUjUzXf9i0HaNTx1nT/iJvo7fQYjRT8cdzhaaYUh21er434cy1wXbR9vUDa
GSDu/7qJ4BfFfy77UfrckHePT0.:17615:0:99999:7:::
[root@localhost ~]#
```

在设置用户密码之后可以登录帐号，例如此处需要登录 wangerma

切换用户命令: #su [用户名] (switch user)

如果用户名不指定则表示切换到 root 用户。

```
wangerma@localhost:~$ su
[root@localhost ~]# su wangerma
[wangerma@localhost root]$ pwd
/root
[wangerma@localhost root]$ ls
ls: cannot open directory .: Permission denied
[wangerma@localhost root]$ cd
[wangerma@localhost ~]$ su
Password:
[root@localhost zhangsan]# ls
[root@localhost zhangsan]# _
```

切换用户需要注意的事项：

- a. 从 root 往普通用户切换不需要密码，但是反之则需要 root 密码；
- b. 切换用户之后前后的工作路径是不变的；
- c. 普通用户没有办法访问 root 用户家目录，但是反之则可以；

④删除用户

常用语法: #userdel 选项 用户名

Userdel: user delete (用户删除)

常用选项:

-r: 表示删除用户的同时, 删除其家目录;

案例: 删除 wangerma 用户

```
[root@localhost zhangsan]# userdel -r wangerma
userdel: user wangerma is currently logged in
[root@localhost zhangsan]# userdel -r lisi
[root@localhost zhangsan]# tail -3 /etc/passwd
tcpdump:x:72:72:/:/sbin/nologin
linux123:x:500:500:linux123:/home/linux123:/bin/bash
wangerma:x:501:500:~/home/zhangsan:/bin/bash
```

注意: 已经登录的 wangerma 用户删除的时候提示删除失败, 但是没有登录的 lisi 用户可以正常删除。

解决办法: 简单粗暴, kill 对应用户的全部进程

```
[root@localhost zhangsan]# cd
[root@localhost ~]# ps -ef|grep wangerma
root      3151   2023   0 18:48 tty1      00:00:00 su wangerma
wangerma  3152   3151   0 18:48 tty1      00:00:00 bash
root      3219   3178   0 18:57 tty1      00:00:00 grep wangerma
[root@localhost ~]# kill 3151
[root@localhost ~]#
Session terminated, killing shell... ..killed.
Terminated
[root@localhost ~]# exit
^C

[root@localhost ~]# userdel -r wangerma
[root@localhost ~]# _
```

提示: 所有跟用户操作的命令 (除 passwd 外) 只有 root 超级管理员有权限执行。

2、用户组管理

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**。

```
stapsys:x:157:
stapdev:x:158:
fuse:x:494:
haldaemon:x:68:haldaemon
ntp:x:38:
apache:x:48:
saslauth:x:76:
```

文件结构:

用户组名:密码:用户组 ID:组内用户名

密码: X 表示占位符, 虽然用户组可以设置密码, 但是绝大部分的情况下不设置密码;



组内用户名：表示附加组是该组的用户名称；

①用户组添加

常用语法：#groupadd 选项 用户组名

常用选项：

-g：类似用户添加里的“-u”，-g 表示选择自己设置一个自定义的用户组 ID 数字，如果自己不确定，则默认从 500 之后递增；

案例：使用 groupadd 指令创建一个新的用户组，命名为 Administrators

```
[root@localhost ~]# groupadd Administrators
[root@localhost ~]# tail -1 /etc/group
Administrators:x:502:
[root@localhost ~]# _
```

成功的结果

②用户组编辑

常用语法：#groupmod 选项 用户组名

常用选项：

-g：类似用户修改里的“-u”，-g 表示选择自己设置一个自定义的用户组 ID 数字

-n：类似于用户修改“-l”，表示设置新的用户组的名称

案例：修改 Administrators 用户组，将组 ID 从 502 改成 520，将名称改为 admins

```
[root@localhost ~]# groupmod -g 520 -n admins Administrators
[root@localhost ~]# tail -1 /etc/group
admins:x:520:
[root@localhost ~]# _
```

③用户组删除

常用语法：#groupdel 用户组名

```
zhangsan:x:501:
[root@localhost ~]# groupdel zhangsan
[root@localhost ~]# groupdel linux123
groupdel: cannot remove the primary group of user 'linux123'
[root@localhost ~]# _
```

注意：当如果需要删除一个组，但是**这个组是某个用户的主组时**，则不允许删除；如果确实需要删除，则先从组内移出所有用户。

```
[root@localhost ~]# usermod -g 0 linux123
[root@localhost ~]# tail -3 /etc/passwd
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
tcpdump:x:72:72:/:/sbin/nologin
linux123:x:500:0:linux123:/home/linux123:/bin/bash
[root@localhost ~]# groupdel linux123
[root@localhost ~]# tail -3 /etc/group
sshd:x:74:
tcpdump:x:72:
slocate:x:21:
[root@localhost ~]# _
```

三、网络设置

首先知道网卡配置文件位置：/etc/sysconfig/network-scripts

```
[root@localhost ~]# ls /etc/sysconfig/network-scripts/
ifcfg-eth0  ifdown-isdn  ifup-aliases  ifup-plusb  init.ipv6-global
ifcfg-lo    ifdown-post  ifup-bnep     ifup-post   net.hotplug
ifdown     ifdown-ppp   ifup-eth      ifup-ppp    network-functions
ifdown-bnep ifdown-routes ifup-ipppp    ifup-routes network-functions-ipv6
ifdown-eth  ifdown-sit   ifup-ipv6     ifup-sit
ifdown-ippp ifdown-tunnel ifup-isdn     ifup-tunnel
ifdown-ipv6 ifup         ifup-plip     ifup-wireless
```

在目录中网卡的配置文件命名格式：ifcfg-网卡名称

```
[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
TYPE=Ethernet
UUID=2838e2cb-4428-4d79-9710-fdee87e0debd
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=dhcp
HWADDR=00:0C:29:E7:DF:85
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME="System eth0"
```

ONBOOT: 是否开机启动

BOOTPROTO: ip 地址分配方式，DHCP 表示动态主机分配协议

HWADDR: 硬件地址，MAC 地址

如果后续需要重启网卡怎么去操作呢？

#service network restart

```
[root@localhost ~]# service network restart
Shutting down interface eth0: Device state: 3 (disconnected)
[ OK ]
Shutting down loopback interface:
[ OK ]
Bringing up loopback interface:
[ OK ]
Bringing up interface eth0: Active connection state: activating
Active connection path: /org/freedesktop/NetworkManager/ActiveConnection/1
state: activated
Connection activated
[ OK ]
[root@localhost ~]# _
```

在有的分支版本中可能没有 service 命令来快速操作服务，但是有一个共性的目录: /etc/init.d 这个目录中放着很对服务的快捷方式。

此处重启网卡命令还可以使用：

#/etc/init.d/network restart



```

[ root@localhost ~ ]# /etc/init.d/network restart
Shutting down interface eth0: Device state: 3 (disconnected)
[ OK ]
Shutting down loopback interface:
[ OK ]
Bringing up loopback interface:
[ OK ]
Bringing up interface eth0: Active connection state: activating
Active connection path: /org/freedesktop/NetworkManager/ActiveConnection/2
state: activated
Connection activated
[ OK ]
[ root@localhost ~ ]# _

```

扩展 1: 如果修改网卡的配置文件，但是配置文件的目录层次很深，此时可以在浅的目录中创建一个快捷方式（软连接），方便以后去查找

#ln -s 原始文件的路径 快捷方式的路径

```

[ root@localhost ~ ]# ls
anaconda-ks.cfg  ifcfg-eth0      password
group           install.log     sss.txt
hebing.txt      install.log.syslog  gunweihenniu
hello.c         passwd
[ root@localhost ~ ]#

```

通过 `ls -l` 可以列出如下的效果:

```

[ root@localhost ~ ]# ls -l
total 128
-rw-----. 1 root root 1641 Mar 23 23:17 anaconda-ks.cfg
-rw-r--r--. 1 root root  887 Mar 25 09:04 group
-rw-r--r--. 1 root root   33 Mar 24 14:32 hebing.txt
-rw-r--r--. 1 root root   64 Mar 25 12:52 hello.c
lrwxrwxrwx. 1 root root   41 Mar 25 22:32 ifcfg-eth0 -> /etc/sysconfig/network-
scripts/ifcfg-eth0
-rw-r--r--. 1 root root 45503 Mar 23 23:17 install.log
-rw-r--r--. 1 root root 18833 Mar 23 23:16 install.log.syslog

```

其中，文件类型位置的“l”表示其类型为 link（连接类型），后面的“->”指向的是原始文件路径。

扩展 2: 如何去重启单个网卡?

停止某个网卡: `#ifdown 网卡名`

开启某个网卡: `#ifup 网卡名`

例如: 需要停止-启动（重启）eth0 网卡，则可以输入

`#ifdown eth0`

`#ifup eth0`

提示: 在实际工作的时候不要随意禁网卡。

四、ssh 服务（重点）

ssh（secure shell，安全外壳协议），该协议有 2 个常用的作用：**远程连接协议**、远程文件传输协议。

协议使用**端口号**：默认是 22

可以是被修改的，如果需要修改，则需要修改 ssh 服务的配置文件：

#/etc/ssh/ssh_config

```
35 # StrictHostKeyChecking ask
36 # IdentityFile ~/.ssh/identity
37 # IdentityFile ~/.ssh/id_rsa
38 # IdentityFile ~/.ssh/id_dsa
39 # Port 22
40 # Protocol 2,1
41 # Cipher 3des
```

端口号可以修改，但是得注意 2 个事项：

- a. 注意范围，端口范围是从 0-65535；
- b. 不能使用别的服务已经占用的端口；

服务启动/停止/重启

#service sshd start/stop/restart

#/etc/init.d/sshd start/stop/restart

```
/etc/init.d/sshd
[root@localhost ~]# /etc/init.d/sshd restart
Stopping sshd: [ OK ]
Starting sshd: [ OK ]
[root@localhost ~]# service sshd restart
Stopping sshd: [ OK ]
Starting sshd: [ OK ]
[root@localhost ~]# _
```

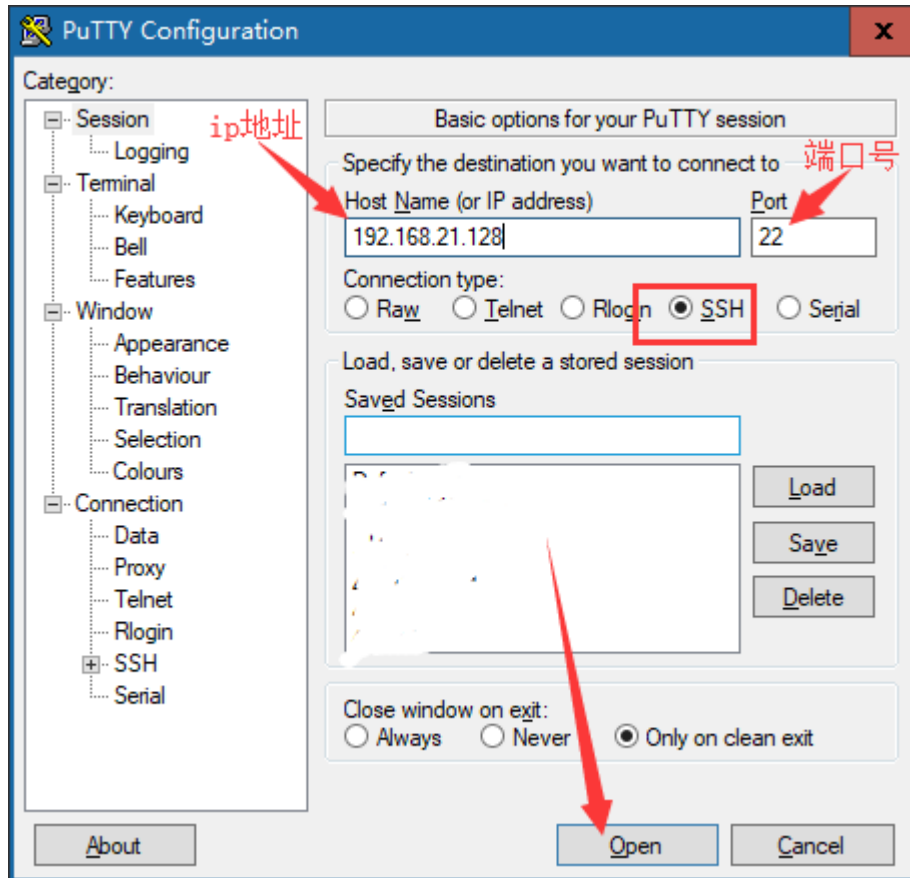
1、远程终端

终端工具主要帮助运维人员连接远程的服务器，常见终端工具有：Xshell、secureCRT、Putty 等。以 putty 为例：

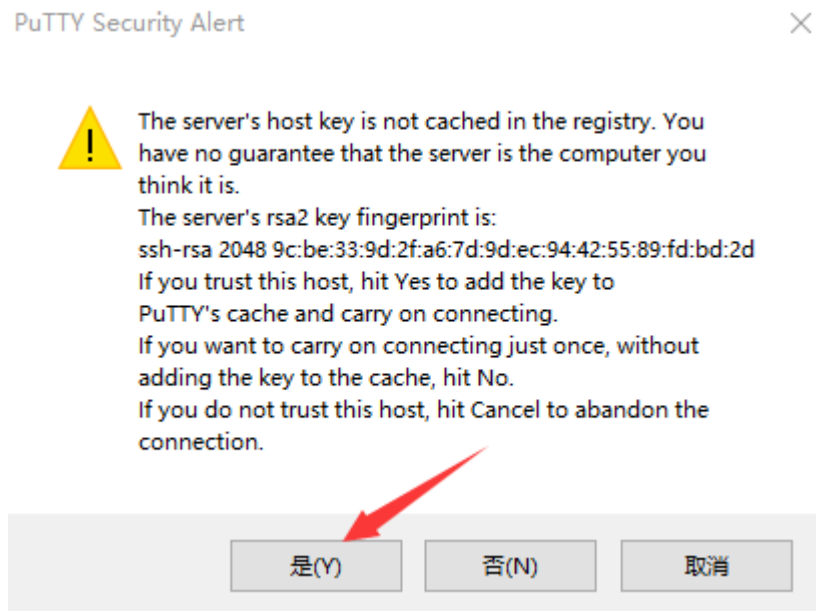
①获取服务器 ip 地址，可以通过 ifconfig 命令进行查看，然后顺手测试 ip 的连接相通性

```
C:\Users\admin>ping 192.168.21.128
正在 Ping 192.168.21.128 具有 32 字节的数据:
来自 192.168.21.128 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.21.128 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.21.128 的回复: 字节=32 时间<1ms TTL=64
```

②打开 putty，输入相关的信息



③在弹出 key 确认的时候点击“是”，以后不会再提示

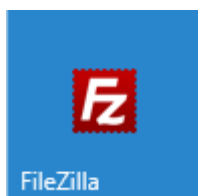


④输入登录信息

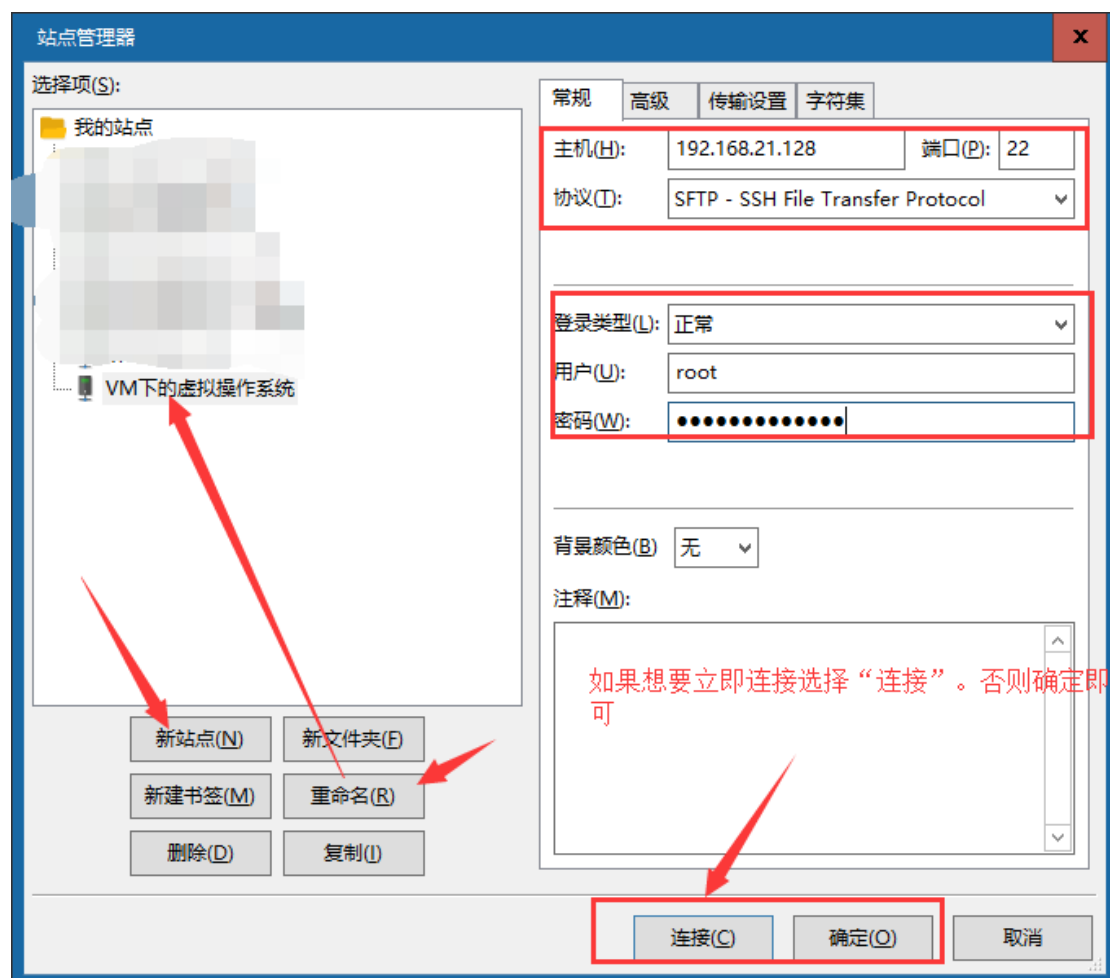
```
login as: root          输入用户名
root@192.168.21.128's password: 输入密码，密码没有提示，完成
Last login: Sun Mar 25 22:02:47 2018 按下回车
[root@localhost ~]#
```

2、SSH 服务文件传输

可视化的界面传输工具：Filezilla
安装好之后可以查看到桌面图标：

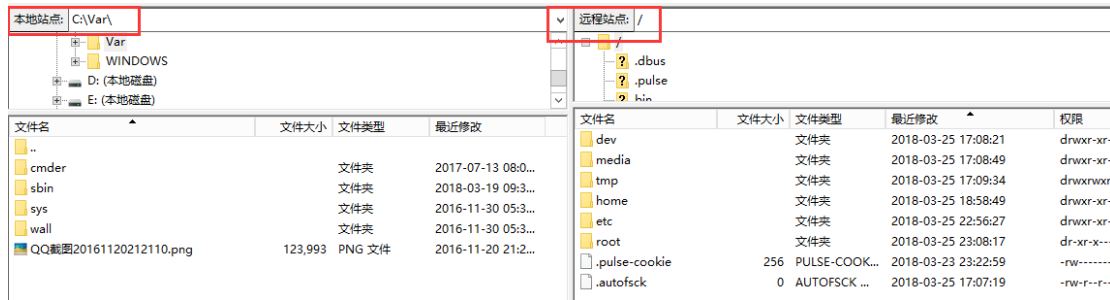


①选择“文件” - “站点管理器 (Ctrl + S)”



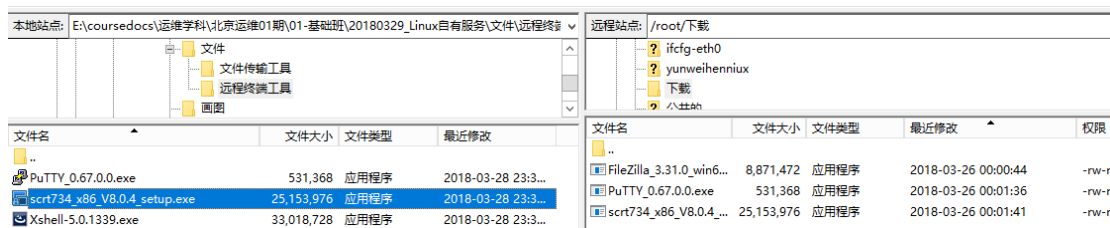


②点击“文件”菜单下方的“▽”选择需要连接的服务器，连接好之后的效果



③从本地 windows 上传文件到 linux 中方式

支持直接拖拽文件，也可以右键本地需要上传的文件，然后点选“上传”即可



④下载 linux 文件到本地

支持服务器文件直接拖拽到本地，也可以在右侧窗口选择需要下载的文件，右键，点选“下载”。

扩展 3：通过命令行工具来传输文件/文件夹

工具：PSCP.exe（必须通过 cmd 命令行打开），为了使用方便可以将其放到环境变量目录中如果不清楚哪些路径是环境变量路径，只需要将其放到 C:/Windows 目录下即可。

