

Linux-Study

1 Vmware16安装和使用 Centos7.6虚拟机

1.1 CentOS 下载

1.2 安装步骤

1.3 网络连接的三种方式

1.4 虚拟机克隆

1.5 虚拟机快照-存档|读档

1.6 虚拟机的迁移和删除

1.7 安装 vmtools 用于和windows主机共享文件夹

1.8 Linux目录结构详解 (一切皆文件)

1.9 Linux远程登录

2 vim快捷键-系统关机-重启-登录-退出

2.1 vi和vim快捷键使用

2.2 关机&重启&数据同步命令

2.3 用户登录和退出系统

3 用户管理

3.1 添加用户

3.2 设置/修改用户密码

3.3 删除用户

3.4 查询用户-切换用户-查看当前登录用户

3.5 用户组和 etc/下相关文件信息

4 运行级别-找回root密码-帮助指令

5 文件目录指令

5.1 pwd ls cd

5.2 mkdir rmdir rm -rf

5.3 touch cp rm

5.4 mv cat less echo head tail

5.5 > 和 >> 指令

5.6 ln history date cal

5.7 查找指令find which grep

5.8 压缩与解压缩 gzip gunzip

5.9 压缩与解压缩 zip unzip

5.10 压缩与解压缩 tar

6 组和rwx权限 P40-P62 集未看

7 网络配置-进程-服务管理

7.1 修改网络配置文件实例

7.2 设置主机名和hosts映射

7.3 进程-ps-kill-pstree指令

7.4 服务管理-systemctl指令

7.5 打开或关闭指定端口-firewall指令

7.6 动态监控系统-top指令

7.7 查看系统网络状态-netstat

8 软件包管理-rpm-yum-安装JDK-Tomcat

8.1 rpm 与 yum 包管理

8.2 安装JDK8

8.3 安装配置Tomcat

8.4 安装 nginx

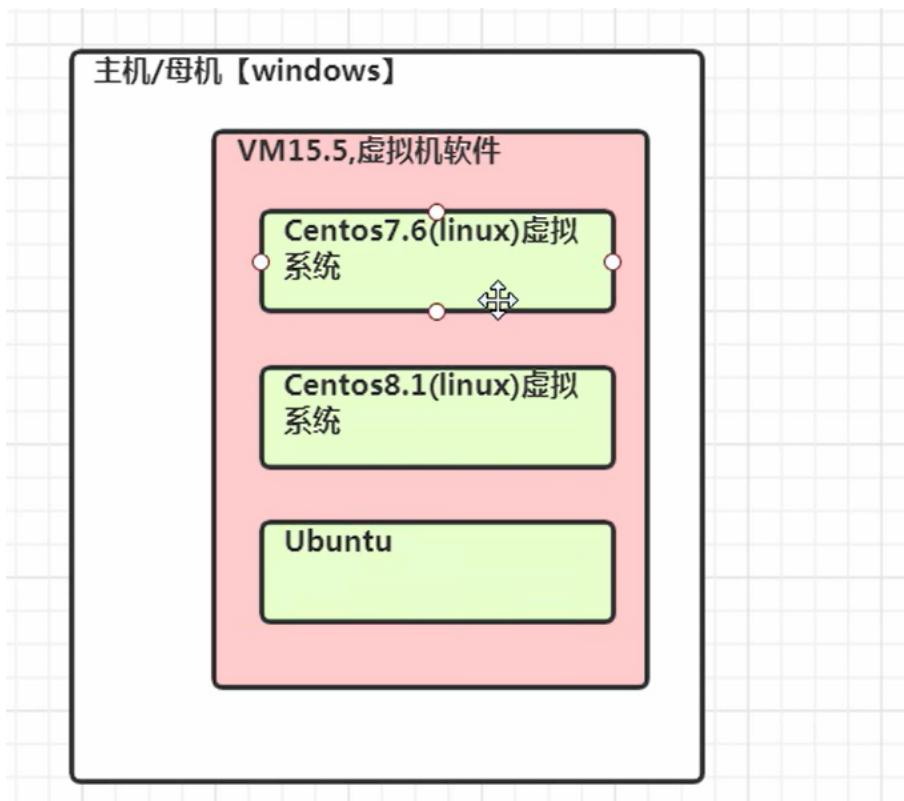
8.5 安装 mysql 5.7

9 P91-P153集未看

| Linux主要的发行版：Ubuntu(乌班图)、RedHat(红帽)、CentOS、Debian(蝶变)等

1 Vmware16安装和使用 Centos7.6虚拟机

| vm和linux的关系，可以创建多个不同版本的虚拟机



1.1 CentOS 下载

```
1  
2  CentOS-7-x86_64-DVD-2009.iso  
3  http://mirrors.aliyun.com/centos/7/isos/x86_64/  
4  
5  CentOS-8.3.2011-x86_64-dvd1.iso  
6  http://mirrors.aliyun.com/centos/8/isos/x86_64/
```

1.2 安装步骤

WORKSTATION 16 PRO™



创建新的虚拟机



打开虚拟机



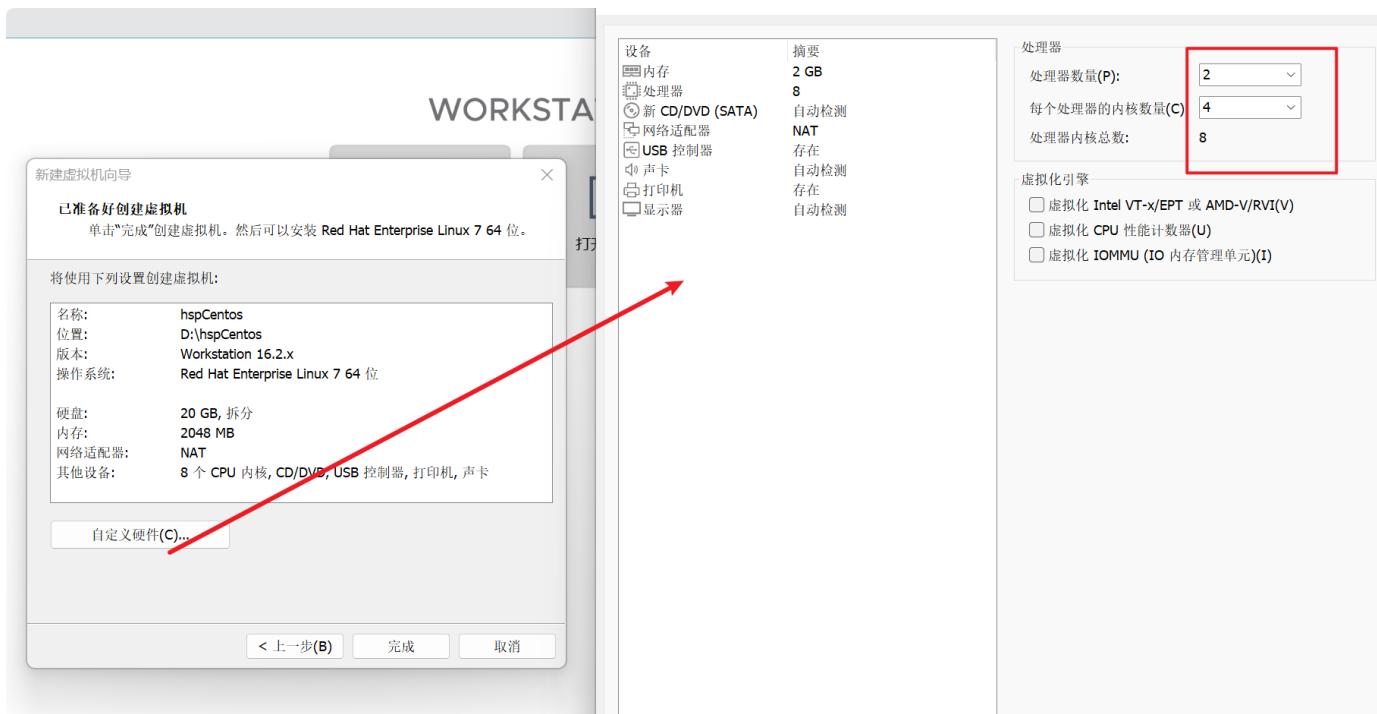
连接远程服务器







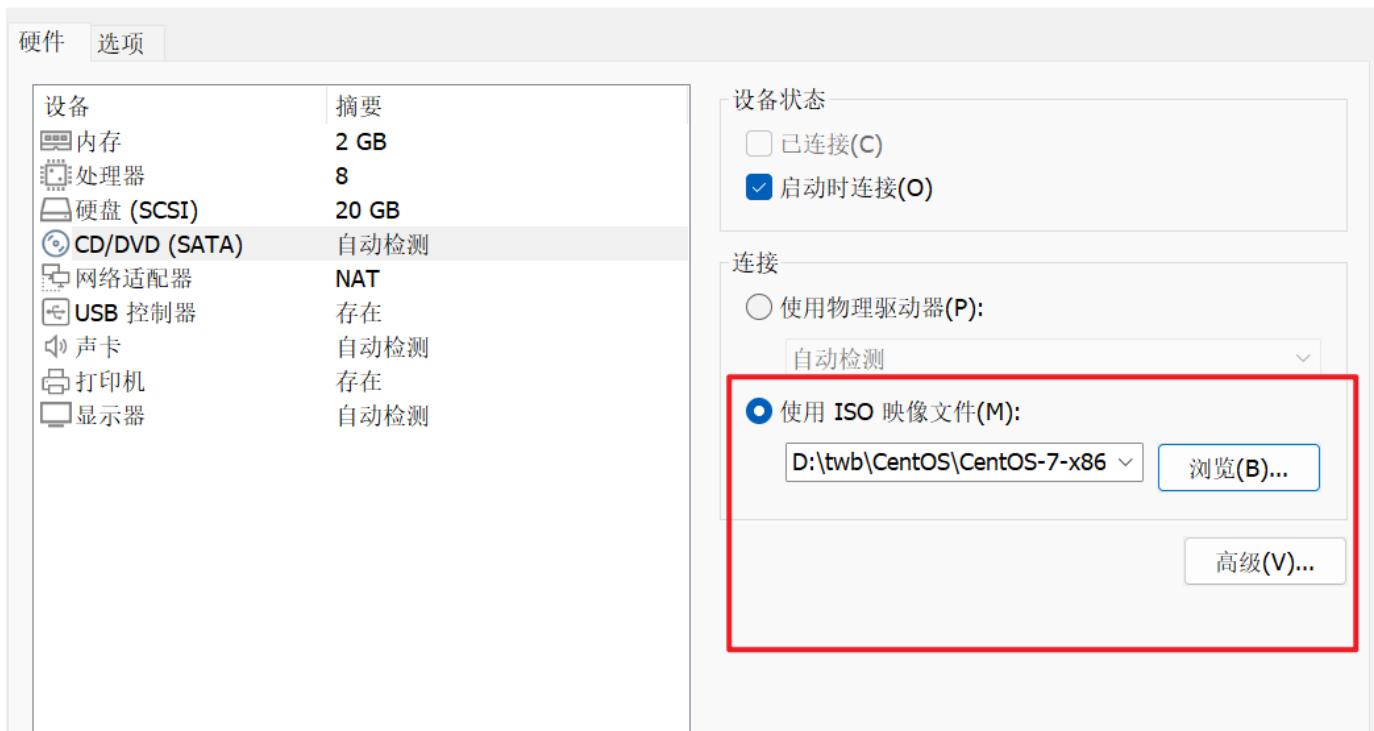
其它默认