```
C:\Users\admin>pscp
PuTTY Secure Copy client
Release 0.63
Usage: pscp [options] [user@]host:source target
       pscp [options] source [source...] [user@]host:target
       pscp [options] -ls [user@]host:filespec
```

用法：

- a. pscp 选项 用户名@linux 主机地址:资源路径 windows 本地的地址（下载到 win）
- b. pscp 选项 资源路径 用户名@linux 主机地址:远程路径（上传到 linux）
- c. pscp 选项 -ls 用户名@linux 主机地址（列出远程路径下结构）

①下载到本地 windows

要求将远程 linux 服务器下的/etc 整个目录下载到本地 E:\tmp 下

```
#pscp -r root@192.168.21.128:/etc E:\tmp
```

在 CMD 中输入之后输入密码

```
... scp force use of scp protocol
C:\Users\admin>pscp -r root@192.168.21.128:/etc E:\tmp
root@192.168.21.128's password: █
```

crontab	8 kB	8.4 kB/s	ETA: 00:00:00	100%
named.conf	4 kB	4.2 kB/s	ETA: 00:00:00	100%
nta-newaliasesman	0 kB	0.0 kB/s	ETA: 00:00:00	100%
libxtables4.i386	28 kB	28.8 kB/s	ETA: 00:00:00	100%
orbd.1.gz	3 kB	3.9 kB/s	ETA: 00:00:00	100%
print-lprm	9 kB	9.5 kB/s	ETA: 00:00:00	100%
libiptc000.i386	2 kB	2.5 kB/s	ETA: 00:00:00	100%
print-lprman	1 kB	1.4 kB/s	ETA: 00:00:00	100%
sbin-iptables-save.i386	56 kB	56.4 kB/s	ETA: 00:00:00	100%
libip6tc0.i386	25 kB	26.0 kB/s	ETA: 00:00:00	100%
orbd	6 kB	6.5 kB/s	ETA: 00:00:00	100%
rmiregistry	6 kB	6.3 kB/s	ETA: 00:00:00	100%
rmid	6 kB	6.3 kB/s	ETA: 00:00:00	100%
unpack200.1.gz	1 kB	1.7 kB/s	ETA: 00:00:00	100%
...	ETA: 00:00:00	100%

②上传文件到 linux

将“E:\coursedocs\运维学科\北京运维 01 期\01-基础班\20180329_Linux 自有服务”所有的内容传输到 linux 下 root 用户的家目录

```
#pscp -r "E:\coursedocs\运维学科\北京运维 01 期\01-基础班\20180329_Linux 自有服务"
root@192.168.21.128:/root
```

```
pscp: E:\coursedocs\运维学科\北京运维01期\01-基础班\20180329_Linux自有服务\笔记\WRL0001.
备份属于 笔记.wbk | 521 kB | 521.5 kB/s | ETA: 00:00:00 | 100%
pscp: E:\coursedocs\运维学科\北京运维01期\01-基础班\20180329_Linux自有服务\笔记\笔记.doc:
01. linux的运行级别.wmv | 73633 kB | 12272.3 kB/s | ETA: 00:00:00 | 100%
02. 用户管理 (1).wmv | 78605 kB | 19651.4 kB/s | ETA: 00:00:00 | 100%
03. 用户管理 (2).wmv | 108634 kB | 13579.3 kB/s | ETA: 00:00:00 | 100%
04. 用户组管理.wmv | 52736 kB | 13184.1 kB/s | ETA: 00:00:00 | 100%
05. 网络设置.wmv | 57920 kB | 28960.5 kB/s | ETA: 00:00:00 | 100%
06. 网络设置扩展.wmv | 16880 kB | 16880.8 kB/s | ETA: 00:00:00 | 100%
07. shell终端使用.wmv | 50910 kB | 12727.6 kB/s | ETA: 00:00:00 | 100%
08. 使用filezilla上传下载? | 38049 kB | 38049.5 kB/s | ETA: 00:00:00 | 100%
```

五、作业

1、能够分别使用 Filezilla 和 PSCP 工具传输给定文件到“/usr/src/data”目录下，如目录不存在则自行创建。

Linux 自有服务 (2)

自有服务，即不需要用户独立去安装的软件的服务，而是当系统安装好之后就可以直接使用的服务（内置）。

一、设置主机名

回顾：

```
#hostname  
#hostname -f FQDN（全限定域名）
```

①临时设置主机名（立竿见影），需要切换用户使之生效

#hostname 设置的主机名

```
[root@localhost ~]# hostname yunwei  
[root@localhost ~]# su  
[root@yunwei ~]#
```

②永久设置主机名（需要重启）

先找到一个文件

/etc/sysconfig/network 【主机名的配置文件】

```
[root@yunwei ~]# cat /etc/sysconfig/network  
NETWORKING=yes  
HOSTNAME=localhost.localdomain
```

修改其中的 HOSTNAME 为自己需要设置的永久主机名

```
1 NETWORKING=yes  
2 HOSTNAME=yunwei  
~  
~  
与临时的主机名一致即可
```

③修改 linux 服务器的 hosts 文件，将 yunwei 指向本地（设置 FQDN）

Hosts 文件的位置：/etc/hosts

```
1 127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4 yunwei  
2 ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
```

问题：不设置 FQDN 会怎么样？

- ①很多开源服务器软件（例如 Apache）则无法启动，或出现报错；
- ②方便记忆，看到主机名对其作用有一个初步判断；
- ③如果不设置则会影响本地的域名的解析（本地访问）；

二、chkconfig

作用：相当于 windows 下“安全卫士”、“电脑管家”之类的安全辅助工具提供“开机启动项”的一个管理服务。

在 linux 下不是所有的软件安装完成之后都有开机启动服务，有的可能需要自己去添加。除此之外还可以查看和删除。

①开机启动服务查询

#chkconfig --list

haldaemon	0:关闭	1:关闭	2:关闭	3:启用	4:启用	5:启用	6:关闭
htcacheclean	0:关闭	1:关闭	2:关闭	3:关闭	4:关闭	5:关闭	6:关闭
httpd	0:关闭	1:关闭	2:关闭	3:关闭	4:关闭	5:关闭	6:关闭
iptables	0:关闭	1:关闭	2:启用	3:启用	4:启用	5:启用	6:关闭
iptables	0:关闭	1:关闭	2:启用	3:启用	4:启用	5:启用	6:关闭
irqbalance	0:关闭	1:关闭	2:关闭	3:启用	4:启用	5:启用	6:关闭
kdump	0:关闭	1:关闭	2:启用	3:启用	4:启用	5:启用	6:关闭

其中 0-6 表示各个启动级别

例如：以 httpd 为例，其 3 级别为关闭 (off)，则表示其在 3 启动形式下默认开机不启动
5 对应的也是关闭，则表示其在桌面环境下也是开机不启动。

再例如：kdump 服务，在 2, 3, 4, 5 的级别下默认开机启动的，其他级别下默认开机不启动

②删除服务

#chkconfig --del 服务名

例如删除 httpd 服务

```
[root@yunwei ~]# chkconfig --list|grep httpd
httpd          0:关闭 1:关闭 2:关闭 3:关闭 4:关闭 5:关闭 6:关闭
[root@yunwei ~]# chkconfig --del httpd
[root@yunwei ~]# chkconfig --list|grep httpd
[root@yunwei ~]#
```

③添加开机启动服务

#chkconfig --add 服务名

【必须要保证服务正常运行，才可以添加】

```
[root@yunwei ~]# chkconfig --add httpd
[root@yunwei ~]# chkconfig --list|grep httpd
httpd          0:关闭 1:关闭 2:关闭 3:关闭 4:关闭 5:关闭 6:关闭
[root@yunwei ~]#
```

④设置服务在某个级别下开机启动/不启动【重点命令】

#chkconfig --level 连在一起的启动级别 服务名 on/off

案例：设置 httpd 服务在 3, 5 级别下默认开机启动

```
httpd      0:关闭 1:关闭 2:关闭 3:关闭 4:关闭 5:关闭 6:关闭
[root@yunwei ~]# chkconfig --level 35 httpd on
[root@yunwei ~]# chkconfig --list|grep httpd
httpd      0:关闭 1:关闭 2:关闭 3:启用 4:关闭 5:启用 6:关闭
[root@yunwei ~]#
```

案例：设置 httpd 服务在 5 的级别下默认开机不启动

```
[root@yunwei ~]# chkconfig --list|grep httpd
httpd      0:关闭 1:关闭 2:关闭 3:启用 4:关闭 5:启用 6:关闭
[root@yunwei ~]# chkconfig --level 5 httpd off
[root@yunwei ~]# chkconfig --list|grep httpd
httpd      0:关闭 1:关闭 2:关闭 3:启用 4:关闭 5:关闭 6:关闭
[root@yunwei ~]#
```

三、ntp 服务

作用：ntp 主要是用于对计算机的时间同步管理操作。

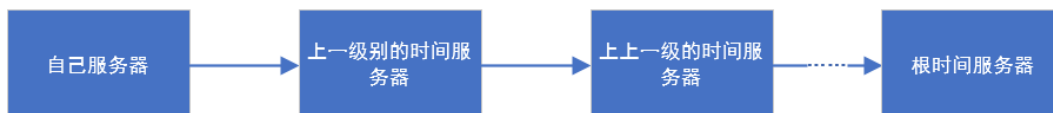
时间是对服务器来说是很重要的，一般很多网站都需要读取服务器时间来记录相关信息，如果时间不准，则可能造成很大的影响。

例如：当前虚拟机里的 linux 时间就是不准确的

```
[root@yunwei ~]# date +"%F %T"
2018-03-26 02:32:19
[root@yunwei ~]#
```

同时服务器时间方式有 2 个：一次性同步（手动同步）、通过服务自动同步。

上游的概念：



①一次性同步时间（简单）

#ntpd 时间服务器的域名或 ip 地址

ip 地址查看可以访问：<http://www.ntp.org.cn/pool.php>

```
2018-03-26 02:42:23
[root@yunwei ~]# ntpdate 120.25.108.11
30 Mar 10:55:59 ntpdate[2276]: step time server 120.25.108.11 offset 375169.715323 sec
[root@yunwei ~]# date +"%F %T"
2018-03-30 10:56:27
[root@yunwei ~]#
```

②设置时间同步服务

服务名: ntpd

启动 ntpd 服务

#service ntpd start 或者 /etc/init.d/ntpd start

```
[root@yunwei ~]# service ntpd start
正在启动 ntpd: [确定]
[root@yunwei ~]# date +"%F %T"
2018-03-26 02:39:14
[root@yunwei ~]# date +"%F %T"
2018-03-30 11:00:22
[root@yunwei ~]#
```

设置 ntpd 服务开机启动:

chkconfig --list|grep ntpd

chkconfig --level 35 ntpd on

```
[root@yunwei ~]# chkconfig --list|grep ntpd
ntpd          0:关闭 1:关闭 2:关闭 3:关闭 4:关闭 5:关闭 6:关闭
ntpdate      0:关闭 1:关闭 2:关闭 3:关闭 4:关闭 5:关闭 6:关闭
[root@yunwei ~]# chkconfig --level 35 ntpd on
[root@yunwei ~]# chkconfig --list|grep ntpd
ntpd          0:关闭 1:关闭 2:关闭 3:启用 4:关闭 5:启用 6:关闭
ntpdate      0:关闭 1:关闭 2:关闭 3:关闭 4:关闭 5:关闭 6:关闭
[root@yunwei ~]#
```

四、防火墙服务

防火墙: 防范一些网络攻击。有软件防火墙、硬件防火墙之分。



防火墙选择让请求通过, 从而保证网络安全性。

在当前的 centos6.5 中防火墙有一个名称: iptables 【7.x 中默认使用的是 firewalld】

①查看 iptables 是否开机启动



```
[root@yunwei ~]# chkconfig --list | grep iptables
iptables          0:关闭 1:关闭 2:启用 3:启用 4:启用 5:启用 6:关闭
[root@yunwei ~]#
```

②iptables 服务启动/重启/关闭

```
#service iptables start/restart/stop
/etc/init.d/iptables start /restart/stop
```

③查看 iptables 的状态（规则）

```
]# service iptables status
```

如果 iptables 没有启动，则提示服务没启动，如果已经启动，则显示防火墙的相关的规则信息

```
[root@yunwei ~]# service iptables status
表格: filter
Chain INPUT (policy ACCEPT)
num target      prot opt source                destination
1  ACCEPT        all  --  0.0.0.0/0             0.0.0.0/0           state RELATED,ESTABLISHED
2  ACCEPT        icmp --  0.0.0.0/0             0.0.0.0/0
3  ACCEPT        all  --  0.0.0.0/0             0.0.0.0/0
4  ACCEPT        tcp  --  0.0.0.0/0             0.0.0.0/0           state NEW tcp dpt:22
5  REJECT        all  --  0.0.0.0/0             0.0.0.0/0           reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
num target      prot opt source                destination
1  REJECT        all  --  0.0.0.0/0             0.0.0.0/0           reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
num target      prot opt source                destination
```

④查看规则的命令

```
#iptables -L -n
```

含义：

- L: 表示列出规则
- n: 表示将单词表达形式改成数字形式显示

⑤简单设置防火墙规则

例如，需要允许 80 端口通过防火墙，则规则可以用以下的命令来设置

```
#iptables -I INPUT -p tcp --dport 80 -j ACCEPT #允许访问 80 端口
```

iptables: 主命令

-I: 表示将规则放到最前面

-A: add, 添加规则 (最后)

INPUT: 进站请求【出站 output】

-p: protocol, 指定协议 (icmp/tcp/udp)

--dport: 指定端口号

-j: 指定行为结果, 允许 (accept) /禁止 (reject) /丢弃 (drop)

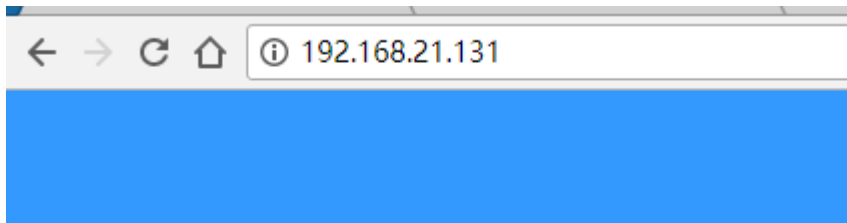
```
[root@yunwei ~]# iptables -L -n
Chain INPUT (policy ACCEPT)                                -I的效果
target prot opt source destination
ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:80
ACCEPT all -- 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
ACCEPT icmp -- 0.0.0.0/0 0.0.0.0/0
ACCEPT all -- 0.0.0.0/0 0.0.0.0/0
ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 state NEW tcp dpt:22
REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibit
ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:80          -A的效果
Chain FORWARD (policy ACCEPT)
```

添加完成之后需要保存操作:

/etc/init.d/iptables save

```
[root@yunwei ~]# /etc/init.d/iptables save
iptables: 将防火墙规则保存到 /etc/sysconfig/iptables: [确定]
[root@yunwei ~]#
```

测试 80 端口访问:



五、rpm 管理（重点）

作用: rpm 的作用类似于 windows 上的电脑管家中“软件管理”、安全卫士里面“软件管家”等产品, 主要作用是对 linux 服务器上的软件包进行对应管理操作, 管理分为: 查询、卸载、安装。

①查询某个软件的安装情况

#rpm -qa|grep 关键词

选项:

-q: 查询, query

-a: 全部, all

案例: 查询 linux 上是否安装 firefox

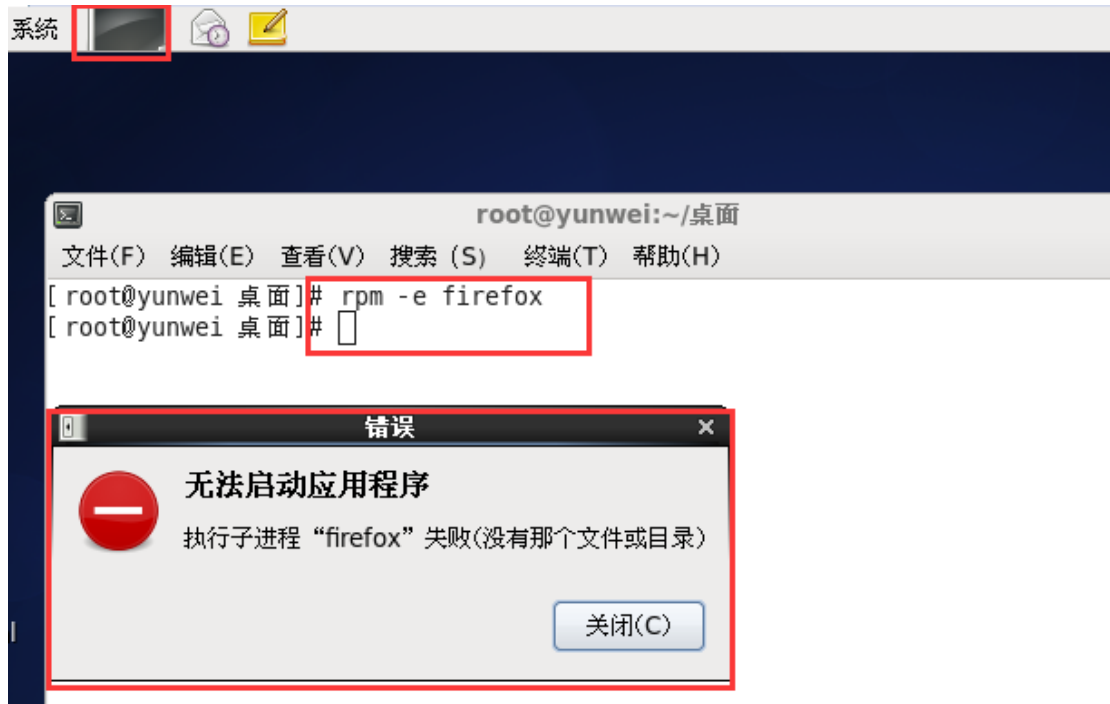
```
[root@yunwei ~]# rpm -qa|grep firefox
firefox-17.0.10-1.el6.centos.i686
```

案例: 查询是否安装 qq