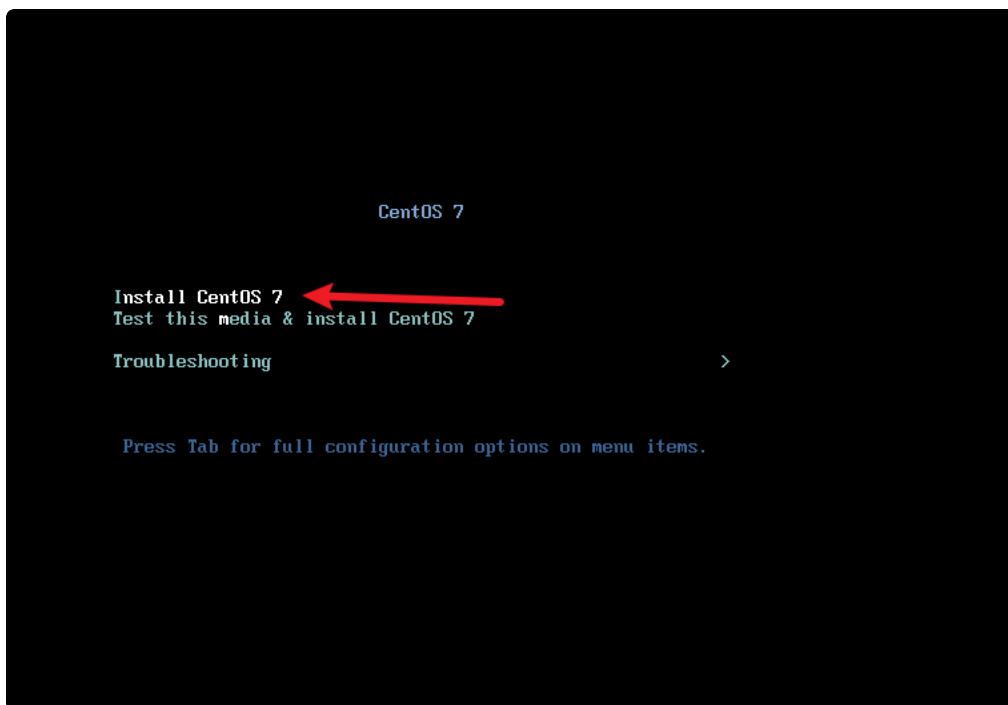
此时创建的虚拟机还没有真正安装操作系统，需要挂载 CD/DVD镜像

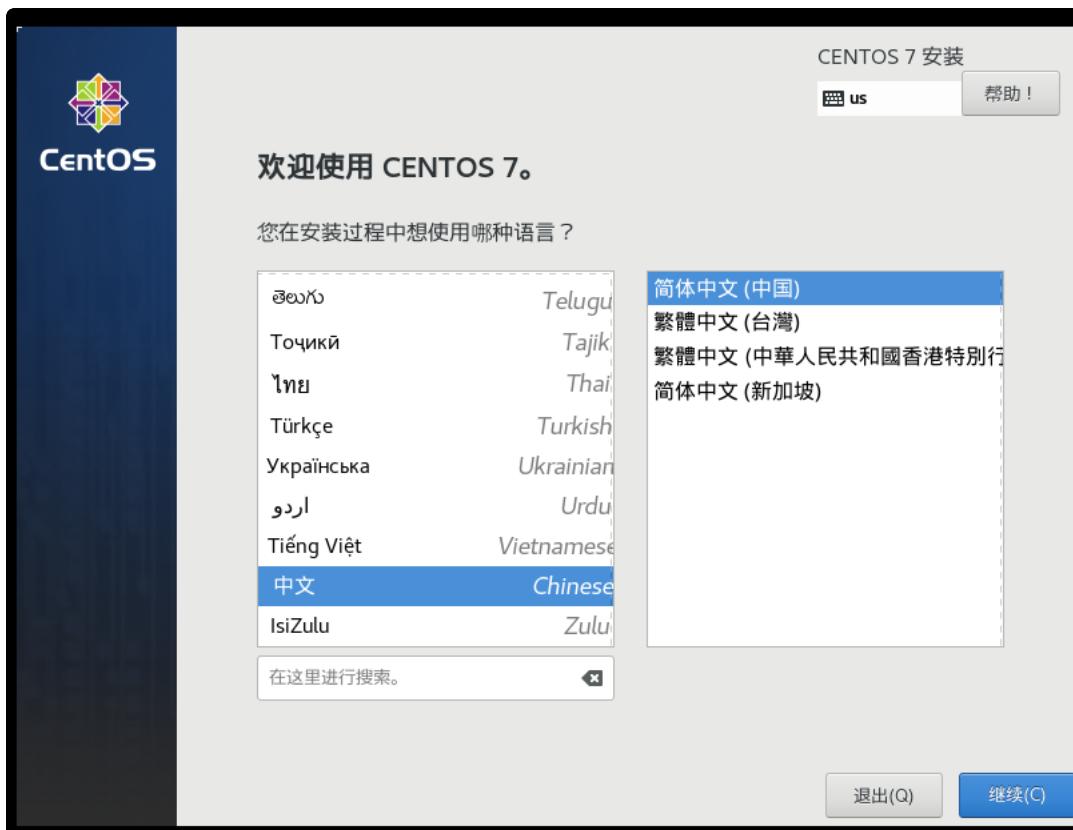
虚拟机设置

X



启动虚拟机，选择第一个 CentOS 7，点击回车等待安装即可





点击 软件选择，进行如下环境配置勾选

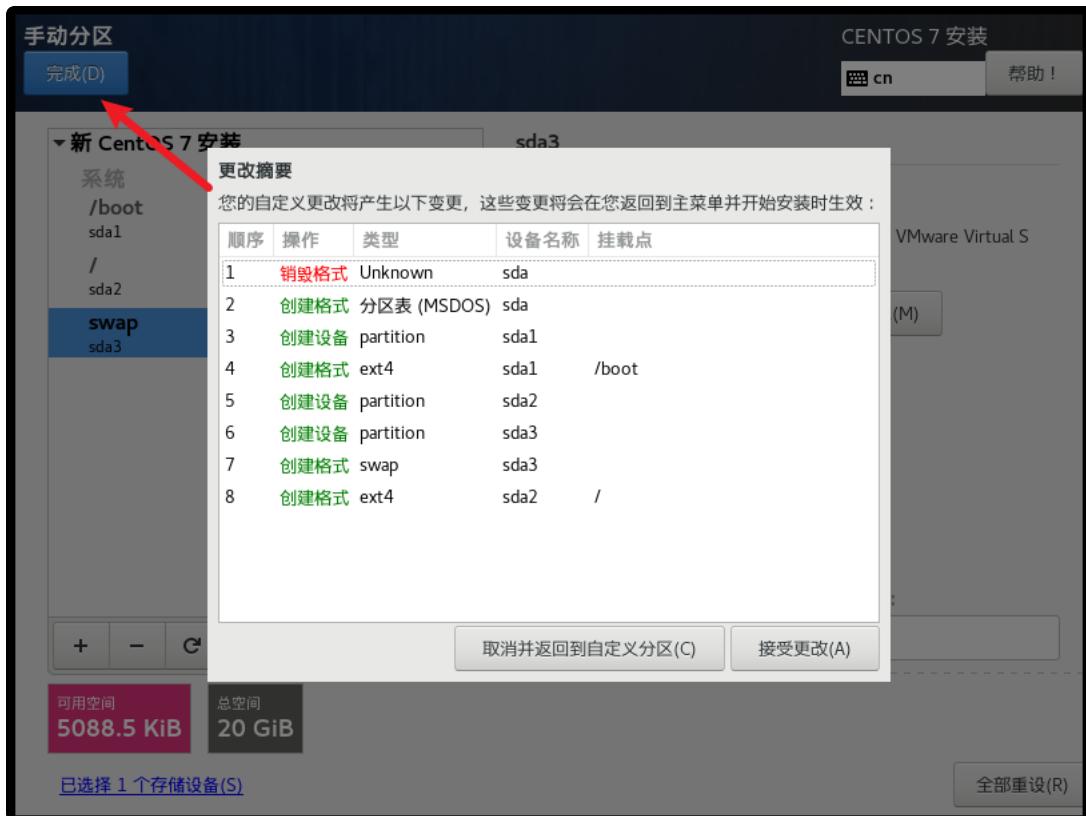






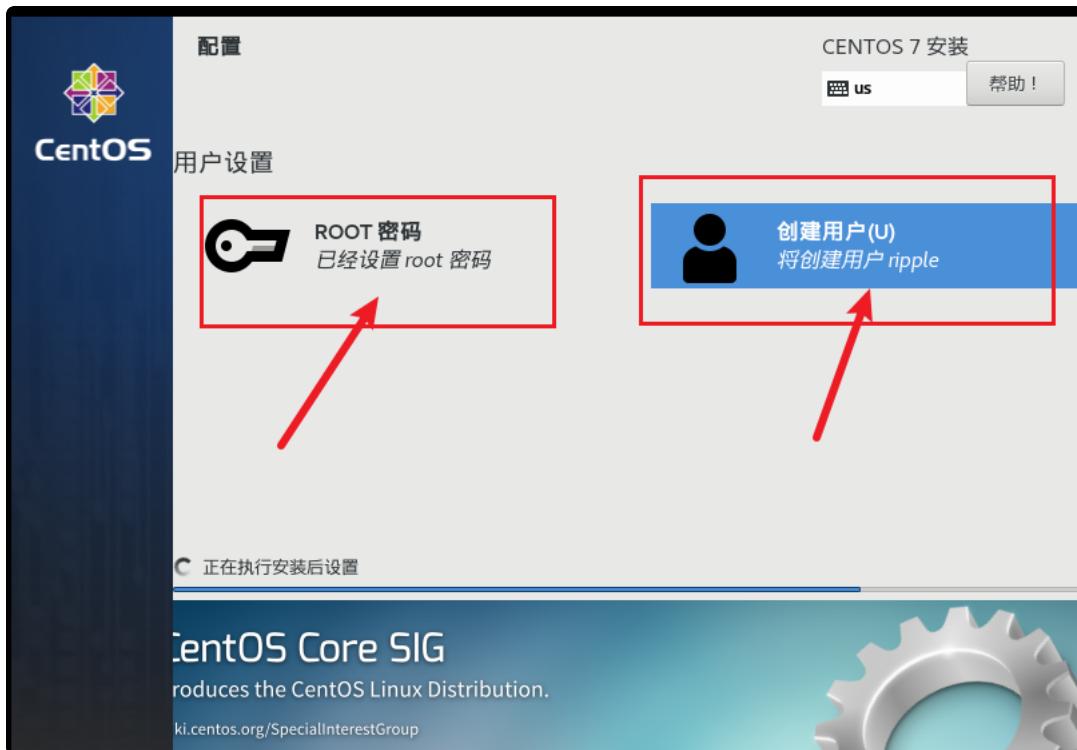
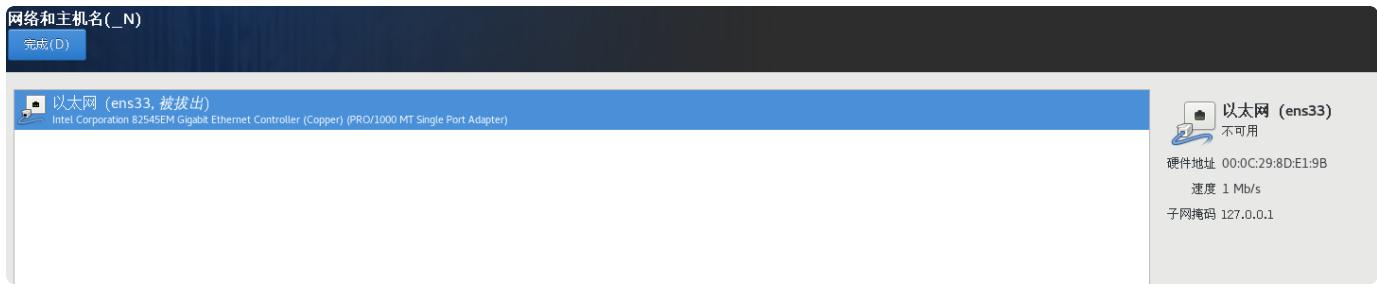
点击完成，设置3个分区统一 修改标准分区、ext4，swap的文件系统类型默认 swap即可