```
[root@yunwei ~]# rpm -qa|grep qq
[root@yunwei ~]#
```


②卸载某个软件

#rpm -e 软件的名称



火狐卸载的时候是没有依赖关系的，所以可以直接卸载。

但是在卸载 Apache 的时候提示无法卸载：

```
[root@yunwei 桌面]# rpm -e httpd
error: Failed dependencies:
    httpd >= 2.2.0 is needed by (installed) gnome-user-share-2.28.2-3.el6.i686
[root@yunwei 桌面]#
```

当存在依赖关系的时候又不想去解决这个问题的时候可以：

```
#rpm -e 软件包名 --nodeps
[root@yunwei 桌面]# rpm -e httpd --nodeps
[root@yunwei 桌面]# rpm -qa|grep httpd
httpd-tools-2.2.15-29.el6.centos.i686
[root@yunwei 桌面]#
```

③软件的安装

要想装软件，和 windows 下一样，先得找到安装包。

软件包的获得方式：

- a. 去官网去下载；
- b. 不介意老版本的话，可以从光盘（或者镜像文件）中读取；

此处以光盘文件为例：

查看块状设备的信息：

#lsblk (list block devices) 查看块状设备的信息

```
[root@yunwei ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO  TYPE MOUNTPOINT
sr0                                  11:0    1   3.6G  0   rom  /media/CentOS_6.5_Final
sda                                  8:0     0   20G   0   disk
├─sda1                               8:1     0   500M  0   part /boot
└─sda2                               8:2     0  19.5G  0   part
   ├─VolGroup-lv_root (dm-0) 253:0    0  17.5G  0   lvm  /
   └─VolGroup-lv_swap (dm-1) 253:1    0    2G   0   lvm  [SWAP]
[root@yunwei ~]#
```

Name: 名称

Size: 设备大小

Type: 类型

MountPoint: 挂载点（类似 windows 下盘符）

扩展：光盘的挂载和解挂

a. 解挂操作

命令: umount

语法: #umount 当前设备的挂载点（路径）

```
[root@yunwei ~]# umount /media/CentOS_6.5_Final
[root@yunwei ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO  TYPE MOUNTPOINT
sr0                                  11:0    1   3.6G  0   rom
sda                                  8:0     0   20G   0   disk
├─sda1                               8:1     0   500M  0   part /boot
└─sda2                               8:2     0  19.5G  0   part
   ├─VolGroup-lv_root (dm-0) 253:0    0  17.5G  0   lvm  /
   └─VolGroup-lv_swap (dm-1) 253:1    0    2G   0   lvm  [SWAP]
[root@yunwei ~]#
```

此时，相当于 U 盘在 windows 上已经被弹出了，但是没有拔下电脑 USB 接口。

b. 挂载光盘

命令: mount

语法: #mount 设备原始地址 要挂载的位置路径

设备原始地址: 地址统一都在/dev下，然后根据大小确定具体 name 值，拼凑在一起组成原始地址，例如当前: “/dev/sr0”

要挂载的位置路径: 挂载目录一般都在 mnt 下，也可以在 mnt 下建目录，此处以 “/mnt/dvd” 为例

```
[root@yunwei ~]# mkdir /mnt/dvd
[root@yunwei ~]# mount /dev/sr0 /mnt/dvd
mount: block device /dev/sr0 is write-protected, mounting read-only
[root@yunwei ~]#
```

安装软件的命令:

#rpm -ivh 软件包完整名称

选项:

-i: install, 安装

-v: 显示进度条

-h: 表示以 “#” 形式显示进度条



```
[root@yunwei Packages]# pwd
/mnt/dvd/Packages
[root@yunwei Packages]# rpm -ivh firefox-17.0.10-1.el6.centos.i686.rpm
Preparing... ##### [100%]
 1:firefox ##### [100%]
[root@yunwei Packages]# rpm -qa|grep firefox
firefox-17.0.10-1.el6.centos.i686
[root@yunwei Packages]#
```

六、cron/crontab 计划任务（重点）

作用：操作系统不可能 24 小时都有人在操作，有些时候想在指定的时间点去执行任务（例如：每天夜里 2 点去重新启动 Apache），此时不可能真有人每天夜里 2 点去执行命令，此时可以交给计划任务程序去执行操作。

语法：#crontab 选项

常用选项：

- l: list, 列出指定用户的计划任务列表
- e: edit, 编辑指定用户的计划任务列表
- u: user, 指定的用户名，如果不指定，则表示当前用户
- r: remove, 删除指定用户的计划任务列表

①列出

```
[root@yunwei ~]# crontab -l
no crontab for root
[root@yunwei ~]#
```

②编辑计划任务（重点）

计划任务的规则语法格式，以行为单位，一行则为一个计划：

分 时 日 月 周 需要执行的命令

例如：如果想要每天的 0 点 0 分执行 reboot 指令，则可以写成

00 * * * reboot

取值范围：

分：0~59

时：0~23

日：1~31

月：1~12

周：0~7，0 和 7 表示星期天

四个符号：

*: 表示取值范围中的每一个数字

-: 做连续区间表达式的，要想表示 1~7，则可以写成：1-7

/: 表示每多少个, 例如: 想每 10 分钟一次, 则可以在分的位置写: */10
,: 表示多个取值, 比如想在 1 点, 2 点 6 点执行, 则可以在时的位置写: 1,2,6

问题 1: 每月 1、10、22 日的 4:45 重启 network 服务
45 4 1,10,22 * * service network restart

问题 2: 每周六、周日的 1:10 重启 network 服务
10 1 * * 6,0 service network restart

问题 3: 每天 18:00 至 23:00 之间每隔 30 分钟重启 network 服务
*/30 18-23 * * * service network restart

问题 4: 每隔两天的上午 8 点到 11 点的第 3 和第 15 分钟执行一次重启
3,15 8-11 */2 * * reboot

案例: 真实测试案例, 每 1 分钟往 root 家目录中的 RT.txt 中输入当前的时间信息, 为了看到效果使用追加输出

计划任务: */1 * * * * ls ~>> /root/RT.txt

Crontab 权限问题: 本身是任何用户都可以创建自己的计划任务。

但是超级管理员可以通过配置来设置某些用户不允许设置计划任务 :
配置文件位于 (黑名单):

/etc/cron.deny 里面写用户名, 一行一个

```
1 yw
```

```
[yw@yunwei root]$ crontab -e
You (yw) are not allowed to use this program (crontab)
See crontab(1) for more information
```

还有一个配置文件: (白名单)

/etc/cron.allow (本身不存在, 自己创建)

注意: 白名单优先级高于黑名单, 如果一个用户同时存在两个名单文件中, 则会被默认允许创建计划任务。

Linux 的权限管理操作

Linux 的权限操作与用户、用户组是兄弟操作。

一、权限概述

总结：Linux 系统一般将文件可存/取访问的身份分为 3 个类别：owner、group、others，且 3 种身份各有 read、write、execute 等权限。

1、权限介绍

什么是权限？

在多用户（可以不同时）计算机系统的管理中，权限是指某个特定的用户具有特定的系统资源使用**权力**，像是文件夹、特定系统指令的使用或存储量的限制。

在 Linux 中分别有读、写、执行权限：

读权限：

对于文件夹来说，读权限影响用户是否能够列出目录结构

对于文件来说，读权限影响用户是否可以查看文件内容

写权限：

对于文件夹来说，写权限影响用户是否可以在文件夹下“创建/删除/复制到/移动到”文档

对于文件来说，写权限影响用户是否可以编辑文件内容

执行权限：

一般都是对于文件来说，特别脚本文件。

2、身份介绍

Owner 身份（文件所有者，默认为文档的创建者）

由于 Linux 是多用户、多任务的操作系统，因此可能常常有多人同时某台主机上工作，但每个人均可在主机上设置文件的权限，让其成为个人的“私密文件”，即个人所有者。因为设置了适当的文件权限，除本人（文件所有者）之外的用户无法查看文件内容。

例如某个 MM 给你发了一封 Email 情书，你将情书转为文件之后存档在自己的主文件夹中。为了不让别人看到情书的内容，你就能利用所有者的身份去设置文件的适当权限，这样，即使你的情敌想偷看你的情书内容也是做不到的。

Group 身份（与文件所有者同组的用户）

与文件所有者同组最有用的功能就体现在**多个团队在同一台主机上开发资源的时候**。例如主机上有 A、B 两个团体，A 中有 a1,a2,a3 三个成员，B 中有 b1,b2 两个成员，这两个团体要共同完成一份报告 F。由于设置了适当的权限，A、B 团体中的成员都能互相修改对方的数据，但是团体 C 的成员则不能修改 F 的内容，甚至连查看的权限都没有。同时，团体的成员也能设置自己的私密文件，让团队的其它成员也读取不了文件数据。**在 Linux 中，每个账户支持多个用户组。如用户 a1、b1 即可属于 A 用户组，也能属于 B 用户组【主组和附加组】。**

Others 身份（其他人，相对于所有者）

这个是个相对概念。打个比方，大明、二明、小明一家三兄弟住在一间房，房产证上的登记者是大明（owner 所有者），那么，大明一家就是一个用户组，这个组有大明、二明、小明三个成员；另外有个人叫张三，和他们三没有关系，那么这个张三就是其他人了。

同时，大明、二明、小明有各自的房间，三者虽然能自由进出各自的房间，但是小明不能让大明看到自己的情书、日记等，这就是文件所有者（用户）的意义。

Root 用户（超级用户）

在 Linux 中，还有一个神一样存在的用户，这就是 root 用户，因为在所有用户中它拥有最大的权限，所以管理着普通用户。

3、Linux 的权限介绍

要设置权限，就需要知道文件的一些基本属性和权限的分配规则。在 Linux 中，ls 命令常用来查看文件的属性，用于显示文件的文件名和相关属性。

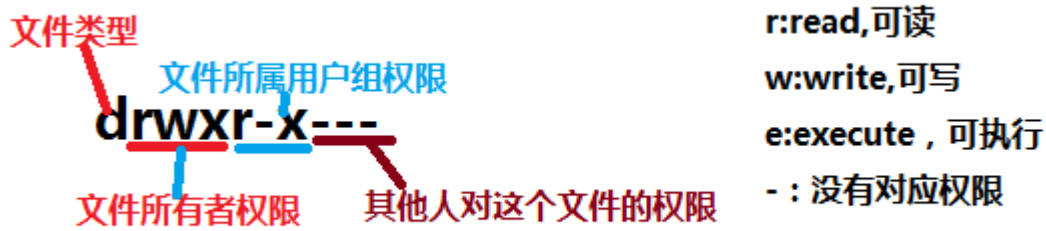
#ls -l 路径 【ls -l 等价于 ll】

```
[root@yunwei ~]# ls -l ~
总用量 128
-rw-----. 1 root root 1641 3月 23 23:17 anaconda-ks.cfg
-rw-r--r--. 1 root root 807 3月 25 09:04 group
-rw-r--r--. 1 root root 33 3月 24 14:32 hebing.txt
-rw-r--r--. 1 root root 64 3月 25 12:52 hello.c
lrwxrwxrwx. 1 root root 41 3月 25 22:32 ifcfg-eth0 -> /etc/sy
-rw-r--r--. 1 root root 45503 3月 23 23:17 install.log
-rw-r--r--. 1 root root 10033 3月 23 23:16 install.log.syslog
-rw-r--r--. 1 root root 1639 3月 25 16:37 passwd
-rw-r--r--. 1 root root 1631 3月 25 14:04 password
-rw-r--r--. 1 root root 1631 3月 25 08:38 sss.txt
```

标红的部分就是 Linux 的文档权限属性信息。



Linux 中存在用户、用户组和其他人概念，各自有不同的权限，对于一个文档来说，其权限具体分配如下：



十位字符表示含义：

第 1 位：表示文档类型，取值常见的有“d 表示文件夹”、“-表示文件”、“l 表示软连接”、“s 表示套接字”等等；

第 2-4 位：表示文档所有者的权限情况，第 2 位表示读权限的情况，取值有 r、-；第 3 位表示写权限的情况，w 表示可写，-表示不可写，第 4 位表示执行权限的情况，取值有 x、-。

第 5-7 位：表示与所有者同在一个组的用户的权限情况，第 5 位表示读权限的情况，取值有 r、-；第 6 位表示写权限的情况，w 表示可写，-表示不可写，第 7 位表示执行权限的情况，取值有 x、-。

第 8-10 位：表示除了上面的前 2 部分的用户之外的其他用户的权限情况，第 8 位表示读权限的情况，取值有 r、-；第 9 位表示写权限的情况，w 表示可写，-表示不可写，第 10 位表示执行权限的情况，取值有 x、-。

权限分配中,均是 rwx 的三个参数组合，且位置顺序不会变化。没有对应权限就用 - 代替。

例如：以下一个文档权限是怎么样的？

```
drwxr-xr-x. 3 root root 4096 3月 24 10:53 yunweihenniu
```

- a. 其是文件夹类型
- b. 所有者：拥有全部权限（读写执行）
- c. 同组用户：可读、可执行
- d. 其他用户：可读、可执行

二、权限设置

语法：#chmod 选项 权限模式 文档

注意事项：

常用选项：

-R: 递归设置权限（当文档类型为文件夹的时候）

权限模式：就是该文档需要设置的权限信息

文档：可以是文件，也可以是文件夹，可以是相对路径也可以是绝对路径。

注意点：如果想要给文档设置权限，操作者要么是 root 用户，要么就是文档的所有者。

1、字母形式

选项	字母	介绍
(谁)	u	用户
(谁)	g	所属群体
(谁)	o	其他人
(谁)	a	所有人 (“全部”)
(作用)	+	增加权限
(作用)	-	减少权限
(作用)	=	确定权限
(权限)	r	可读
(权限)	w	可写
(权限)	x	执行

给谁设置：

- u: 表示所有者身份 owner (user)
- g: 表示给所有者同组用户设置 (group)
- o: 表示 others, 给其他用户设置权限
- a: 表示 all, 给所有人 (包含 ugo 部分) 设置权限

如果在设置权限的时候不指定给谁设置, 则默认给所有用户设置

权限字符：

- r: 读
- w: 写
- x: 表示执行
- : 表示没有权限

权限分配方式：

- +: 表示给具体的用户新增权限 (相对当前)
- : 表示删除用户的权限 (相对当前)
- =: 表示将权限设置成具体的值 (注重结果)【赋值】

例如：需要给 `anaconda-ks.cfg` 文件 (`-rw-----`) 设置权限，要求所有者拥有全部的权限，同组用户拥有读和执行权限，其他用户只读权限。

答案：

①#`chmod u+x,g+rx,o+r anaconda-ks.cfg`

```
[root@yunwei ~]# ll anaconda-ks.cfg
-rw-----. 1 root root 1641 3月 23 23:17 anaconda-ks.cfg
[root@yunwei ~]# chmod u+x,g+rx,o+r anaconda-ks.cfg
[root@yunwei ~]# ll anaconda-ks.cfg
-rwxr-xr--. 1 root root 1641 3月 23 23:17 anaconda-ks.cfg
```

②#`chmod u=rwx,g=rx,o=r anaconda-ks.cfg`



```
[root@yunwei ~]# ll anaconda-ks.cfg
-rw-----. 1 root root 1641 3月 23 23:17 anaconda-ks.cfg
[root@yunwei ~]# chmod u=rwx,g=rx,o=r anaconda-ks.cfg
[root@yunwei ~]# ll anaconda-ks.cfg
-rwxr-xr--. 1 root root 1641 3月 23 23:17 anaconda-ks.cfg
```

提示：当文档拥有**执行**权限（任意部分），则其颜色在终端中是绿色。

#chmod ug=rwx 形式，如果有两部分权限一样则可以合在一起写的

例如：如果 anaconda-ks.cfg 文件什么权限都没有，可以使用 root 用户设置所有人都有执行权限，则可以写成

- ①#chmod +x anaconda-ks.cfg
- ②#chmod a=x anaconda-ks.cfg
- ③#chmod a+x anaconda-ks.cfg

2、数字形式

经常会在一些技术性的网页上看到类似于#chmod 777 a.txt 这样的权限，这种形式称之为数字形式权限（777）。

- 读：r 4
- 写：w 2
- 执行：x 1

没有任何权限：0

数值	权限	目录列表
0	不能读，不能写，不能执行	---
1	不能读，不能写，可执行	--x
2	不能读，可写，不能执行	-w-
3	不能读，可写，可执行	-wx
4	可读，不能写，不能执行	r--
5	可读，不能写，可执行	r-x
6	可读，可写，不能执行	rw-
7	可读，可写，可执行	rwx

例如：需要给 anaconda-ks.cfg 设置权限，权限要求所有者拥有全部权限，同组用户拥有读执行权限，其他用户只读。

全部权限（u）：读+写+执行=4+2+1=7

读和执行（g）：读+执行=4+1=5

读权限（o）：读=4

由上得知权限为：754

#chmod 754 anaconda-ks.cfg

```
[root@yunwei ~]# ll anaconda-ks.cfg
-----x. 1 root root 1641 3月 23 23:17 anaconda-ks.cfg
[root@yunwei ~]# chmod 754 anaconda-ks.cfg
[root@yunwei ~]# ll anaconda-ks.cfg
-rwxr-xr--. 1 root root 1641 3月 23 23:17 anaconda-ks.cfg
[root@yunwei ~]#
```

面试题：用超级管理员设置文档的权限命令是#chmod -R 731 aaa，请问这个命令有没有什么不合理的地方？

拥有者：7=4+2+1=读+写+执行

同组用户：3=2+1=写+执行

其他用户：1=1=执行

注意：在写权限的时候千万不要设置类似于上面的这种“奇葩权限”。如果一个权限数字中但凡出现 2 与 3 的数字，则该权限有不合理的情况。

3、注意事项

使用 root 用户创建一个文件夹 (/oo)，权限默认，权限如下：

```
drwxr-xr-x. 2 root root 0 3月 30 15:07 net
drwxr-xr-x. 2 root root 4096 4月 1 14:43 oo
drwxr-xr-x. 3 root root 4096 3月 23 23:16 opt
```

需要在 oo 目录下创建文件 (oo/xx.txt)，需要给 777 权限：

```
[root@yunwei /]# touch oo/xx.txt
[root@yunwei /]# ls -l oo/
总用量 0
-rw-r--r--. 1 root root 0 4月 1 14:45 xx.txt
[root@yunwei /]# chmod 777 oo/xx.txt
[root@yunwei /]# ls -l oo/
总用量 0
-rwxrwxrwx. 1 root root 0 4月 1 14:45 xx.txt
```

切换到 test 用户（不是文档所有者，也不是同组用户，属于 other 部分）：

问题 1：test 用户是否可以打开 oo/xx.txt 文件？【能打开】

问题 2：test 用户是否可以编辑 oo/xx.txt 文件？【可以】

问题 3：test 用户是否可以删除 oo/xx.txt 文件？【不可以，同样还不允许创建文件/文件夹、移动文件、重命名文件】