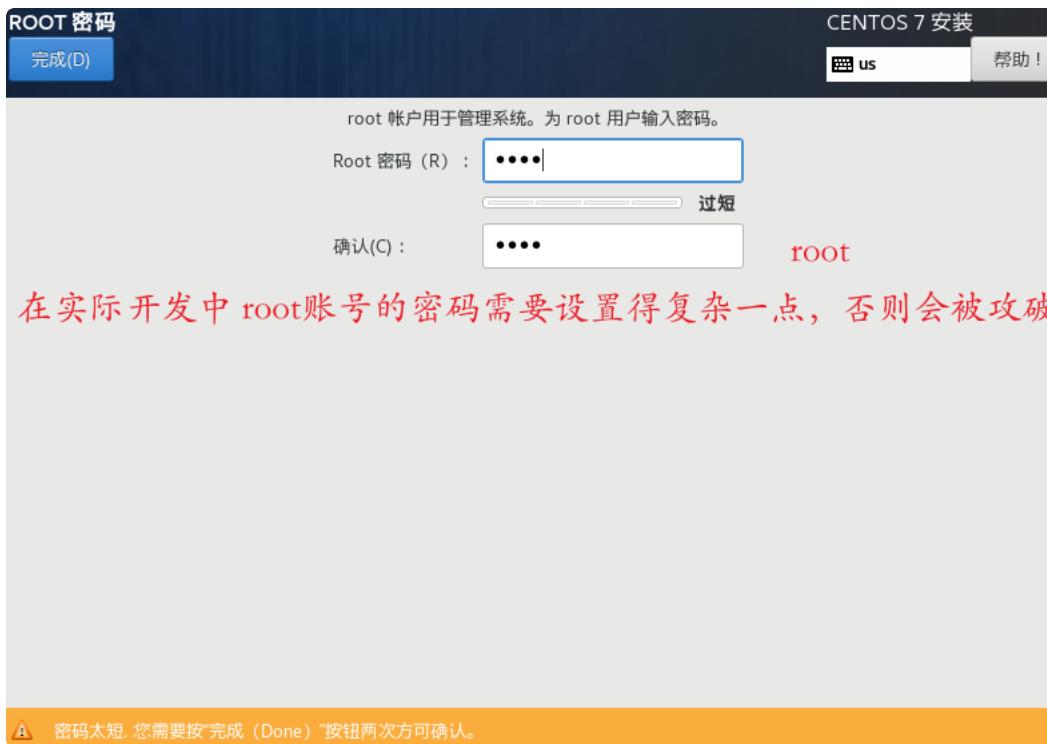




这里得网络 ens33 需要自己得电脑插入网线才可以连接，安装过程中设置账号密码，之后重启





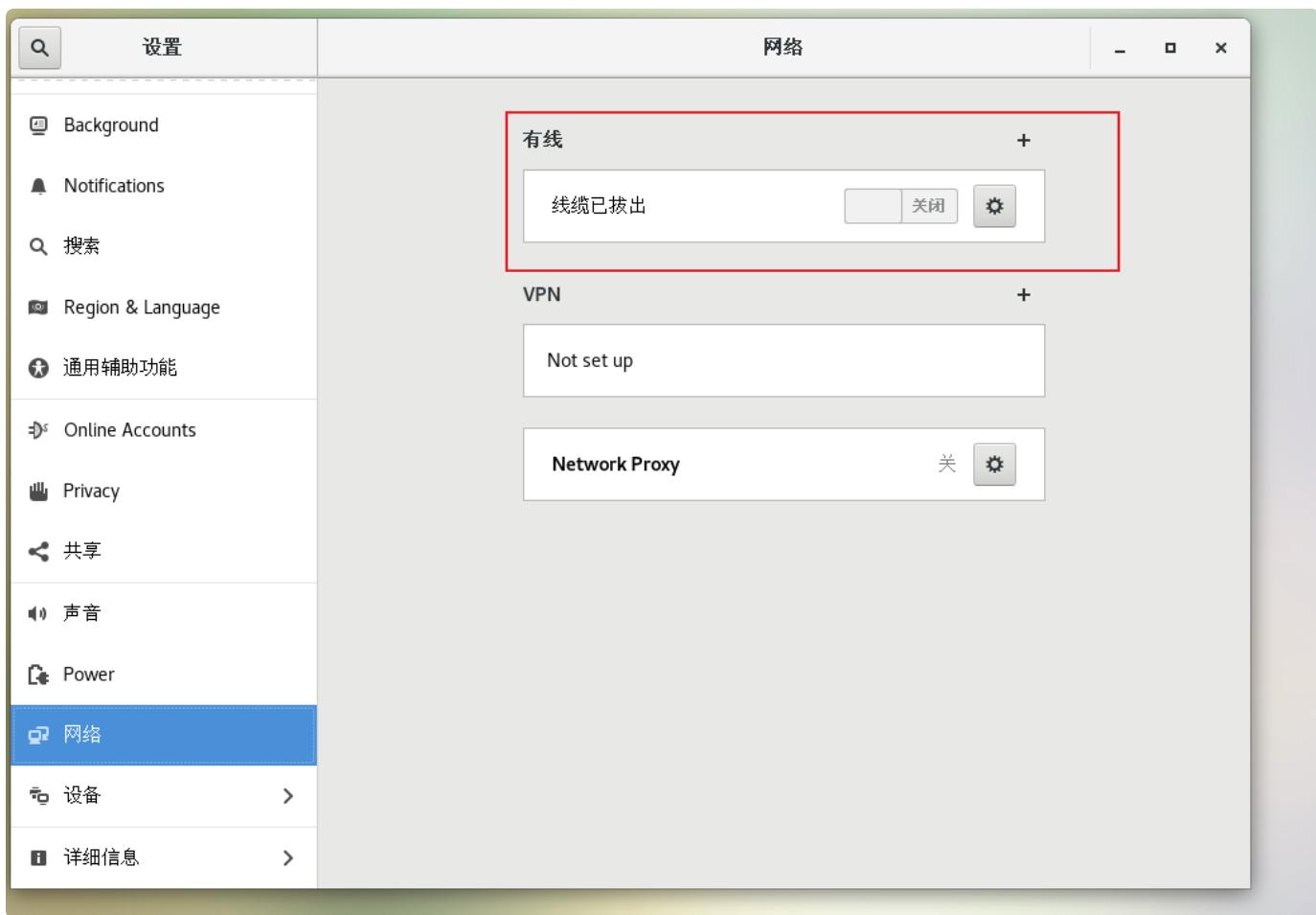


默认是以普通用户登录，可以点击下方得“未列出”，输入 root用户名和密码，以 root方式登录



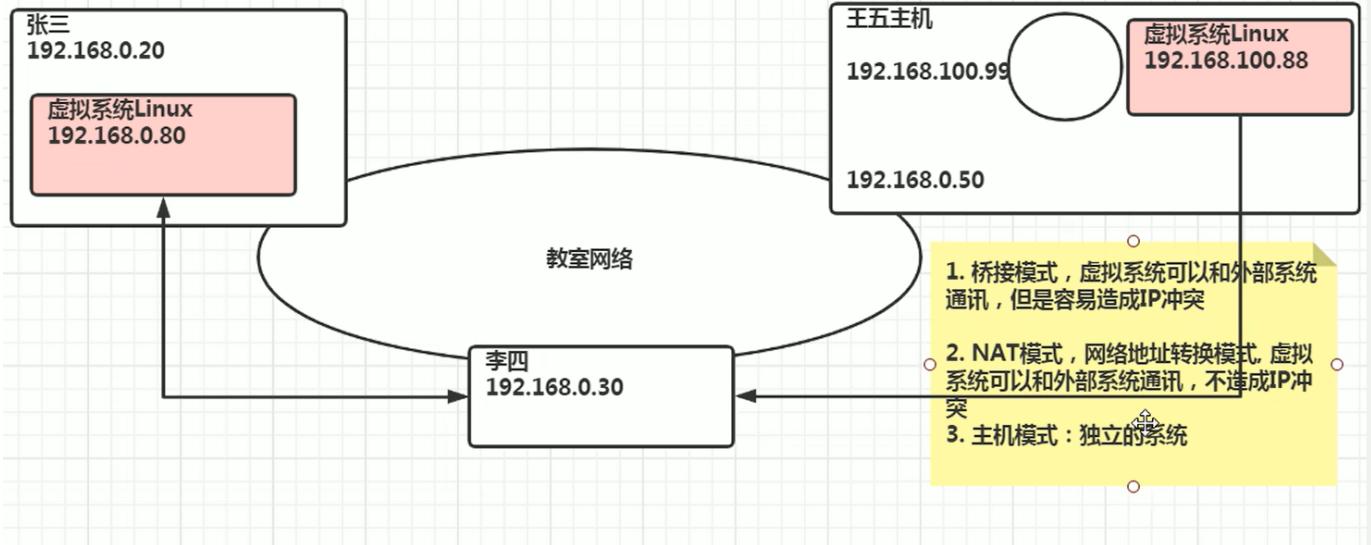
1.3 网络连接的三种方式

目前没有插入网线得情况下



示意图

网络连接的三种模式



1.4 虚拟机克隆

如果已经安装了一台Linux操作系统，直接克隆即可：

1. 方式1：直接拷贝一份安装好的虚拟机文件
2. 方式2：使用vmware的克隆操作，克隆时需要先关闭linux系统

克隆之后的虚拟机和原来的用户、密码、安装的软件、程序、环境都是一样的。



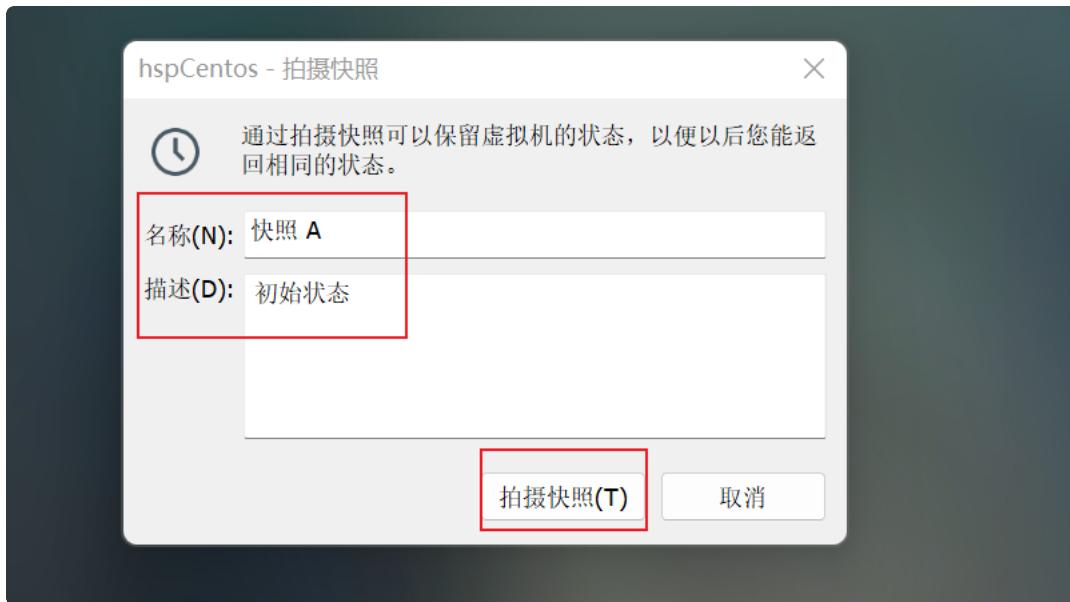


1.5 虚拟机快照-存档|读档

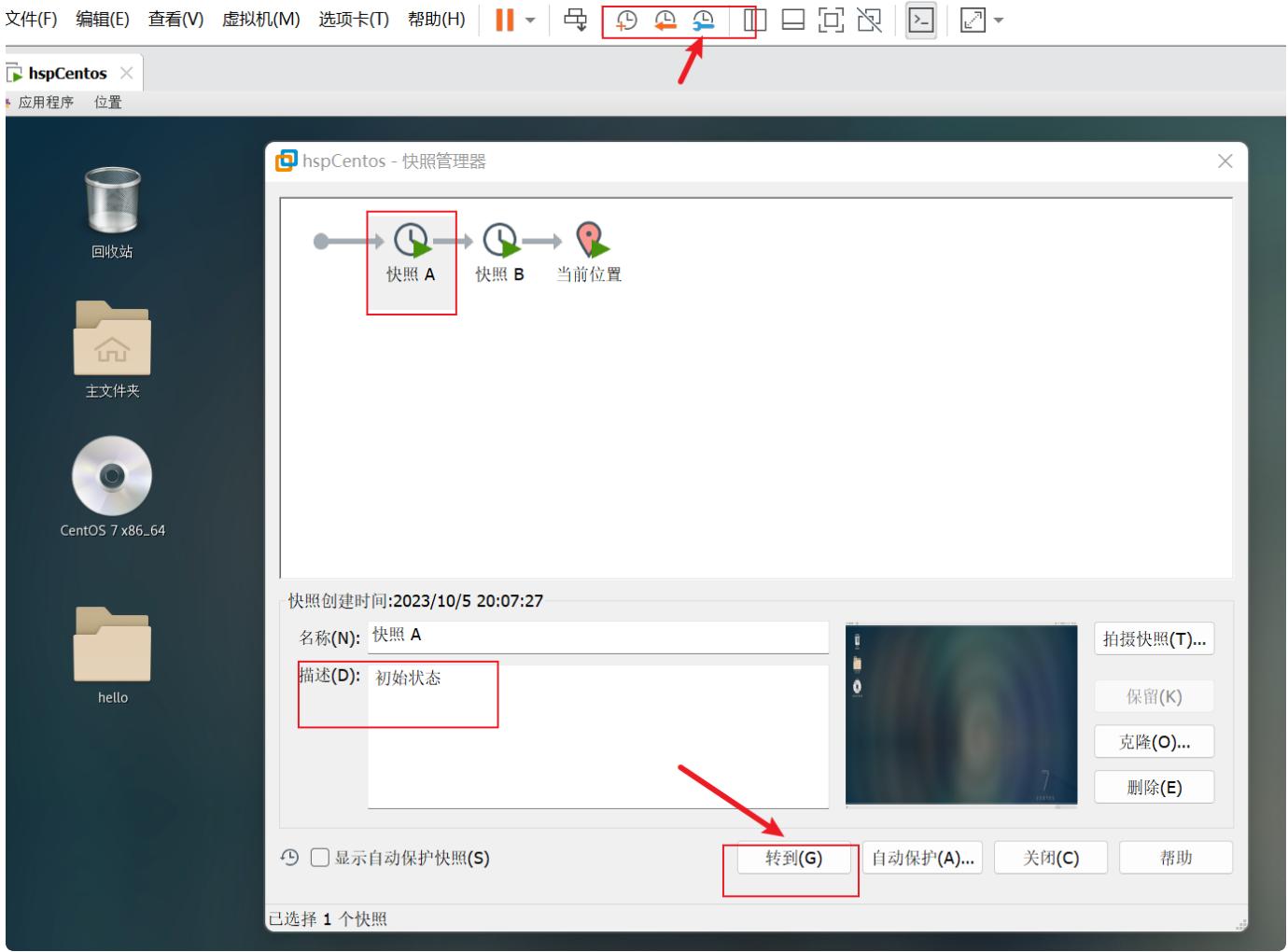
如果在使用虚拟机的时候，想回到原先的某一个状态，也就是说你担心可能有些误操作造成系统异常需要回到原先某个正常运行的状态