```
[test@yunwei /]# rm -f oo/xx.txt
rm: 无法删除"oo/xx.txt": 权限不够
[test@yunwei /]# touch oo/aa.txt
touch: 无法创建"oo/aa.txt": 权限不够
[test@yunwei /]# mv oo/xx.txt /
mv: 无法将"oo/xx.txt" 移动至"/xx.txt": 权限不够
```



在 Linux 中，如果要删除一个文件，不是看文件有没有对应的权限，而是看文件所在的目录是否有写权限，如果有才可以删除。

三、属主与属组设置

属主：所属的用户（文件的主人）

属组：所属的用户组

```
drwxr-xr-x.  3 root root  4096 4月  1 11:38 home
dr-xr-xr-x. 18 root root 12288 3月  23 16:12 lib
drwx-----. 2 root root 16384 3月  23 22:58 lost+found
drwxr-xr-x.  3 root root  4096 3月  30 14:57 media
drwxr-xr-x.  2 root root    0 3月  30 15:07 misc
drwxr-xr-x.  2 root root  4096 9月  23 2011 mnt
drwxr-xr-x.  2 root root    0 3月  30 15:07 net
drwxr-xrwx.  2 root root  4096 4月  1 14:45 oo
```

所属用户 指向 root (属主)
用户组 指向 root (属组)

前面的那个 root 就是属主

后面的那个 root 就是属组

这两项信息在文档创建的时候会使用创建者的信息（用户名、用户所属的主组名称）。

如果有时候去删除某个用户，则该用户对应的文档的属主和属组信息就需要去修改。

1、chown（重点）

作用：更改文档的所属用户

语法：#chown -R username 文档路径

案例：将刚才 root 用户创建的 oo 目录，所有者更改为 test

#chown test oo/

```
drwxr-xr-x.  2 root root    0 3月  30 15:07 net
drwxr-xrwx.  2 test root  4096 4月  1 14:54 oo
drwxr-xr-x.  3 root root  4096 3月  23 23:16 opt
```

2、chgrp（了解）

作用：更改文档的所属用户组

语法：#chgrp -R groupname 文档的路径

案例：将刚才 root 用户创建的 oo 目录，所有者更改为 test，并且将所属用户组也改为 test

#chgrp test oo/

```
drwxr-xr-x.  2 root root      0 3月  30 15:07 net
drwxr-xrwx.  2 test test    4096 4月   1 14:54 oo
drwxr-xr-x.  3 root root    4096 3月  23 23:16 opt
```

思考，如何通过一个命令实现既可以更改所属的用户，也可以修改所属的用户组呢？

答：可以实现的，通过 chown 命令

```
语法：#chown -R username:groupname 文档路径
```

案例：要求只使用 chown 指令，将 oo 目录的所属用户和用户组改回成 root，并且包含其子目录

```
drwxr-xr-x.  22 root root    4096 3月  23 23:16 var
[root@yunwei /]# chown -R root:root oo/
[root@yunwei /]# ll
```

四、扩展（1）

问题：reboot、shutdown、init、halt、user 管理，在普通用户身份上都是操作不了，但是有些特殊的情况下又需要有执行权限。又不可能让 root 用户把自己的密码告诉普通用户，这个问题该怎么解决？

该问题是可以被解决的，可以使用 sudo（switch user do）命令来进行权限设置。Sudo 可以让管理员（root）事先定义某些特殊命令谁可以执行。

默认 sudo 中是没有除 root 之外用户的规则，要想使用则先配置 sudo。

Sudo 配置文件：/etc/sudoers

```
[root@yunwei /]# vim /etc/sudoers
[root@yunwei /]#
```

a. 配置 sudo 文件请使用“#visudo”，打开之后其使用方法和 vim 一致

b. 配置普通用户的权限

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

Root 表示用户名，如果是用户组，则可以写成“%组名”

ALL: 表示允许登录的主机（地址白名单）



(ALL): 表示以谁的身份执行，ALL 表示 root 身份

ALL: 表示当前用户可以执行的命令，多个命令可以使用“,”分割

案例：本身 test 用户不能添加用户，要求使用 sudo 配置，将其设置为可以添加用户，并且可以修改密码（但是不能修改 root 用户密码）。

注意：在写 sudo 规则的时候不建议写直接形式的命令，而是写命令的完整路径。

路径可以使用 which 命令来查看

语法：#which 指令名称

```
[No write since last change]
/usr/sbin/useradd
```

```
Press ENTER or type command to continue
[No write since last change]
/usr/bin/passwd
```

```
root    ALL=(ALL)    ALL
#要求test用户可以在任何主机上以root身份进行添加用户和设置密码操作（但是不能修改root用户密码）
test    ALL=(ALL)    /usr/sbin/useradd,/usr/bin/passwd
## Allows members of the 'sys' group to run networking, software,
```

在添加好对应的规则之后就可以切换用户，切换到普通用户 test，再去执行：

```
[root@yunwei ~]# su test
[test@yunwei ~]$ useradd aop
bash: /usr/sbin/useradd: 权限不够
[test@yunwei ~]$
```

此时要想使用刚才的规则，则以以下命令进行：

#sudo 需要执行的指令

```
[test@yunwei ~]$ sudo useradd aop
[sudo] password for test:
[test@yunwei ~]$ tail -1 /etc/passwd
aop:x:502:502:~/home/aop:/bin/bash
[test@yunwei ~]$ sudo useradd aopp
[test@yunwei ~]$ tail -1 /etc/passwd
aopp:x:503:503:~/home/aopp:/bin/bash
[test@yunwei ~]$
```

在输入 sudo 指令之后需要输入当前的用户密码进行确认的操作（不是 root 用户密码），输入之后在接下来 5 分钟内再次执行 sudo 指令不需要密码。

特别注意：此处按照案例要求，不能让 test 用户修改 root 密码，因此规则还需要调整，不然其可以修改 root 密码的：

禁止修改 root 密码的配置（先允许全部，再拒绝 root 密码设置）：`/usr/bin/passwd [A-Za-z]*, !/usr/bin/passwd root`

```
root    ALL=(ALL)        ALL
#要求test用户可以在任何主机上以root身份进行添加用户和设置密码操作（但不能修改root用户密码）
test    ALL=(ALL)        /usr/sbin/useradd, /usr/bin/passwd [A-Za-z]*, !/usr/bin/passwd root
## Allows members of the 'sys' group to run networking, software,
```

补充：在普通用户下怎么查看自己具有哪些特殊权限呢？

#sudo -l

```
[test@yunwei root]$ sudo -l
匹配此主机上 test 的默认条目:
    requiretty, !visiblepw, always_set_home, env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE
    KDEDIR LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE", env_keep
    LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_P
    LC_TELEPHONE", env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

用户 test 可以在该主机上运行以下命令:
    (ALL) /usr/sbin/useradd, (ALL) /usr/bin/passwd [A-Za-z]*, (ALL) !/usr/bin/passwd root
[test@yunwei root]$
```

最后：sudo 不是任何 Linux 分支都有的命令，常见 centos 与 ubuntu 都存在 sudo 命令。

作业：给普通用户设置一个关机命令执行权限。

Linux 的网络基础

一、网络相关概述

1、网络发展

信息传递

- 远古时期，人们就通过简单的语言、壁画等方式交换信息
- 千百年来，人们一直在用语言、图符、钟鼓、烟火、竹简、纸书等传递信息
- 古代人的烽火狼烟、飞鸽传信、驿马邮递
- 现代社会中，交警的指挥手语、航海中的旗语等
- 这些信息传递的基本方式都是依靠人的视觉与听觉

电的产生

- 1831 年，法拉第制出了世界上最早的第一台发电机
- 1866 年，德国人西门子（Siemens）制成世界上第一台大功率发电机
- 1837 年，美国人塞缪乐·莫尔斯成功地研制出世界上第一台电磁式电报机
- 1844 年 5 月 24 日，莫尔斯在国会大厦联邦最高法院会议厅进行了“用莫尔斯电码”发出了人类历史上的第一份电报，从而实现了长途电报通信

网络诞生

- 1957 年，前苏联发射了第一颗人造卫星，震惊了美国
- 1958 年美国成立了国防部**高级研究计划署**（ARPA, Advanced Research Projects Agency），应对冷战形势，ARPA 是一个管理机构，没有实验室和科学家



- 1969 年，ARPANET（阿帕网）开始联机，因此 1969 年被称为 **Internet 元年**

网络分类（记忆）

- **局域网**（Local Area Network, **LAN**）是指范围在几百米到十几公里内办公楼群或校园内的计算机相互连接所构成的计算机网络。
- **城域网**（Metropolitan Area Network, **MAN**）所采用的技术基本上与局域网相类似，只是规模上要大一些。城域网既可以覆盖相距不远的几栋办公楼，也可以覆盖一个城。
- **广域网**（Wide Area Network, **WAN**）通常跨接很大的物理范围，如一个国家。

除了上述的划分，网络还可以按照所有者分为公网、私网是两种 Internet 的接入方式。
公网接入方式：上网的计算机得到的 IP 地址是 Internet 上的非保留地址，公网的计算机和 Internet 上的其他计算机可随意互相访问。私网则反之。

2、ip 地址（重点记忆）

IP 是英文 Internet Protocol 的缩写，意思是“网络之间互连的协议”，也就是为计算机网络相互连接进行通信而设计的**协议**。

IP 地址类型分为：**公有地址、私有地址**。

公有地址

公有地址（Public address）由 Inter NIC（Internet Network Information Center 因特网信息中心）负责。这些 IP 地址分配给注册并向 Inter NIC 提出申请的组织机构。通过它直接访问因特网。

私有地址（重点）

私有地址（Private address）属于**非注册地址**，专门为组织机构内部使用。以下列出留用的

内部私有地址:

A 类 10.0.0.0--10.255.255.255

B 类 172.16.0.0--172.31.255.255

C 类 192.168.0.0--192.168.255.255

IP 地址按类型可以分为三类:

类别	最大网络数	IP 地址范围	最大主机数	私有 IP 地址范围
A	126 (2^7-2)	1.0.0.0-127.255.255.255	16777214	10.0.0.0-10.255.255.255
B	16384(2^{14})	128.0.0.0-191.255.255.255	65534	172.16.0.0-172.31.255.255
C	2097152(2^{21})	192.0.0.0-223.255.255.255	254	192.168.0.0-192.168.255.255

网络运维相关技能: ip 分类、**子网划分**、划分 vlan、ACL、综合布线、各种 Serve 的搭建。

127.0.0.1 本机 ip

3、网卡



网卡是一个网络组件，属于硬件范畴，主要负责计算机之间数据的**封装和解封**。

MAC 地址: 网卡的物理地址，网卡设备的编号，**默认**情况是全球唯一的（16 进制）。

网络连接详细信息(D):

属性	值
连接特定的 DNS 后缀	
描述	Realtek PCIe GBE Family Controller #2
物理地址	DC-0E-A1-7B-37-EE

与 IP 地址的区别:

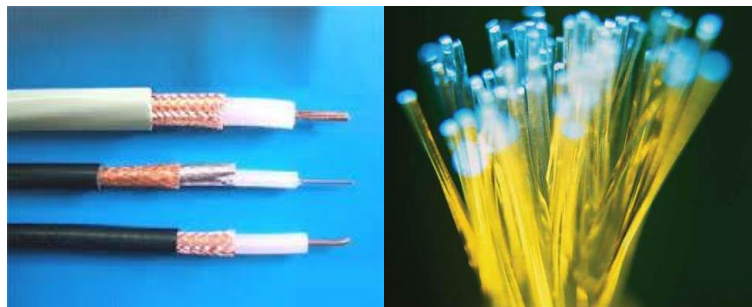
- 长度不同。IP 地址为 32 位，MAC 地址为 48 位。
- 分配依据不同。
- **网络寻址方式不同**。OSI 参考模型，ip 地址是基于第三层工作（网络层），mac 地址是第二层（数据链路层）

4、网线

网线是连接局域网必不可少的。在局域网中常见的网线主要有双绞线（RJ45 接口）、铜轴电缆、光缆三种。



双绞线



铜轴电缆

光纤

5、交换机

交换机（Switch）意为“开关”，是一种用于电（光）信号转发的网络设备，交换机它可以为接入交换机的任意两个网络节点提供**独享**的电信号通路。



目前，交换机品牌比较有名的是：华为、华三（h3c）、思科、锐捷。

6、路由器

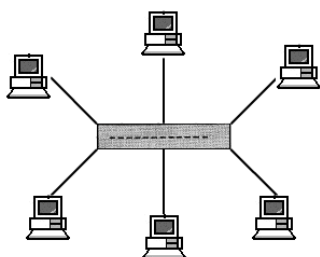
路由器（Router）又称网关设备（Gateway）是用于**连接多个逻辑上分开、相对独立的网络**。



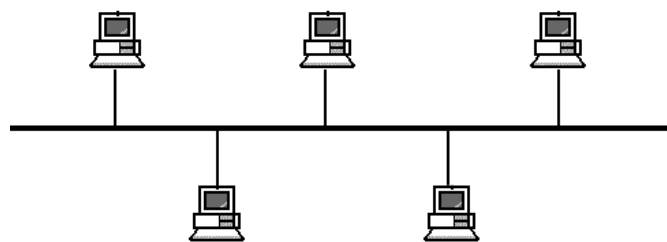
7、拓扑结构图（扩展）

所谓“拓扑”就是把实体抽象成与其大小、形状无关的“点”，而把连接实体的线路抽象成“线”，进而以图的形式来表示这些点与线之间关系的方法，**其目的在于研究这些点、线之间的相连关系**。表示点和线之间关系的图被称为拓扑结构图。

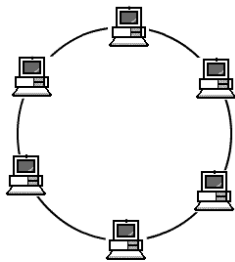
常见的几种拓扑结构图：



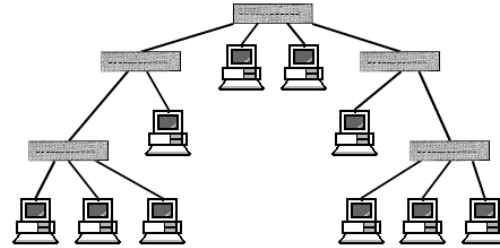
星型拓扑结构



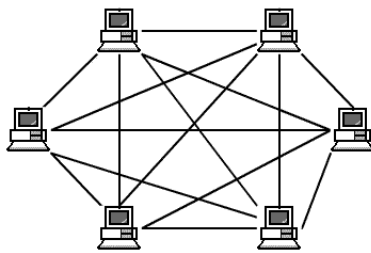
总线型拓扑结构



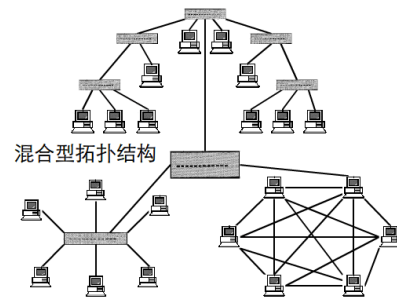
环型拓扑结构



树型拓扑结构



网状拓扑结构



混合型拓扑结构



二、网络相关命令

1、ping

作用：检测当前主机与目标主机之间的连通性（不是 100%准确，有的服务器是禁 ping）

语法：#ping 主机地址（ip 地址、主机名、域名等）

例如：测试和 baidu.com 之间的连通性。

```
[test@yunwei /]$ ping www.baidu.com
PING www.a.shifen.com (119.75.216.20) 56(84) bytes of data.
64 bytes from 127.0.0.1 (119.75.216.20): icmp_seq=1 ttl=128 time=4.97 ms
64 bytes from 127.0.0.1 (119.75.216.20): icmp_seq=2 ttl=128 time=4.33 ms
64 bytes from 127.0.0.1 (119.75.216.20): icmp_seq=3 ttl=128 time=4.04 ms
64 bytes from 127.0.0.1 (119.75.216.20): icmp_seq=4 ttl=128 time=5.79 ms
64 bytes from 127.0.0.1 (119.75.216.20): icmp_seq=5 ttl=128 time=5.43 ms
64 bytes from 127.0.0.1 (119.75.216.20): icmp_seq=6 ttl=128 time=5.32 ms
```

该命令可以跨平台，windows 下也可以使用，语法一致。（区别在于 Linux 下默认一直发送，windows 下默认发送 4 个数据包）

```
C:\Users\admin>ping www.baidu.com

正在 Ping www.a.shifen.com [119.75.213.61] 具有 32 字节的数据:
来自 119.75.213.61 的回复: 字节=32 时间=6ms TTL=56
来自 119.75.213.61 的回复: 字节=32 时间=4ms TTL=56
来自 119.75.213.61 的回复: 字节=32 时间=4ms TTL=56
来自 119.75.213.61 的回复: 字节=32 时间=4ms TTL=56

119.75.213.61 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 4ms, 最长 = 6ms, 平均 = 4ms
```

2、netstat

作用：表示查看网络的连接信息

语法：#netstat -tnlp (-t: tcp 协议, -n: 将字母转化成数字, -l: 列出状态为监听, -p: 显示进程相关信息)
#netstat -an (-a: 表示全部, -n: 将字母转化为数字)

TCP/IP 协议需要使用这个命令。

3、tracert

作用：查找当前主机与目标主机之间所有的网关（路由器，会给沿途各个路由器发送 icmp 数据包，路由器可能会不给响应）。

该命令不是内置命令，需要安装，但是目前的已经安装好了（之前选了开发工具）。

语法：#tracert 主机地址

```
[root@iZm5e2obxmodwo19h2x51Z ~]# traceroute www.baidu.com
traceroute to www.baidu.com (180.149.132.151), 30 hops max, 60 byte packets
 1 * * *
 2 11.208.177.109 (11.208.177.109) 1.139 ms 11.208.176.157 (11.208.176.157) 0.9
ms
 3 11.208.177.162 (11.208.177.162) 183.060 ms 11.208.177.238 (11.208.177.238) 0
.009 ms
 4 116.251.114.130 (116.251.114.130) 0.623 ms 116.251.114.134 (116.251.114.134)
70) 0.815 ms
```

类似于查看快递的跟踪路由：



扩展：在 windows 下也有类似的命令：tracert 主机地址

```
C:\Users\admin>tracert 114.114.114.114

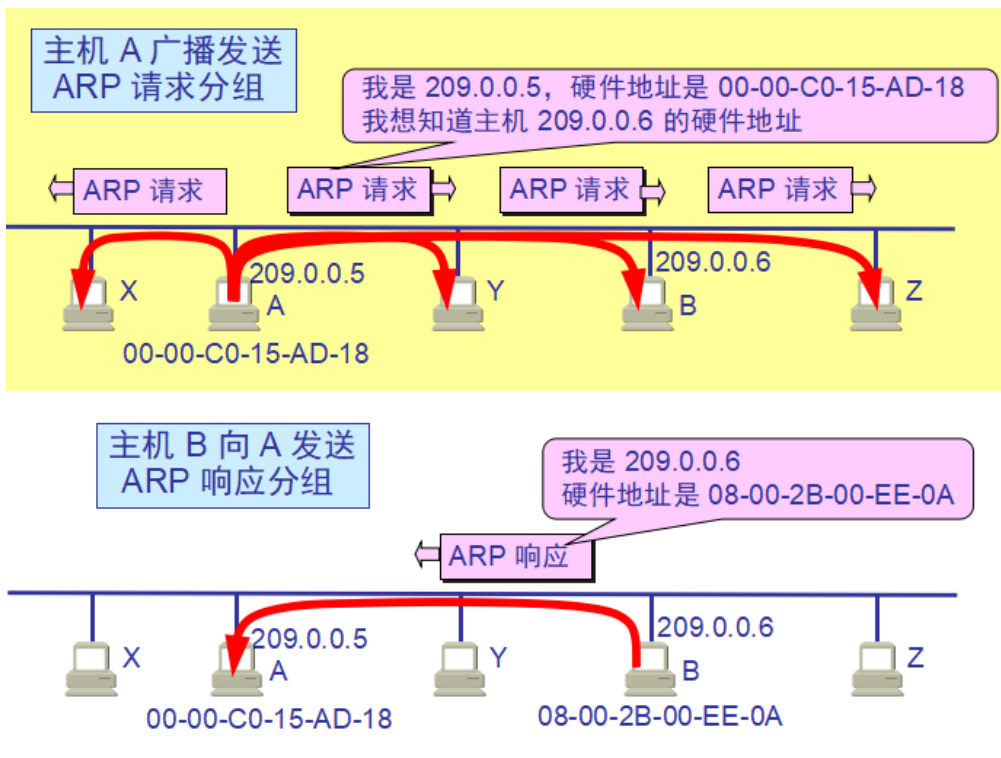
通过最多 30 个跃点跟踪
到 public1.114dns.com [114.114.114.114] 的路由:

  1    1 ms    <1 毫秒    1 ms    10.254.0.1
  2    *      *          *      请求超时。
  3    *      21 ms     10 ms    211.103.136.241
  4    2 ms    61 ms     53 ms    10.255.60.25
  5    4 ms    3 ms      5 ms     124.205.98.121
  6    4 ms    2 ms      2 ms     218.241.253.134
  7    8 ms    5 ms      3 ms     public1.114dns.com [114.114.114.114]
```

在线工具网址：<http://tool.chinaz.com>

4、arp

地址解析协议，即 ARP (Address Resolution Protocol)，是根据 IP 地址获取 (MAC) 物理地址的协议。



当一个主机发送数据时，首先查看本机 MAC 地址缓存中有没有目标主机的 MAC 地址，如果有就使用缓存中的结果；如果没有，ARP 协议就会发出一个广播包，该广播包要求查询目标主机 IP 地址对应的 MAC 地址，拥有该 IP 地址的主机会发出回应，回应中包括了目标主机的 MAC 地址，这样发送方就得到了目标主机的 MAC 地址。如果目标主机不在本地子网中，则 ARP 解析到的 MAC 地址是默认网关的 MAC 地址。

常用语法：`#arp -a` 查看本地缓存 mac 表
`#arp -d 主机地址` 删除指定的缓存记录

```
[root@yunwei /]# arp -a
? (192.168.21.1) at 00:50:56:c0:00:08 [ether] on eth0
? (192.168.21.2) at 00:50:56:f8:63:e5 [ether] on eth0
[root@yunwei /]#
```

该命令在 windows 下同样适用。

5、tcpdump(了解)

作用：抓包，抓取数据表

常用语法：

`#tcpdump 协议 port 端口`
`#tcpdump 协议 port 端口 host 地址`



#tcpdump -i 网卡设备名

查看 22 端口 (ssh) 的数据包:

```

192
00:09:17.430877 IP 192.168.21.1.rimf-ps > 192.168.21.136.ssh:
00:09:17.431715 IP 192.168.21.136.ssh > 192.168.21.1.rimf-ps:
288
00:09:17.431838 IP 192.168.21.1.rimf-ps > 192.168.21.136.ssh:
00:09:17.432675 IP 192.168.21.136.ssh > 192.168.21.1.rimf-ps:
288
00:09:17.433724 IP 192.168.21.136.ssh > 192.168.21.1.rimf-ps:
192
00:09:17.433783 IP 192.168.21.136.ssh > 192.168.21.1.rimf-ps:
192
00:09:17.433885 IP 192.168.21.1.rimf-ps > 192.168.21.136.ssh:
00:09:17.434939 IP 192.168.21.136.ssh > 192.168.21.1.rimf-ps:
288
00:09:17.435384 IP 192.168.21.1.rimf-ps > 192.168.21.136.ssh:
00:09:17.435393 IP 192.168.21.1.rimf-ps > 192.168.21.136.ssh:
4
00:09:17.435468 IP 192.168.21.136.ssh > 192.168.21.1.rimf-ps:

```

00:09:17.xxxx 监听数据的时分秒
IP: 使用的协议类型
192.168.21.1 数据包的一个方向 (来自)
> 数据的流向
192.168.21.136 数据包的另外一个方向 (到达)

三、项目上线流程 (必须掌握)

1、服务器选配购买

项目上线服务器必须是外网服务器。

一般服务器有 2 种情况：购买真实服务器、**购买云服务器**。

购买真实服务器一次性成本过高，所以现在基本都是选择云服务器。

云服务的厂商：**阿里云**、腾讯云、知道创宇 (加速乐)、华为云、盛大云、新浪云 (sae)、亚马逊云等等。

以后以阿里云为例：

官网：<http://www.aliyun.com>

①打开阿里云官网，选择产品中的“云服务器 ECS”



在页面上点击“立即购买”：



②选择具体的配置





当前选择实例: ecs.t5-lic1m1.small (1 vCPU 1 GB, 突发性能实例 t5) [突发性能实例](#) ① 不通用

购买实例数量: 台

您预设的配置参数为: 通用网络增强型 sn1ne, ecs.sn1ne.large, 2 vCPU, 4 GB
无法自动填充您预设的配置参数, 请手动选择

镜像 *

公共镜像 自定义镜像 共享镜像 镜像市场 ?

CentOS 安全加固 ?

存储

系统盘 高效云盘 40GB

云盘参数和性能

高效云盘 GB 1240 IOPS

如何选择 SSD云盘 / 高效云盘 / 普通云盘, 请看 [详细说明](#)>

数据盘 0/16

购买周期: 配置费用: **¥ 22.57**

下一步: 网络和安全组

网络 *

教我选择网络

专有网络 ?

默认专有网络 [前往控制台创建](#)

如需创建新的专有网络, 您可 [前往控制台创建](#)>

所选专有网络: 默认专有网络 所选交换机: 默认交换机
交换机所在可用区: 华北5 可用区A 交换机网段:

公网带宽

分配公网IP地址 系统会分配公网 IP, 也可采用更加灵活的弹性公网 IP 方案, 配置并 [绑定弹性公网 IP 地址](#)

公网带宽计费

? 后付费模式, 按使用流量 (单位为GB) 计费, 每小时扣费。请保证余额充足

1M 25M 50M 75M 100M Mbps

阿里云免费提供最高 5Gbps 的恶意流量攻击防护, [了解更多](#) | [提升防护能力](#) 如果流量不大, 建议使用按流量付费

安全组需要先在控制面板中创建, 创建好之后才能在这里进行选择 (安全组类似于防火墙, 可以设置相关规则):



进入后台查看信息：



需要重置密码的话，则可以选择右侧“更多”选择“重置密码”，然后重启服务器，最后可以通过远程终端连接服务器：

```
login as: root
root@39.104.60.10's password:

Welcome to Alibaba Cloud Elastic Compute Service !

[root@iZhp38oxmb9oefvqu4ep4tZ ~]#
```

2、域名购买

①在首页产品中找到域名注册



域名注册得先查看是否可以注册：



选择需要的域名：



确认购买信息：



✔ 恭喜，支付成功！
您可以[点击此处索取发票](#)或进入[域名控制台](#)进行产品管理
[用户指南:如何将域名指向网站](#)

购买之后就可以在后台控制面板中去查看域名情况。

3、域名备案

备案：当申请域名的人要想在国内使用域名，则需要向当地的通信管理局（省级）去申请报备。

备案前提：想要使用境内服务器的话，则必须得备案。

在管理后台点击“ICP 备案系统”



点击新增主体备案：

新增主体备案

增加网站

继续接入备案

填写完基本信息之后点击增加网站：

* 域名： ✓

* 产品类型： ✓
[如何选择产品类型？](#)

* 备案服务号： ✓
[获得备案服务号？](#)

备案服务号可以在控制台顶部去获取：



申请到备案服务号之后填写继续：

* 域名： ✓

* 产品类型： ✓
[如何选择产品类型?](#)

* 备案服务号： ✓
[获得备案服务号?](#)

会让用户下载一个图片：网站真实性核验单
下载打印，填写好上传到阿里云备案系统中。

后面等待初审，初审通过之后继续下一步（初审时间一般 1 天即可）

拍照



上传照片

等待管局审核（到这个步骤基本是已经通过，审核周期一般是 15 个工作日）。

等待审核通过，就会收到工信部发送的短信与邮件通知，邮件中有备案号和备案密码（备案密码用于注销备案）。

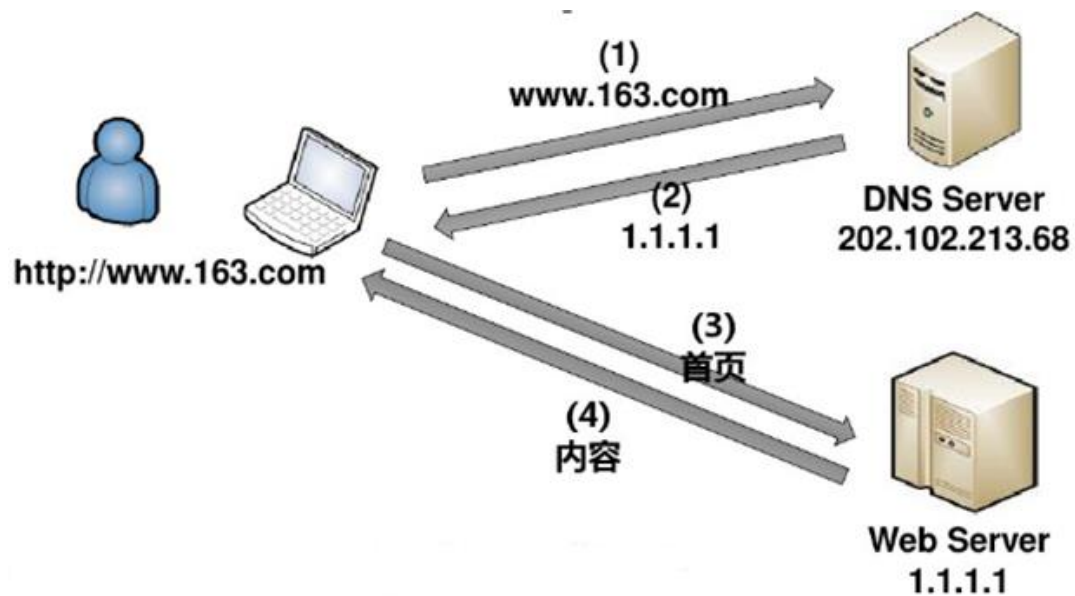
4、域名解析

点击“解析”

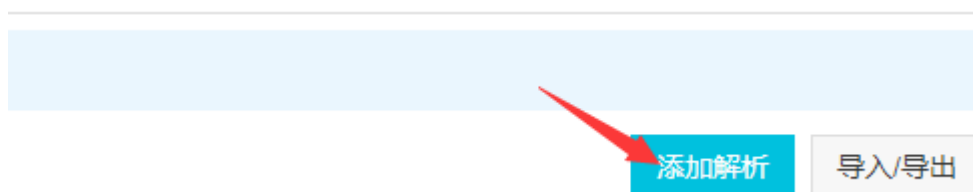


解析：将域名绑定到一个服务器地址的操作

DNS：domain name server，用于将域名转化成 ip 地址的服务器。



点击右上角的添加记录：



选择记录：



记录类型: A - 将域名指向一个IPV4地址

主机记录: A - 将域名指向一个IPV4地址
CNAME - 将域名指向另外一个域名

解析线路: AAAA - 将域名指向一个IPV6地址
NS - 将子域名指定其他DNS服务器解析

记录值: MX - 将域名指向邮件服务器地址

TTL值: SRV - 记录提供特定的服务的服务器
TXT - 文本长度限制512, 通常做SPF记录 (反垃圾邮件)
CAA - CA证书颁发机构授权校验
显性URL - 将域名302重定向到另外一个地址

例如：需要将 www.linux123.xyz 解析到之前购买的云主机上，则解析可以设置如下：

记录类型: A - 将域名指向一个IPV4地址

主机记录: www .linux123.xyz ?

解析线路: 默认 - 必填! 未匹配到智能解析线路时, 返回【默认】线路... ?

记录值: 39.104.60.10

TTL值: 10 分钟

同步默认线路

确认 取消

解析之后可以通过在线 ping 命令检测效果：

广东深圳[电信]	39.104.60.10	香港 特别行政区
四川成都[电信]	39.104.60.10	香港 特别行政区
浙江湖州[电信]	39.104.60.10	香港 特别行政区
浙江宁波[电信]	39.104.60.10	香港 特别行政区
江苏徐州[电信]	39.104.60.10	香港 特别行政区
浙江温州[电信]	39.104.60.10	香港 特别行政区
广东惠州[电信]	39.104.60.10	香港 特别行政区

5、配置生产环境（最后 1 天）

6、上传代码

此时需要使用上传工具：pscp，filezilla。

和之前使用的方式一样。

Shell 基础

一、关于 shell

1、什么是 shell

什么是 shell?

Shell（外壳）是一个用 C 语言编写的程序，它是用户使用 Linux 的桥梁。**Shell 既是一种命令语言，又是一种程序设计语言。**

Shell 是指一种应用程序，**这个应用程序提供了一个界面，用户通过这个界面访问操作系统内核的服务。**

什么是脚本?

脚本简单地说就是一条条的文字命令，这些文字命令是可以看到的（如可以用记事本打开查看、编辑）。

常见的脚本：**JavaScript（JS，前端）**，VBScript，**ASP，JSP，PHP（后端）**，SQL（数据库操



作语言), Perl, **Shell**, python, Ruby, JavaFX, Lua 等。

为什么要学习和使用 shell?

Shell 属于内置的脚本

程序开发的效率非常高, 依赖于功能强大的命令可以迅速地完成任务(批处理)

语法简单, 代码写起来比较轻松, 简单易学

常见的 shell 种类?

在 linux 中有很多类型的 shell, 不同的 shell 具备不同的功能, shell 还决定了脚本中函数的语法, Linux 中默认的 shell 是 **/bin/bash (重点)**, 流行的 shell 有 ash、bash、ksh、csh、zsh 等, 不同的 shell 都有自己的特点以及用途。

csh

C shell 使用的是“类 C”语法, csh 是具有 C 语言风格的一种 shell, 其内部命令有 52 个, 较为庞大。目前使用的并不多, 已经被 /bin/tcsh 所取代。

ksh

Korn shell 的语法与 Bourne shell 相同, 同时具备了 C shell 的易用特点。许多安装脚本都使用 ksh, ksh 有 42 条内部命令, 与 bash 相比有一定的限制性。

tcsh

tcsh 是 csh 的增强版, 与 C shell 完全兼容。

sh

是一个快捷方式, 已经被 /bin/bash 所取代。

nologin

指用户不能登录

```
[root@yunwei ~]# usermod -s /sbin/nologin aop
[root@yunwei ~]# vim /etc/passwd
```

zsh

目前 Linux 里最庞大的一种 shell: zsh。它有 84 个内部命令, 使用起来也比较复杂。一般情况下, 不会使用该 shell。

bash

大多数 Linux 系统默认使用的 shell, bash shell 是 Bourne shell 的一个免费版本, 它是最早的 Unix shell, bash 还有一个特点, 可以通过 help 命令来查看帮助。包含的功能几乎可以涵盖 shell 所具有的功能, 所以一般的 shell 脚本都会指定它为执行路径。

2、shell 入门

编写规范:

代码规范:

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

[指定告知系统当前这个脚本要使用的 shell 解释器]

Shell 相关指令

文件命名规范:

文件名.sh

.sh 是 linux 下 bash shell 的默认后缀

使用流程:

①创建.sh 文件

touch/vim

②编写 shell 代码

③执行 shell 脚本

脚本必须得有执行权限

案例 1: 创建 test.sh, 实现第一个 shell 脚本程序, 输出 hello world.

输出命令: #echo 123

注意: 输出的内容如果包含字母和符号 (不包含变量), 则需要用引号包括起来。如果是纯数字可以包也可以不包。

```
1 #!/bin/bash
2 echo 'hello world!'
3 █
```

```
[root@yunwei ~]# chmod +x test.sh
[root@yunwei ~]# ll test.sh
-rwxr-xr-x. 1 root root 33 4月  2 12:39 test.sh
[root@yunwei ~]# test.sh
-bash: test.sh: command not found
[root@yunwei ~]# ./test.sh
hello world!
[root@yunwei ~]# █
```

注意, 这里在运行时一定要写成 ./test.sh, 而不是 test.sh, 运行其它二进制的程序也一样, 直接写 test.sh, Linux 系统会去 PATH(环境变量) 里寻找有没有叫 test.sh 的, 而只有 /bin, /sbin, /usr/bin, /usr/sbin 等在 PATH 里, 你的当前目录通常不在 PATH 里, 所以写成 test.sh 是会找不到命令的, 要用 ./test.sh 告诉系统说, 就在当前目录找。

案例 2: 使用 root 用户帐号创建并执行 test2.sh, 实现创建一个 shelltest 用户, 并在其家目录中新建文件 try.html。

```
1 #!/bin/bash
2 useradd shelltest
3 touch /home/shelltest/try.html █
```

脚本执行的另外一个方式: /bin/bash 脚本的路径 (了解)



```
[root@yunwei ~]# touch test2.sh
[root@yunwei ~]# vim test2.sh
[root@yunwei ~]# chmod +x test2.sh
[root@yunwei ~]# /bin/bash test2.sh
[root@yunwei ~]# ls /home/shelltest/
try.html
[root@yunwei ~]#
```

Shell 脚本分为简单的写法（简单命令的堆积）和复杂写法（程序的设计）

二、shell 进阶（重点）

1、变量（重点）

1.1、变量的含义

a. 什么是量

量就是数据.

b. 什么是变量

数据可以发生改变就是变量.

在一个脚本周期内,其值可以发生改变的量就是变量.

c. 什么叫做一个脚本周期

一个脚本周期我们可以简单的理解为当前的 shell 文件

变量是 shell 中不可或缺的一部分，也是**最基础、最重要**的组成部分。

1.2、变量的定义与使用（重点）

变量，先定义后使用。

定义形如：`class_name="yunwe "`

使用形如：`echo $class_name`

变量就是由 2 部分组成,一个是变量名（左边），另外一部分是变量的值（右边）

变量名和变量值是什么关系??

变量名和变量值是**使用和被使用**关系; 我们的变量名来使用变量值;

在使用变量的时候一定需要在变量名前面添加一个\$符号,该要求在其他语言中也存在的(例

如 php)。

变量名的规范

注意，变量名后面的等号左右不能有空格，这可能和你熟悉的所有编程语言都不一样。同时，变量名的命名须遵循如下规则：

命名只能使用英文字母，数字和下划线，首个字符不能以数字开头。

中间不能有空格，可以使用下划线“_”。

不能使用标点符号。

不能使用 bash 里的关键字（可用 help 命令查看保留关键字）。

问题：以下哪个 shell 变量名是合法的？

A. var B. ?var C. user*name D. echo

案例 1：使用变量改写入门脚本中的第 1 个 shell 脚本。