应用场景：

1. 安装好系统以后，先做一个快照 A
2. 进入到系统后，创建一个文件夹，再保存一个快照 B
3. 回到系统刚刚安装好的状态，即快照 A
4. 也可以再次回到 B



选择想要回到的状态，点击“转到”



1.6 虚拟机的迁移和删除

虚拟机系统安装后，本质就是一个文件放在文件夹里，可以把安装好的虚拟机系统这个文件夹整体拷贝或剪切到另外地方使用，删除也很简单，从磁盘中删除对应文件夹即可

1.7 安装 vmtools 用于和windows主机共享文件夹

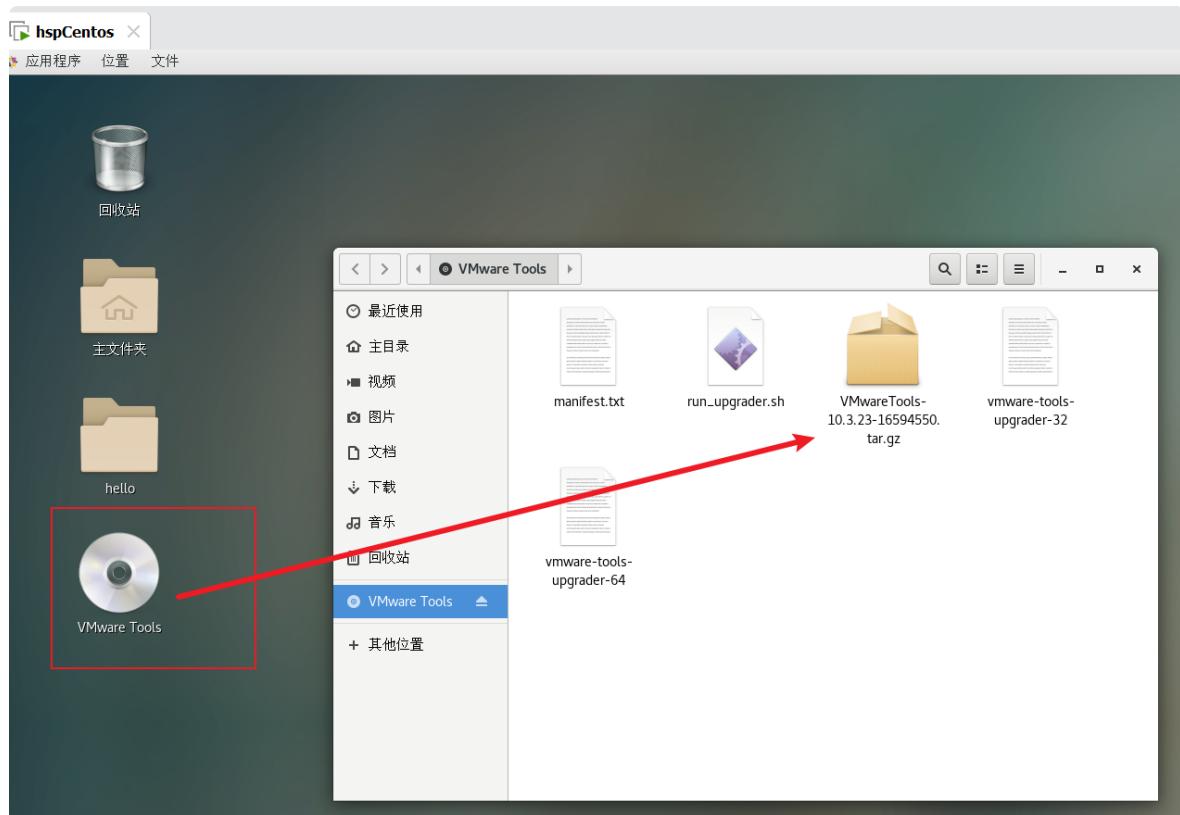
该工具可以在windows下更好的管理vm虚拟机，可以设置windows和CentOS虚拟机的文件共享
安装步骤：