```
1 #!/bin/bash
2 str='hello world'
3 echo $str
```

```
[root@yunwei ~]# touch test3.sh
[root@yunwei ~]# vim test3.sh
[root@yunwei ~]# chmod +x test3.sh
[root@yunwei ~]# ./test3.sh
hello world
```

关于单双引号的问题：

双引号能够识别变量，双引号能够实现转义（类似于“*”）

单引号是不能识别变量，只会原样输出，单引号是不能转义的

案例 2：定义一个变量，输出当前时间，要求格式为“年-月-日 时:分:秒”。

```
1 #!/bin/bash
2 dt=`date +%F %T`
3 echo $dt
```

注意：反引号（esc 键下方的那个键），当在脚本中需要执行一些指令并且将执行的结果赋给变量的时候需要使用“反引号”。

```
[root@yunwei ~]# vim ./test4.sh
[root@yunwei ~]# ./test4.sh
2018-04-02 13:33:15
[root@yunwei ~]#
```



1.3、只读变量（了解）

语法：readonly 变量名

案例：定义变量 a 并且其值为 10，随后设置其为只读变量，再去尝试重新赋值

```
1 #!/bin/bash
2 a=10
3 readonly a
4 a=20
5 echo $a
```

```
[root@yunwei ~]# ./test5.sh
./test5.sh: line 4: a: readonly variable
10
```

1.4、接收用户输入（重点）

语法：read -p 提示信息 变量名

案例：编写一个脚本 test6.sh，要求执行之后提示用户输入文件的名称（路径），然后自动为用户创建该文件

```
1 #!/bin/bash
2 read -p '请输入需要创建的文件路径：' filepath
3 touch $filepath
4 echo '文件创建成功！'
5 ls -l $filepath
```

```
[root@yunwei ~]# ./test6.sh
请输入需要创建的文件路径： /root/20180404.txt
文件创建成功！
-rw-r--r--. 1 root root 0 4月  2 14:07 /root/20180404.txt
```

1.5、删除变量（了解）

语法：unset 变量名

案例：定义变量 b=20，再输出 b 的值，随后删除 b，最后再输出下 b

```
1 #!/bin/bash
2 b=20
3 echo $b
4 unset b
5 echo $b
```

```
[root@yunwei ~]# chmod +x ./test7.sh
[root@yunwei ~]# ./test7.sh
20
[root@yunwei ~]# vim test7.sh
```

2、条件判断语句

老婆给当程序员的老公打电话：**下班顺路买一斤包子带回来，如果看到卖西瓜的，买一个**。当晚，程序员老公手捧一个包子进了家门…老婆怒道：你怎么就买了一个包子？！老公答曰：因为看到了卖西瓜的。

把程序员老婆的话当作一段需求分析一下吧。买一斤包子是一个确定无疑的需求项，无论后面是什么情况什么条件，前面这一斤包子是肯定要买的。看到卖西瓜的是一个条件判断，后面“买一个”是一个模糊不清的需求项，买一个什么呢？需求里没说啊。客户把这个当作开发人员默认了解的内容了。可是作为一个成熟合格的程序员，该老婆的丈夫应该马上跟进确认需求“买一个什么？”，要不然程序可怎么写呢？所以笑话里该程序员是不合格的，起码是不积极不负责的。在没有明确需求的情况下，他只能按照自己的理解来完成工作了。那比较可能的结果就有如下几种：

- 1 看到卖西瓜的，买一个西瓜
 如果 看到卖西瓜的
 那么
 买一个西瓜
 否则
 买一斤包子
- 2 看到卖西瓜的，买一个包子
 如果 看到卖西瓜的
 那么
 买一个包子
- 3 看到卖西瓜的，买一个卖西瓜的
- 4 看到卖西瓜的，买一个老婆一直想买的东西
- 5 看到卖西瓜的，随便买一个东西

上述 1 和 2 下面的条件汉字描述称之为“伪代码”，也是属于条件表达式的语法。

语法 1（一个条件）：

if condition



```

then
    command1
    command2
    ...
fi

```

单行写法（一般在命令行中执行的时候）：`if [condition]; then command; fi`

语法 2（两个条件）：

```

if condition
then
    command1
    command2
    ...
else
    command
fi

```

语法 3（多个条件）：

```

if condition1
then
    command1
elif condition2
then
    command2
else
    commandN
fi

```

3、运算符

在 shell 中，运算符和其他编程语言一样，常见的有算数运算符、关系运算符、逻辑运算符、字符串运算符、文件测试运算符等

3.1、算数运算符

下表列出了常用的算术运算符，假定变量 a 为 10，变量 b 为 20：

运算符	说明	举例
+	加法	<code>`expr \$a + \$b`</code> 结果为 30。

-	减法	<code>`expr \$a - \$b`</code> 结果为 -10。
*	乘法	<code>`expr \$a * \$b`</code> 结果为 200。
/	除法	<code>`expr \$b / \$a`</code> 结果为 2。
%	取余	<code>`expr \$b % \$a`</code> 结果为 0。
=	赋值	<code>a=\$b</code> 将把变量 b 的值赋给 a。
==	相等。用于比较两个数字，相同则返回 true（真）。	<code>[\$a == \$b]</code> 返回 false（假）。
!=	不相等。用于比较两个数字，不相同则返回 true。	<code>[\$a != \$b]</code> 返回 true。

注意：条件表达式要放在方括号之间，并且要有空格，例如：`[$a == $b]` 是错误的，必须写成 `[$a == $b]`。

原生 bash 不支持简单的数学运算，但是可以通过其他命令来实现，例如 awk 和 expr，expr 最常用。

expr 是一款表达式计算工具，使用它能完成表达式的求值操作。

例如，两个数相加(注意使用的是反引号 ` 而不是单引号 '):

```
#!/bin/bash
val=`expr 2 + 2`
echo "两数之和为 : $val"
```

两点注意：

表达式和运算符之间要有空格，例如 `2+2` 是不对的，必须写成 `2 + 2`，这与我们熟悉的大多数编程语言不一样。

完整的表达式要被 ` 包含，注意这个字符不是常用的单引号，在 Esc 键下边。



```
9 echo ''
10 echo 'a + b = ' `expr $a + $b`
11 echo 'a - b = ' `expr $a - $b`
12 echo 'a * b = ' `expr $a \* $b`
13 echo 'b / a = ' `expr $b / $a`
14 echo 'b % a = ' `expr $b % $a`
15
16 #判断是否相等
17 if [ $a == $b ]
18 then
19     echo 'a等于b'
20 else
21     echo 'a不等于b'
22 fi
23
24 #判断是否不相等
25 if [ $a != $b ]
26 then
27     echo 'a不等于b'
28 else
29     echo 'a等于b'
30 fi
```

```
[root@yunwei ~]# ./test8.sh
a=10
b=20

a + b = 30
a - b = -10
a * b = 200
b / a = 2
b % a = 0
a不等于b
a不等于b
```

脚本的执行结果

3.2、关系运算符

关系运算符只支持数字，不支持字符串，除非字符串的值是数字。

下表列出了常用的关系运算符，假定变量 a 为 10，变量 b 为 20:

运算符	说明	举例
-eq	检测两个数是否相等，相等返回 true。	[\$a -eq \$b] 返回 false。
-ne	检测两个数是否相等，不相等返回 true。	[\$a -ne \$b] 返回 true。
-gt	检测左边的数是否大于右边的，如果是，则返回 true。	[\$a -gt \$b] 返回 false。
-lt	检测左边的数是否小于右边的，如果是，则返回 true。	[\$a -lt \$b] 返回 true。
-ge	检测左边的数是否大于等于右边的，如果是，则返回 true。	[\$a -ge \$b] 返回 false。
-le	检测左边的数是否小于等于右边的，如果是，则返回 true。	[\$a -le \$b] 返回 true。

-eq: equal

-ne: not equal

-gt: great than

-lt: less than

-ge: great than or equal

-le: less than or equal

案例：使用 a=10, b=20 来实现本案例

```

23
24 if [ $a -gt $b ]
25 then
26     echo 'a > b'
27 else
28     echo 'a < b'
29 fi
30
31 if [ $a -lt $b ]
32 then
33     echo 'a < b'
34 else
35     echo 'a > b'
36 fi

```



```
[root@yunwei ~]# ./test9.sh
a = 10
b = 20

a ≠ b
a ≠ b
a < b
a < b
```

课堂作业:

写一个脚本，判断当前输入的用户是否存在。如果存在则提示“用户存在”否则提示“用户不存在”。

```
1 #!/bin/bash
2 #接收用户的输入
3 read -p '请输入需要查询的用户名: ' username
4
5 #获取指定用户名在passwd文件中出现的次数
6 times=`cat /etc/passwd|grep $username|wc -l`
7
8 #判断出现的次数，如果次数=0则用户存在，反之则不存在
9 if [ $times == 0 ]
10 then
11     echo '用户不存在'
12 else
13     echo '用户已存在'
14 fi
```

3.3、逻辑运算符

下表列出了常用的布尔运算符，假定变量 a 为 10，变量 b 为 20:

运算符	说明	举例
!	非运算，表达式为 true 则返回 false，否则返回 true。	[!false] 返回 true。
-o	或（或者）运算，有一个表达式为 true 则返回 true。	[\$a -lt 20 -o \$b -gt 100] 返回 true。
-a	与（并且）运算，两个表达式都为 true 才返回 true。	[\$a -lt 20 -a \$b -gt 100] 返回 false。

或运算：一个为真即为真，全部为假才是假

与运算：一个为假即为假，全部为真才是真

```
9
10 #或运算，有一个为真即为真
11 if [ $a -lt 20 -o $b -gt 100 ]
12 then
13     echo '真'
14 fi
15
16 #与运算，全部为真才为真
17 if [ $a -lt 20 -a $b -gt 100 ]
18 then
19     echo '真'
20 else
21     echo '假'
22 fi
```

3.4、字符串运算符

下表列出了常用的字符串运算符，假定变量 a 为 "abc"，变量 b 为 "efg"：

运算符	说明	举例
=	检测两个字符串是否相等，相等返回 true。	[\$a = \$b] 返回 false。
!=	检测两个字符串是否相等，不相等返回 true。	[\$a != \$b] 返回 true。
-z	检测字符串长度是否为 0，为 0 返回 true。	[-z \$a] 返回 false。
-n	检测字符串长度是否为 0，不为 0 返回 true。	[-n \$a] 返回 true。
str	检测字符串是否为空，不为空返回 true。	[\$a] 返回 true。

案例：将上述的语法验证下



```

18
19 if [ -z $a ]
20 then
21     echo 'true'
22 else
23     echo 'false'
24 fi
25
26 if [ -n $a ]
27 then
28     echo 'true'
29 else
30     echo 'false'
31 fi
32
33 if [ $a ]
34 then
35     echo 'true'
36 else
37     echo 'false'
38 fi

```

3.5、文件测试运算符（重点）

文件测试运算符用于检测 Unix/Linux 文件的各种属性。

属性检测描述如下：

操作符	说明	举例
-b file	检测文件是否是块设备文件，如果是，则返回 true。	[-b \$file] 返回 false。
-c file	检测文件是否是字符设备文件，如果是，则返回 true。	[-c \$file] 返回 false。
-d file	检测文件是否是目录，如果是，则返回 true。	[-d \$file] 返回 false。
-f file	检测文件是否是普通文件（既不是目录，也不是设备文件），如果是，则返回 true。	[-f \$file] 返回 true。
-g file	检测文件是否设置了 SGID 位，如果是，则返回 true。	[-g \$file] 返回 false。

-k file	检测文件是否设置了粘着位(Sticky Bit), 如果是, 则返回 true。	[-k \$file] 返回 false。
-p file	检测文件是否有名管道, 如果是, 则返回 true。	[-p \$file] 返回 false。
-u file	检测文件是否设置了 SUID 位, 如果是, 则返回 true。	[-u \$file] 返回 false。
-r file	检测文件是否可读, 如果是, 则返回 true。	[-r \$file] 返回 true。
-w file	检测文件是否可写, 如果是, 则返回 true。	[-w \$file] 返回 true。
-x file	检测文件是否可执行, 如果是, 则返回 true。	[-x \$file] 返回 true。
-s file	检测文件是否为空 (文件大小是否大于 0), 不为空返回 true。	[-s \$file] 返回 true。
-e file	检测文件 (包括目录) 是否存在, 如果是, 则返回 true。	[-e \$file] 返回 true。

案例: 测试上述标绿色的效果

```
[root@yunwei ~]# ./test13.sh
不是目录
是普通文件
可读
可写
不可执行
存在
[root@yunwei ~]#
```

注意: 权限几个判断, 如果只有一个部分符合, 则认为是有权限的。

4、shell 脚本附带选项 (重点)

问题描述: 在 linux shell 中如何处理 tail -10 access.log 这样的命令行选项?

步骤:

调用 tail 指令



系统把后续选项传递给 tail

Tail 先去打开指定的文件
取出最后 10 行

问题：自己写的 shell 是否也可以像内置命令一样传递一些选项呢？

答：可以的，传递方式与上述的描述是一样的，关键是怎么接收。例如：

传递：

```
#!/test.sh a b c
```

接收：

在脚本中可以用“\$1”来表示 a，“\$2”来表示 b，以此类推。

接收可以用“\$”加上选项对应的序号即可。

测试：编写 test14.sh，传递 a, b, c，输出其值

```
1 #!/bin/bash
2 echo $1 $2 $3
~
```

其实\$1、\$2 是变量。

练习：创建自定义指令“user”，可以直接执行，要求该指令具备以下语法和功能：

- a. #user -add 用户名 【添加用户】
- b. #user -del 用户名 【删除用户及其家目录】

```
1 #!/bin/bash
2
3 if [ $1 = '-add' ]
4 then
5     useradd $2
6 else
7     userdel -r $2
8 fi
```

同时题目中要求是指令，所以可以再去添加个别名：

```
8 alias cls='clear'
9 alias gj='shutdown -h now'
10 alias user='/root/test15.sh'
11 # Source global definitions
```

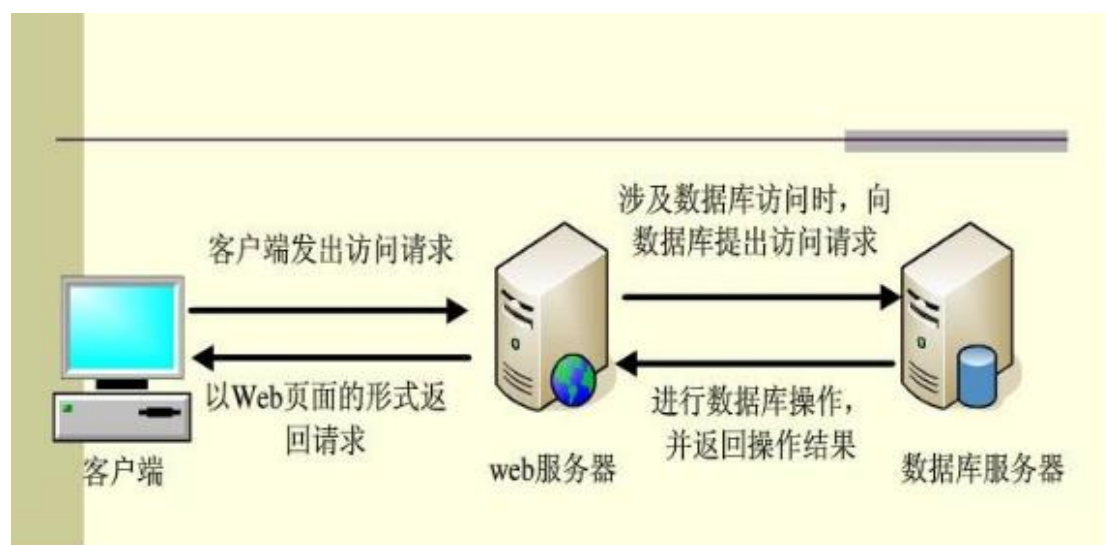
三、作业

- 1、尝试写一个 shell 的简易计算器功能，实现加减乘除。
- 2、作业：使用 -e 文件测试运算符，改写“1.4 接收用户输入”的案例，在创建文件的时候需要先判断是否存在，如果存在则提示用户并且不执行创建操作，如果不存在则创建。
- 3、尝试创建一个 shell 脚本，该脚本要求可以类似于“touch”指令一样，能够使用“touch 文件路径”的形式进行创建文件操作，并且要求创建好的文件权限默认为 755。

MySQL 基础

一、关于数据库

1、什么是数据库



如果一个项目是**动态**（内容会变化的，网页后缀.jsp、.php、.shtml等）内容的话，则数据库是必不可少的一个环节。

2、MySQL 简介

MySQL 是一个关系型**数据库管理系统**，由瑞典 MySQL AB 公司开发，2008 年被 Sun 公司收购，目前属于 Oracle 旗下产品。MySQL 是**最流行的数据库管理系统之一**，在 WEB 应用方面，MySQL 是**最好的** RDBMS (Relational Database Management System, 关系数据库管理系统) 应用软件。其和 php 是黄金搭档 (**LAMP/LNMP**)。



3、常见的其他数据库软件

目前市场上还有：Oracle（重量级的数据库）、MS SQL Server（微软）、Access（微软）、PostgreSQL、DB2、Mariadb（MySQL 分支，完全兼容 MySQL）。

二、MySQL 的安装与初始化

操作之前先确保计算机时间准确。

1、Linux 下的软件安装方式（初步）

a. 源码包（建议）

优点

开源，如果有足够的能力，可以修改源代码编译安装，更加适合自己的系统，稳定高效

缺点

安装步骤较多，**容易出错**
编译过程时间较长

案例：使用源码编译安装方式安装 ncurses（一种常用的终端库）

扩展：解包

常用语法：

```
#tar -zxvf *.tar.gz      (大多数)
#tar -jxvf *.tar.bz2
```

选项含义：

- z 或 --gzip 或 --ungzip：通过 gzip 指令处理文件；
- x 或 --extract 或 --get：从文件中还原文件；
- v：显示操作过程；
- f 或 --file：指定一个文件；
- j：支持 bzip2 解压文件；

①先将软件包传递到服务器上“/usr/local/src”

```
[root@yunwei ~]# cd /usr/local/src/
[root@yunwei src]# ls
mobile.sql  ncurses-6.1.tar.gz  openssl-1.1.1-pre4.tar.gz
[root@yunwei src]#
```

②解压需要安装的源码包

```
# tar -zxvf ncurses-6.1.tar.gz
```

```
[root@yunwei src]# ll
总用量 26532
-rw-r--r--. 1 root root 15534987 4月 6 09:47 mobile.sql
drwxr-xr-x. 15 1001 users 4096 1月 28 04:27 ncurses-6.1
-rw-r--r--. 1 root root 3365395 4月 6 09:47 ncurses-6.1.tar.gz
-rw-r--r--. 1 root root 8259067 4月 6 09:47 openssl-1.1.1-pre4.tar.gz
```

③切换到源码文件夹，然后执行后续操作

配置（config/configure/bootstrap） → 编译（make/bootstrapd） → 安装（make install/bootstrap install）

配置操作主要是指指定软件的安装目录、需要的依赖在什么地方、指定不需要可选依赖、配置文件的路径、通用数据存储位置等等。

指定安装的路径：--prefix=路径

需要依赖的路径：--with-PACKAGE 名=[包所在的路径]

不需要依赖：--without-PACHAGE 名

```
# ./configure --prefix=/usr/local/ncurses
```

```
** Configuration summary for NCURSES 6.1 20180127:

extended funcs: yes
xterm terminfo: xterm-new

bin directory: /usr/local/ncurses/bin
lib directory: /usr/local/ncurses/lib
include directory: /usr/local/ncurses/include/ncurses
man directory: /usr/local/ncurses/share/man
terminfo directory: /usr/local/ncurses/share/terminfo
```

```
# make
```

```
_GNU_SOURCE -D_FILE_OFFSET_BITS=64 -DNDEBUG -O2
make[1]: Leaving directory `/usr/local/src/ncurses-6.1/c++'
[root@yunwei ncurses-6.1]#
```

```
# make install
```

```
installing ./cursesw.h in /usr/local/ncurses/include/ncurses
installing ./cursslk.h in /usr/local/ncurses/include/ncurses
installing etip.h in /usr/local/ncurses/include/ncurses
make[1]: Leaving directory `/usr/local/src/ncurses-6.1/c++'
[root@yunwei ncurses-6.1]#
```

成功之后查看目录（可选）：

```
[root@yunwei ncurses-6.1]# ls /usr/local/ncurses/
bin include lib share
[root@yunwei ncurses-6.1]# /usr/local/ncurses/bin/
captainfo infotocap tabs tput
clear ncurses6-config tic tset
infocmp reset toe
```



b. 二进制包 (rpm)

优点: 包管理系统简单, 只需要几个命令就可以实现包的安装, 升级, 查询和卸载

缺点: 经过编译, 不再可以看到源代码

回顾 rpm 相关指令:

#rpm -qa|grep 关键词

#rpm -e 关键词 [--nodeps]

#rpm -ivh 完整名称

#rpm -Uvh 完整名称

#rpm -qf 文件路径 **【查询指定文件属于哪个包】**

```
[root@yunwei src]# rpm -qf /etc/httpd/conf/httpd.conf
httpd-2.2.15-29.el6.centos.i686
```

案例: 使用二进制包安装 lynx (一款纯命令行的浏览器)

在光盘中就有这个包

```
[root@yunwei src]# mkdir /mnt/dvd
[root@yunwei src]# mount /dev/sr0 /mnt/dvd/
mount: block device /dev/sr0 is write-protected, mounting read-only
[root@yunwei src]# rpm -ivh /mnt/dvd/Packages/lynx-2.8.6-27.el6.i686.rpm
Preparing... ##### [100%]
 1:lynx ##### [100%]
[root@yunwei src]#
```

例如查看百度: #lynx --dump www.baidu.com

c. yum 等傻瓜式安装

优点: 安装简单, 快捷

缺点: 完全丧失了自定义性

注意: 如果不更改软件来源的情况下, 是需要联网才能使用 yum 的。

```
常用的 yum 指令:
#yum list [installed]      列出当前已经装的和可以装的软件 (全部)
#yum search 名            搜索指定的关键词的包
#yum [-y] install 包名    安装指定的包 (-y 表示允许不再确认)
#yum [-y] update [包名]   更新指定的包, 不指定包则更新全部软件
```

```
#yum [-y] remove 包名      卸载指定的包
```

案例：使用 yum 指令卸载火狐浏览器

```
#yum remove firefox
```

```
Verifying   : firefox-17.0.10-1.el6.centos.i686

Removed:
  firefox.i686 0:17.0.10-1.el6.centos

Complete!
```

案例：使用 yum 指令安装火狐浏览器

```
#yum install firefox
```

2、安装 MySQL（重点）

注：此处安装以 yum 安装为例

2.1、MySQL 安装

```
#yum install mysql-server
```

```
perl-DBD-MySQL      i686      4.013-3.el6
perl-DBI            i686      1.609-4.el6
Updating for dependencies:
mysql-libs         i686      5.1.73-8.el6_8

Transaction Summary
-----
Install      4 Package(s)
Upgrade     1 Package(s)

Total download size: 12 M
Is this ok [y/N]: y
```

完成的：



```

Installed:
  mysql-server.i686 0:5.1.73-8.el6_8

Dependency Installed:
  mysql.i686 0:5.1.73-8.el6_8          perl-DBD-MySQL.i686 0:4.011
  perl-DBI.i686 0:1.609-4.el6

Dependency Updated:
  mysql-libs.i686 0:5.1.73-8.el6_8

Complete!
[root@yunwei src]#

```

2.2、MySQL 初始化

#service mysqld start

```

Please report any problems with the /usr/bin/mysqlbug script!

正在启动 mysqld:

```

查看端口号 (默认端口号 3306):

```

[root@yunwei src]# netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:3306              0.0.0.0:*                LISTEN      20877/mysqld

```

mysql_secure_installation

Enter current password for root (enter for none):请输入当前 root 用户的密码, 如果没有按回车, 注意此 root 并非 linux 的 root 用户。

Set root password?是否设置 root 密码?

需要设置的密码: qhabOfhlluB9

Remove anonymous users?是否移除匿名用户, 选择移除 (Y)

Disallow root login remotely?是否不允许 root 远程登录 (默认不允许)

Remove test database and access to it?是否移除测试数据库 (建议先不移除)

Reload privilege tables now?是否重新加载权限表 (当我们更改了 mysql 用户相关的信息之后建议去重载权限)

```

All done!  If you've completed all of the above steps, your MySQL
installation should now be secure.

Thanks for using MySQL!

```

2.3、MySQL 的启动控制

语法: `service mysqld start/stop/restart`

进入 mysql 的方式:

```
#mysql -u 用户名 -p
```

退出 MySQL 到 linux 命令行:

```
mysql > exit
```

2.4、默认目录/文件位置 (了解)

数据库存储目录: `/var/lib/mysql`

配置文件: `/etc/my.cnf`

```
[root@yunwei src]# ls /var/lib/mysql
ibdata1  ib_logfile0  ib_logfile1  mysql  mysql.sock  tes
[root@yunwei src]# ls /etc/my.cnf
/etc/my.cnf
[root@yunwei src]#
```

三、MySQL 的基本操作 (难点)

1、名词介绍

以 Excel 文件举例:

数据库: 可以看作是整个 excel 文件。

数据表: 可以看作是一个 excel 文件中的工作表。

行 (**记录**): 可以看作是一个工作表中的一行

列 (**字段**): 可以看作是一个工作表总的一列

2、库操作 (重点)

以下命令在 MySQL 终端命令行中执行 (大小写均可):

```
SHOW DATABASES;          显示当前 MySQL 中全部的数据库
```

```
CREATE DATABASE 库名;    创建数据库
```



DROP DATABASE 库名;	删除数据库
USE 库名;	切换数据库

Show databases 效果

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| test |
+-----+
3 rows in set (0.00 sec)
```

创建数据库：创建 yunwei 数据库

```
mysql> create database yunwei;
Query OK, 1 row affected (0.00 sec)
```

删除数据库：删除 yunwei 数据库

```
mysql> drop database yunwei;
Query OK, 0 rows affected (0.00 sec)
```

切换数据库：切换到 test 数据库

```
mysql> use test;
Database changed
```

3、表操作

SHOW TABLES;	显示当前数据库中所有的表名（必须先 use 数据库）
CREATE TABLE 表名称	在当前数据库下创建数据表
(

列名称 1 数据类型 [NOT NULL AUTO_INCREMENT],

列名称 2 数据类型,

列名称 3 数据类型,

.....,

PRIMARY KEY(主键字段名)

);

常见的数据类型: int (整型)、char (定长字符)、varchar (不定长字符)。

主键一般就是序号所在的那一列 (主键不能重复)。

DESC 表名;

描述一个数据表 (查看表结构)

DROP TABLE [IF EXISTS] 表名;

删除一个数据表

案例: 使用上述的语法

查看所有的数据表

```
mysql> show tables;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| event          |
| func           |
| general_log     |
```

创建数据表 (去 test 库中创建)

要求: 表名 xg, 要求有字段如下:

Id 字段, 11 位整型, 不为空, 自增, 主键

Username 字段, varchar 类型, 20 长度

Password 字段, char 类型, 32 长度

SQL (standard query language) 语句:

Create table xg(

Id int(11) not null auto_increment,

Username varchar(20),

Password char(32),

Primary key(id)

);



```
mysql> use test;
Database changed
mysql> create table xg(
  -> id int(11) not null auto_increment,
  -> username varchar(20),
  -> password char(32),
  -> primary key(id)
  -> );
Query OK, 0 rows affected (0.38 sec)
```

查看表结构:

```
mysql> desc xg;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)       | NO   | PRI | NULL    | auto_increment |
| username   | varchar(20)   | YES  |     | NULL    |                |
| password   | char(32)      | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

删除数据表:

```
mysql> drop table if exists xg;
Query OK, 0 rows affected (0.00 sec)
```

4、记录/字段操作 (重点)

4.1、增加记录

语法 1: INSERT INTO 表名称 VALUES (值 1, 值 2,....);

语法 2: INSERT INTO 表名称 (列 1, 列 2,...) VALUES (值 1, 值 2,....);

案例: 往数据表 xg 表中新增一个记录 username 为 zhangsan, password 为 123456 (加密结果 E10ADC3949BA59ABBE56E057F20F883E)

Sql 语句:

```
insert into xg (username,password) values ('zhangsan','E10ADC3949BA59ABBE56E057F20F883E')
```

```
mysql> insert into xg (username,password) values ('zhangsan','E10ADC3949BA59ABBE56E057F20F883E');
Query OK, 1 row affected (0.00 sec)
```

要求前面的列名与值要一一对应。

4.2、更新记录

语法: **UPDATE** 表名称 SET 列名称 1 = 新值 1,列名称 2 = 新值 2... **WHERE** 列名称 = 某值;

案例: 使用更新语句更新 **id** 大于等于 **2** 的记录, 将其密码改为: 25F9E794323B453885F5181F1B624D0B

SQL 语句:

```
Update xg set password = '25F9E794323B453885F5181F1B624D0B' where id >= 2;
```

```
mysql> update xg set password = '25F9E794323B453885F5181F1B624D0B' where id >= 2;
Query OK, 2 rows affected (0.00 sec)
Rows matched: 2 Changed: 2 Warnings: 0
```

以后在执行影响行数的 **sql** 操作的时候一定需要注意条件是否写错或者漏写。

4.3、查询记录

SELECT 列名称 1,列名称 2... FROM 表名称 WHERE 条件;

SELECT * FROM 表名称 WHERE 条件;

案例: 查询刚才新增的记录

只查询用户名和密码, 并且是 **id=2** 的:

```
Select username,password from xg where id = 2;
```

```
mysql> select username,password from xg where id = 2;
+-----+-----+
| username | password |
+-----+-----+
| lisi     | E10ADC3949BA59ABBE56E057F20F883E |
+-----+-----+
1 row in set (0.00 sec)
```

查询全部:

```
Select * from xg;
```



```
mysql> select * from xg;
+-----+-----+-----+
| id | username | password |
+-----+-----+-----+
| 1 | zhangsan | E10ADC3949BA59ABBE56E057F20F883E |
| 2 | lisi     | E10ADC3949BA59ABBE56E057F20F883E |
| 3 | wangwu  | E10ADC3949BA59ABBE56E057F20F883E |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

4.4、删除记录

DELETE FROM 表名称 **WHERE** 列名称 = 值;

案例：删除 id 为 2 的记录
Delete from xg where id = 2;

```
mysql> delete from xg where id = 2;
Query OK, 1 row affected (0.00 sec)
```

5、备份与还原（重点）

5.1、备份（导出）

全量备份（数据+结构）：#mysqldump -uroot -p123456 -A > 备份文件路径

指定库备份（数据+结构）：#mysqldump -uroot -p123456 库名 > 备份文件路径

多个库备份（数据+结构）：#mysqldump -uroot -p123456 --databases db1 db2 > 备份文件路径

案例：备份整个库

```
# mysqldump -uroot -pqhabOfhlluB9 -A > /root/sql_201804061609.sql
```

```
[root@yunwei ~]# ls
0406.txt          hello.c          passwd          te
20180404.txt     ifcfg-eth0      password        te
anaconda-ks.cfg  install.log     sql_201804061609.sql  te
group            install.log.syslog  sss.txt        te
hebing.txt       ooa             test10.sh       te
[root@yunwei ~]#
```

案例：每 1 分钟自动备份 1 次 test 数据库

```
1 #!/bin/bash
2 filename="test_`date +%Y%m%d%H%M%S`".sql"
3 mysqldump -uroot -pqhabOfhlluB9 test > /root/$filename
```

计划任务编写：

```
* * * * * /root/test19.sh
```

等待几分钟观察目录情况：

```
[root@yunwei ~]# ls test*
test10.sh  test15.sh  test_20180406161937.sql  test3.sh
test11.sh  test16.sh  test_20180406162101.sql  test4.sh
test12.sh  test17.sh  test_20180406162201.sql  test5.sh
test13.sh  test18.sh  test_20180406162301.sql  test6.sh
test14.sh  test19.sh  test2.sh                 test7.sh
```

5.2、还原（导入）

还原部分分（1）mysql 命令行 source 方法和（2）系统命令行方法

1. 还原全部数据库：

(1) mysql 命令行：mysql> source 备份文件路径

(2) 系统命令行：#mysql -uroot -p123456 < 备份文件路径

2. 还原单个数据库(需指定数据库)

(1) mysql> use 库名

mysql> source 备份文件路径

(2) #mysql -uroot -p123456 库名 < 备份文件路径

3. 还原单个数据库的多个表(需指定数据库)



(1) mysql> use 库名

mysql> source 备份文件路径

(2) mysql -uroot -p123456 库名 < 备份文件路径

4. 还原多个数据库，（一个备份文件里有多个数据库的备份，此时不需要指定数据库）

(1) mysql 命令行: mysql> source 备份文件路径

(2) 系统命令行: mysql -uroot -p123456 < 备份文件路径

案例 1: 人为删除 xg 表（模拟数据表丢失），然后通过最后一次备份还原数据表。
先删除数据表

```
mysql> show tables;
+-----+
| Tables_in_test |
+-----+
| xg              |
+-----+
1 row in set (0.00 sec)

mysql> drop table xg;
Query OK, 0 rows affected (0.00 sec)
```

还原操作:

```
mysql> source /root/test_20180406162601.sql
Query OK, 0 rows affected (0.00 sec)
```

案例 2: 需要还原 sql 文件到 test 库（mobile.sql 31 万条数据）

```
mysql> use test
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with 'no-tab-completion'

Database changed
mysql> source /usr/local/src/mobile.sql
```

设置 Mysql 连接字符集:

Mysql> set names utf8;

【三码一致，服务器端+传输过程中+客户端】

```
| 315970 | 1899992 | 新疆 | 乌鲁木齐 | 电信
| 315971 | 1899993 | 新疆 | 乌鲁木齐 | 电信
| 315972 | 1899994 | 新疆 | 乌鲁木齐 | 电信
| 315973 | 1899995 | 新疆 | 乌鲁木齐 | 电信
| 315974 | 1899996 | 新疆 | 乌鲁木齐 | 电信
| 315975 | 1899997 | 新疆 | 乌鲁木齐 | 电信
| 315976 | 1899998 | 新疆 | 乌鲁木齐 | 电信
| 315977 | 1899999 | 新疆 | 乌鲁木齐 | 电信
+-----+-----+-----+-----+-----+
315977 rows in set (0.29 sec)
```

四、扩展

1、mysql 的远程管理工具

分为两大类：B/S 架构、C/S 架构。

B/S: B 是指浏览器，S 是指服务器。例如：百度搜索应用就属于 BS 架构软件。

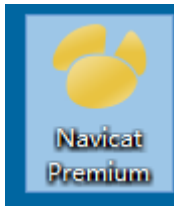
C/S: C 是指客户端，S 是指服务器。例如：QQ、电脑端微信等应用程序都是 CS 架构。

在 BS 中，mysql 有个典型的管理工具：PMA (phpMyAdmin)

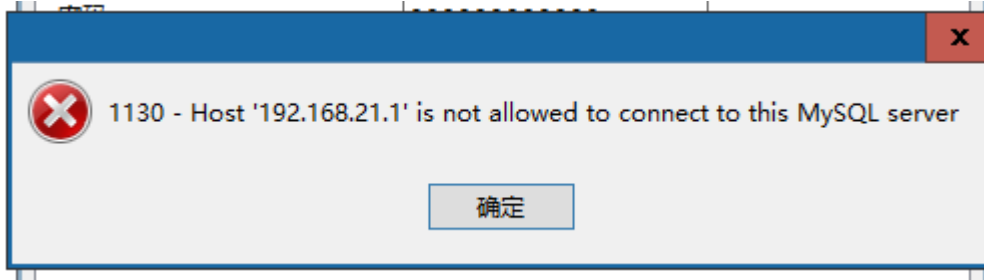




CS 中比较典型的软件：**navicat**、mysql workbrach



要解决的问题：允许 mysql 远程登录



- a. 先进入数据库选择 mysql 数据库；
- b. 执行 sql 语句：select host,user from user;

```
mysql> select host,user from user;
+-----+-----+
| host      | user  |
+-----+-----+
| 127.0.0.1 | root  |
| localhost | root  |
| yunwei    | root  |
+-----+-----+
3 rows in set (0.00 sec)
```

- c. 将其中的一个记录的 host 值改为 “%”，表示可以允许任何地方登录

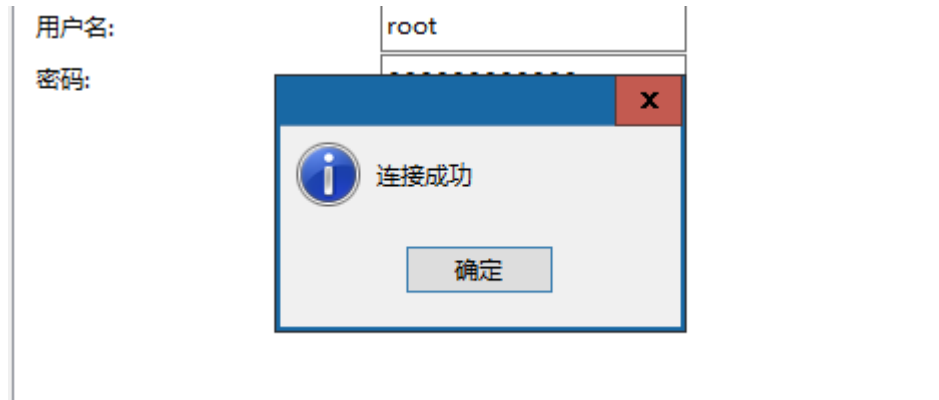
```
mysql> update user set host = '%' where host = 'yunwei';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select host,user from user;
+-----+-----+
| host      | user  |
+-----+-----+
| %         | root  |
| 127.0.0.1 | root  |
+-----+-----+
```

- d. 刷新权限表或者重启 mysql
刷新权限：**mysql> flush privileges;**

```
mysql> flush privileges;  
Query OK, 0 rows affected (0.00 sec)
```

e. navicat 登录成功



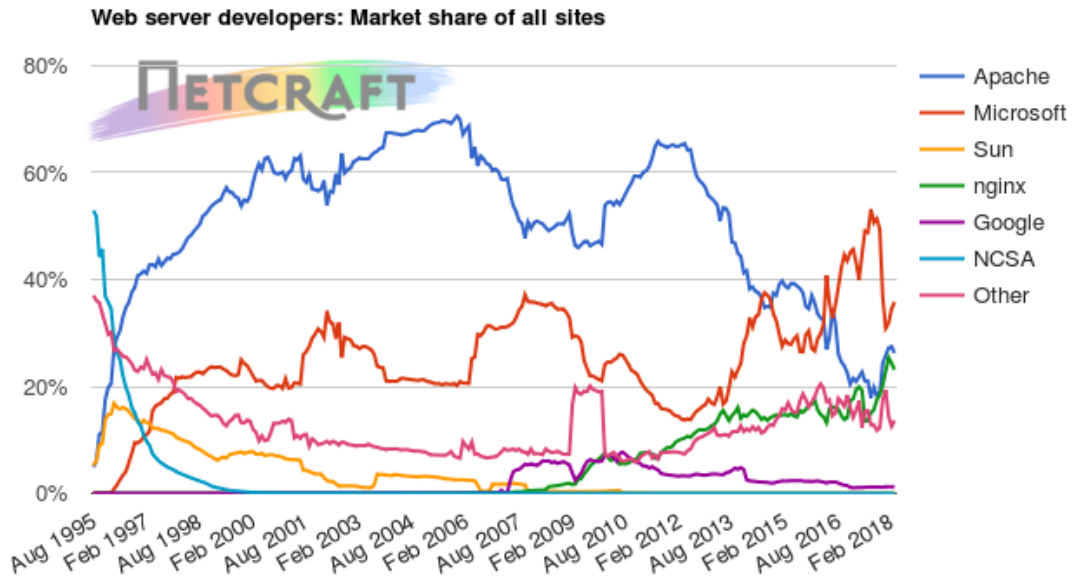
五、作业

- 1、导入 sql 文件（mobile.sql）到 test 数据库，并使用该数据表进行数据的增删改查练习。
- 2、请编写一个简单的 shell 脚本，文件名为 autoBackup.sh，并写出计划任务指令，能够实现每天 0 点整自动备份整个 MySQL 数据库。

Yum 项目上线实战（网站运维）

一、编译安装与卸载 Nginx

Nginx: 是一款比较流行的 web 服务器软件，类似于 Apache。



1、安装 nginx

①下载 nginx

下载地址：<https://nginx.org/en/download.html>

使用在服务器端下载的方式进行下载（此处不使用 filezilla）：

#wget 地址

例如当前需要下载 nginx 到 “/usr/local/src”

#wget <https://nginx.org/download/nginx-1.13.11.tar.gz>

```
[root@yunwei src]# wget https://nginx.org/download/nginx-1.13.11.tar.gz
--2018-04-06 18:27:36-- https://nginx.org/download/nginx-1.13.11.tar.gz
正在解析主机 nginx.org... 206.251.255.63, 95.211.80.227, 2001:1af8:4060:a004
[root@yunwei src]# ls
mobile.sql      ncurses-6.1.tar.gz      openssl-1.1.1-pre4.tar.gz
ncurses-6.1    nginx-1.13.11.tar.gz
```

②解压 nginx 安装包

tar -zxvf nginx-1.13.11.tar.gz

③进入 nginx 解压目录

开始进行配置、编译、安装操作

在配置时候报错：没有 PCRE 库

```
./configure: error: the HTTP rewrite module requires the PCRE library.
You can either disable the module by using --without-http_rewrite_module
option, or install the PCRE library into the system, or build the PCRE library
statically from the source with nginx by using --with-pcre=<path> option.
```

直接 yum 安装 pcre-devel:

```
#yum install pcre-devel
```

```
=====  
Installing:  
  pcre-devel                i686                7.8-7.el6  
Updating for dependencies:  
  pcre                      i686                7.8-7.el6  
  
Transaction Summary  
=====  
Install      1 Package(s)  
Upgrade     1 Package(s)  
  
Total download size: 517 k  
Is this ok [y/N]: █
```

报错缺少 zlib 库:

```
./configure: error: the HTTP gzip module requires the zlib library.  
You can either disable the module by using --without-http_gzip_module  
option, or install the zlib library into the system, or build the zlib library  
statically from the source with nginx by using --with-zlib=<path> option.
```

直接使用 yum 安装 zlib 库:

```
#yum install zlib-devel
```

```
=====  
Package                Arch                Version  
=====  
Installing:  
zlib-devel             i686                1.2.3-2  
  
Transaction Summary  
=====  
Install      1 Package(s)  
  
Total download size: 44 k  
Installed size: 115 k  
Is this ok [y/N]: █
```

还需要自己去下载一个 zlib 的源码包, 然后解压出来:

```
[root@yunwei src]# ls  
mobile.sql  ncurses-6.1.tar.gz  nginx-1.13.11.tar.gz  zlib-1.2.11  
ncurses-6.1  nginx-1.13.11      openssl-1.1.1-pre4.tar.gz  zlib-1.2.11.tar.gz  
[root@yunwei src]# █
```



```
Configuration summary
+ using system PCRE library
+ OpenSSL library is not used
+ using zlib library: /usr/lib/

nginx path prefix: "/usr/local/nginx"
nginx binary file: "/usr/local/nginx/sbin/nginx"
nginx modules path: "/usr/local/nginx/modules"
nginx configuration prefix: "/usr/local/nginx/conf"
nginx configuration file: "/usr/local/nginx/conf/nginx.conf"
nginx pid file: "/usr/local/nginx/logs/nginx.pid"
nginx error log file: "/usr/local/nginx/logs/error.log"
nginx http access log file: "/usr/local/nginx/logs/access.log"
nginx http client request body temporary files: "client_body_temp"
nginx http proxy temporary files: "proxy_temp"
nginx http fastcgi temporary files: "fastcgi_temp"
nginx http uwsgi temporary files: "uwsgi_temp"
nginx http scgi temporary files: "scgi_temp"
```

最终的 nginx 配置命令:

```
#!/configure --prefix=/usr/local/nginx --with-pcre --with-zlib=/usr/local/src/zlib-1.2.11
```

开始安装:

```
#make
```

```
-Wl,-E
sed -e "s|%%PREFIX%%|/usr/local/nginx|" \
    -e "s|%%PID_PATH%%|/usr/local/nginx/logs/nginx.pid|" \
    -e "s|%%CONF_PATH%%|/usr/local/nginx/conf/nginx.conf|" \
    -e "s|%%ERROR_LOG_PATH%%|/usr/local/nginx/logs/error.log|" \
    < man/nginx.8 > objs/nginx.8
make[1]: Leaving directory `/usr/local/src/nginx-1.13.11'
```

最后安装:

```
#make install
```

```
|| cp -R html '/usr/local/nginx'
test -d '/usr/local/nginx/logs' \
    || mkdir -p '/usr/local/nginx/logs'
make[1]: Leaving directory `/usr/local/src/nginx-1.13.11'
[root@yunwei nginx-1.13.11]#
```

安装好的目录:

```
[root@yunwei nginx-1.13.11]# ls /usr/local/nginx/
conf  html  logs  sbin
[root@yunwei nginx-1.13.11]#
```

④运行 nginx

先停止 Apache, 然后再运行 nginx

```
nginx: [emerg] still could not bind()
[root@yunwei nginx-1.13.11]# service httpd stop
停止 httpd: [确定]
[root@yunwei nginx-1.13.11]# /usr/local/nginx/sbin/nginx
[root@yunwei nginx-1.13.11]#
```

```
#/usr/local/nginx/sbin/nginx          【启动命令】  
#/usr/local/nginx/sbin/nginx -s reload 【重载，重载配置文件】
```

启动效果：

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

⑤了解：卸载编译安装的软件

`#rm -rf` 软件的安装目录

注意：卸载一个编译安装的软件的时候必须先停止。

二、关于 LAMP

LAMP: Linux + Apache + MySQL + PHP LAMP 架构（组合）

LNMP: Linux + Nginx + MySQL + php-fpm LNMP 架构（组合）

LNMPA: Linux + Nginx + MySQL + PHP + Apache Nginx 代理方式





Apache是世界使用排名第一的Web服务器软件。它可以运行在几乎所有广泛使用的计算机平台上，由于其跨平台和安全性被广泛使用，是最流行的Web服务器端软件之一。



PHP（外文名:PHP: Hypertext Preprocessor，中文名：“超文本预处理器”）是一种通用开源脚本语言。语法吸收了C语言、Java和Perl的特点，利于学习，使用广泛，主要适用于Web开发领域。PHP独特的语法混合了C、Java、Perl以及PHP自创的语法。它可以比CGI或者Perl更快速地执行动态网页。用PHP做出的动态页面与其他的编程语言相比，PHP是将程序嵌入到HTML（标准通用标记语言下的一个应用）文档中去执行，执行效率比完全生成HTML标记的CGI要高许多；PHP还可以执行编译后代码，编译可以达到加密和优化代码运行，使代码运行更快。

三、LAMP 环境部署

首先登录控制台获取需要连接的主机 ip 地址：

标签	监控	所在可用区	IP地址	状态	网络类型	配置
efvqu4ep4t		华北5可用区A	39.104.60.10(公有) 192.168.0.210(私有)	运行中	专有网络	1 vCPU 2 GB (虚拟化) ecs.t5-lc1m2.s 100Mbps (峰值)

后续可以进行远程登录。

```
login as: root
root@39.104.60.10's password:
Last login: Fri Apr 6 22:56:13 2018 from 222.129.228.228

Welcome to Alibaba Cloud Elastic Compute Service !

[root@iZhp38oxmb9oefvqu4ep4tZ ~]#
```

在整个 LAMP 中需要自己安装的也就只有 Apache + PHP + Mysql。后续以 yum 为例。

1、PHP 与 Apache 的安装

`#yum install php` 【在安装好 php 的同时会一起顺带安装 Apache】

```
Installing:
php          x86_64      5.3.3-49.el6          base          1.1 M
Installing for dependencies:
apr          x86_64      1.3.9-5.el6_9.1      updates      124 k
apr-util    x86_64      1.3.9-3.el6_0.1      base         87 k
apr-util-ldap x86_64      1.3.9-3.el6_0.1      base         15 k
httpd       x86_64      2.2.15-60.el6.centos.6 updates      836 k
httpd-tools x86_64      2.2.15-60.el6.centos.6 updates       80 k
```

启动 Apache: `#service httpd start`

```
[root@iZhp38oxmb9oefvqu4ep4tZ ~]# service httpd start
Starting httpd: httpd: apr_sockaddr_info_get() failed for iZhp38oxmb9oefvqu4ep4tZ
httpd: Could not reliably determine the server's fully qualified domain name using
ServerName
[ OK ]
```

此处会有一个警告，无法确定主机的 FQDN，如果需要处理，则需要修改 Apache 的配置文件 (`/etc/httpd/conf/httpd.conf`)

`# vim /etc/httpd/conf/httpd.conf`

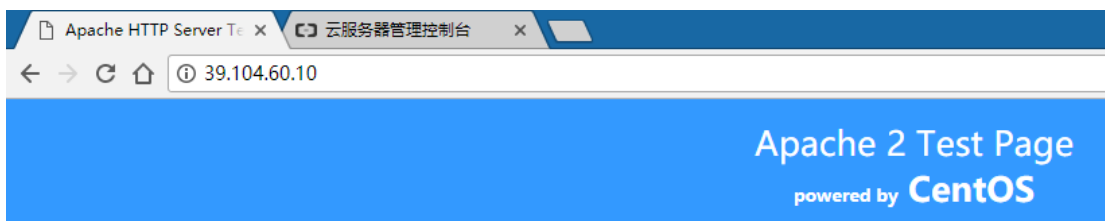
在文件中搜索“ServerName”

```
#
ServerName www.example.com:80
```

将前面的“#”去除，保存退出，重启 apache

```
[root@iZhp38oxmb9oefvqu4ep4tZ ~]# service httpd start
Starting httpd: [ OK ]
[root@iZhp38oxmb9oefvqu4ep4tZ ~]#
```

测试访问，在地址栏中输入 ip 地址直接访问（关闭防火墙）:



This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can see this site is working properly.

测试 php 是否可以运行（默认的 Apache 站点目录: `/var/www/html`）:

创建一个 `index.php` 文件

```
<?php
phpinfo();
```



运行 php 看到页面:



2、MySQL 的安装与初始化

```
#yum install mysql-server
```

```
=====
Package                Arch                Version
=====
Installing:
mysql-server            x86_64              5.1.73-8.el6_8
Installing for dependencies:
mysql                   x86_64              5.1.73-8.el6_8
perl-DBD-MySQL          x86_64              4.013-3.el6
perl-DBI                x86_64              1.609-4.el6

Transaction Summary
=====
Install                4 Package(s)

Total download size: 10 M
Installed size: 29 M
Is this ok [y/N]: y
```

初始化操作:

```
#service mysqld start      【启动】
```

```
You can test the MySQL daemon with mysql-test-run.pl
cd /usr/mysql-test ; perl mysql-test-run.pl

Please report any problems with the /usr/bin/mysqlbug script!

Starting mysqld:          [ OK ]
```

```
# mysql_secure_installation
```

```
All done!  If you've completed all of the above steps, your MySQL
installation should now be secure.

Thanks for using MySQL!
```

测试进行命令行登录:

```
#mysql -uroot -p
```

如果需要远程登录则需要修改登录主机:

```
mysql> update user set host = '%' where host = '127.0.0.1';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select user,host from user;
+-----+-----+
| user | host      |
+-----+-----+
| root | %         |
| root | localhost |
+-----+-----+
2 rows in set (0.00 sec)
```

重启 MYSQL 或者刷新权限:

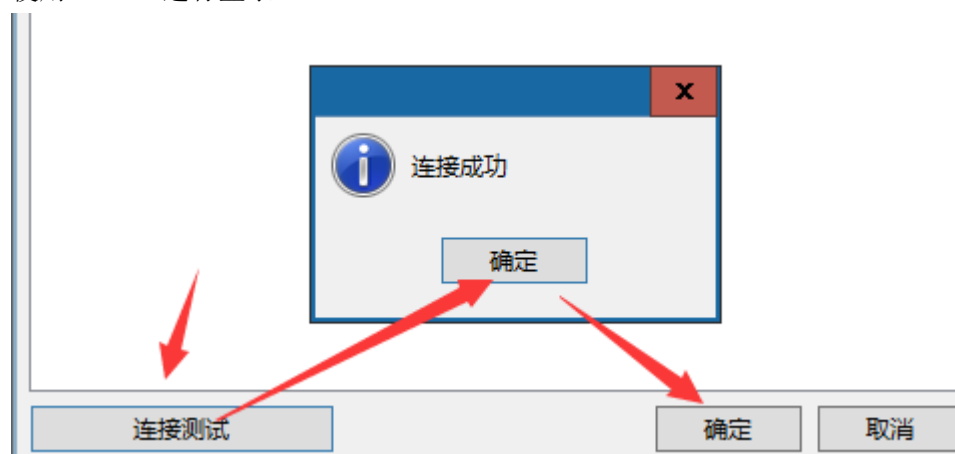
Mysql> flush privileges;

```
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

阿里云上的安全组端口放行:

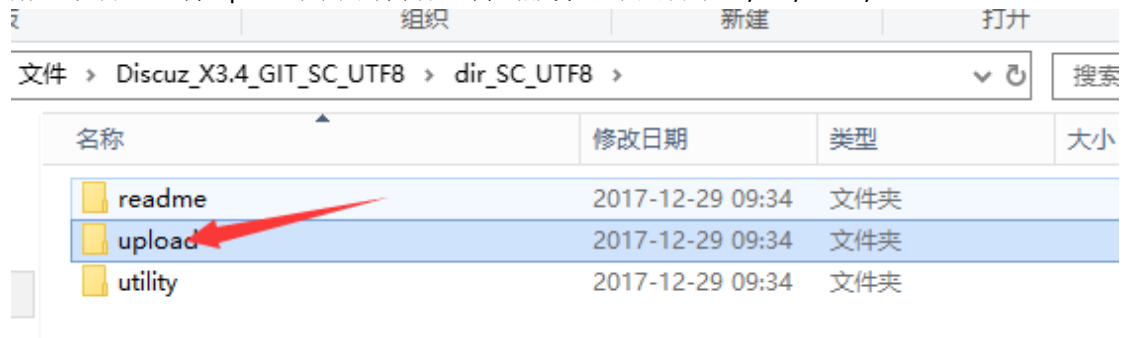


使用 navicat 进行登录:



3、项目上线

解压项目包，将 upload 其中的内容上传到服务器站点目录 (/var/www/html)



①使用 filezilla 上传需要的代码文件

②传完之后打开网站的首页，会运行 DZ 的安装向导

a. 选择同意协议



b. 赋予指定目录写权限

目录、文件权限检查

目录文件	所需状态	当前状态
./config/config_global.php	✔ 可写	✘ 目录不存在
./config/config_ucenter.php	✔ 可写	✘ 目录不存在
./config	✔ 可写	✘ 不可写
./data	✔ 可写	✘ 不可写
./data/cache	✔ 可写	✘ 不可写
./data/avatar	✔ 可写	✘ 不可写
./data/plugindata	✔ 可写	✘ 不可写
./data/download	✔ 可写	✘ 不可写
./data/addonmd5	✔ 可写	✘ 不可写
./data/template	✔ 可写	✘ 不可写
./data/threadcache	✔ 可写	✘ 不可写
./data/attachment	✔ 可写	✘ 不可写

```
# chmod 777 -R /var/www/html
```

函数依赖性检查

需要mysql扩展

函数名称	检查结果	建议
mysqli_connect()	✘ 不支持	advice_mysqli_connect
gethostbyname()	✔ 支持	无

```
#yum install php-mysql
```

```
=====  
Package                Arch                Version  
=====  
Installing:  
php-mysql              x86_64             5.3.3-49.el  
Installing for dependencies:  
php-pdo                x86_64             5.3.3-49.el  
  
Transaction Summary  
=====  
Install                2 Package(s)  
  
Total download size: 166 k  
Installed size: 384 k  
Is this ok [y/N]: █
```

重启 Apache:



```
Complete!
[root@iZhp38oxmb9oefvqu4ep4tZ ~]# service httpd restart
Stopping httpd:          [ OK ]
Starting httpd:         [ OK ]
[root@iZhp38oxmb9oefvqu4ep4tZ ~]#
```

重启之后保证所有的配置项都是绿色的勾才可以下一步。

c. 选择 DZ 的安装方式



d. 填写数据库与管理员的信息

填写数据库信息

数据库服务器:	<input type="text" value="localhost"/>	数据库服务器地址, 一般为 localhost
数据库名:	<input type="text" value="ultrax"/>	
数据库用户名:	<input type="text" value="root"/>	
数据库密码:	<input type="text" value="LYL4yEBiiW8lOJl7"/>	
数据表前缀:	<input type="text" value="pre_"/>	同一数据库运行多个论坛时, 请修改前缀
系统信箱 Email:	<input type="text" value="admin@admin.com"/>	用于发送程序错误报告

填写管理员信息

管理员账号:	<input type="text" value="admin"/>	
管理员密码:	<input type="password" value="....."/>	管理员密码不能为空
重复密码:	<input type="password" value="....."/>	
管理员 Email:	<input type="text" value="admin@admin.com"/>	

e. 安装完成



f. 首页