1. 右击虚拟机 找到 "重新安装 VMware Tools" 选项，如果该项是灰色的无法点击，需要关闭虚拟机

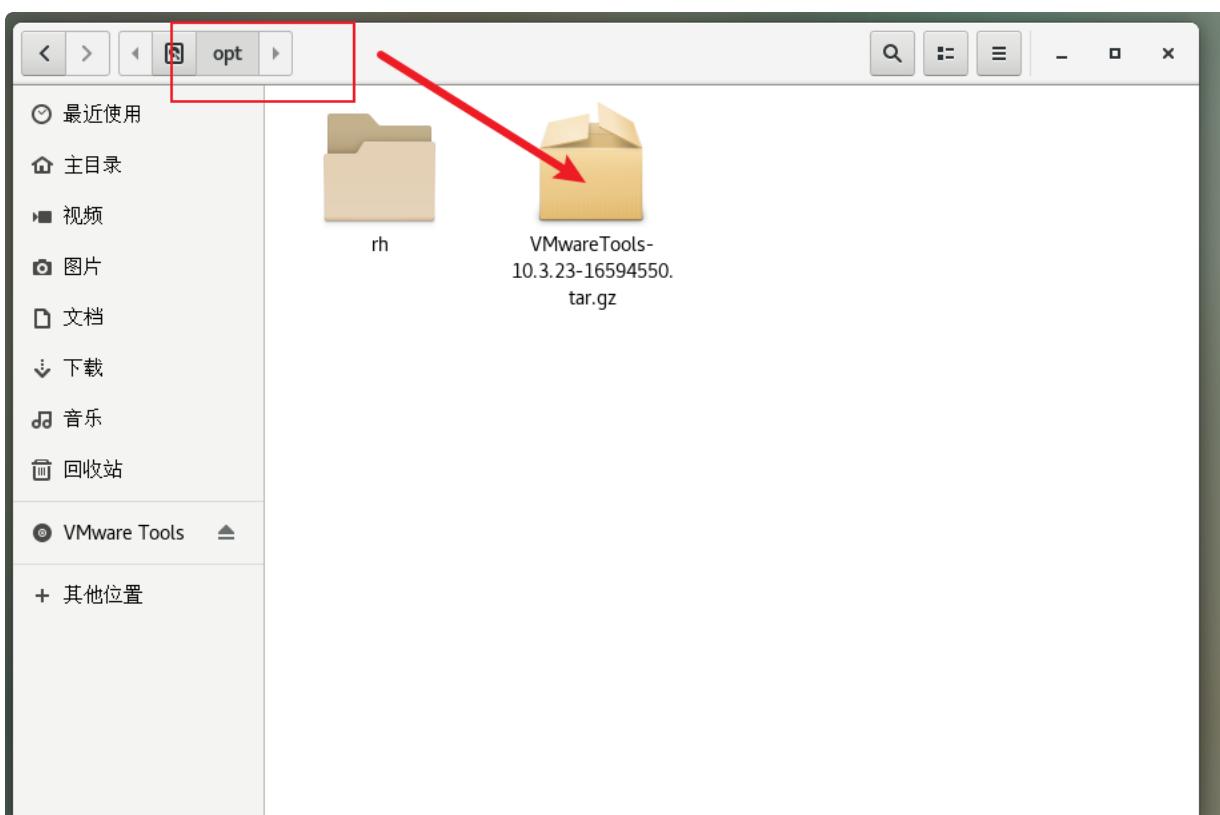
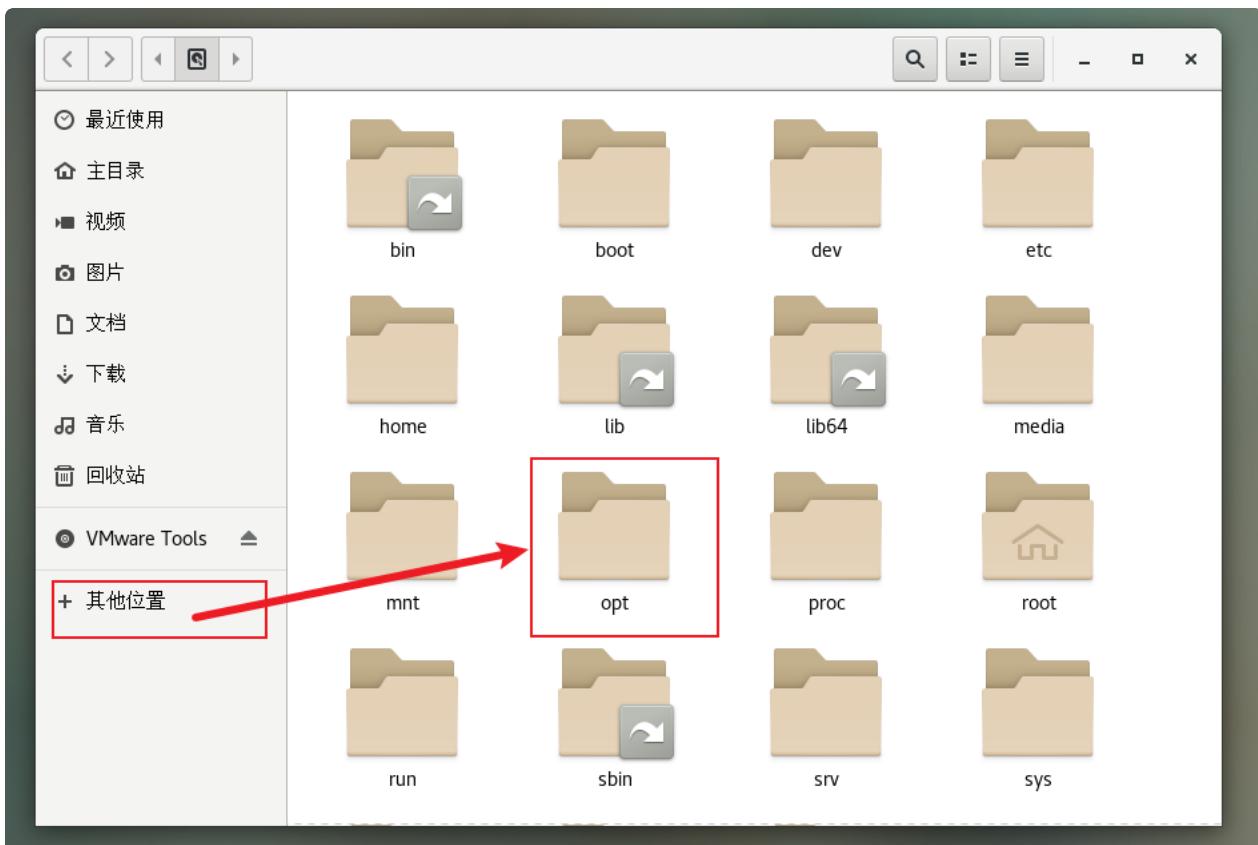
然后设置 CD/DVD连接选择使用物理驱动器(自动检测)



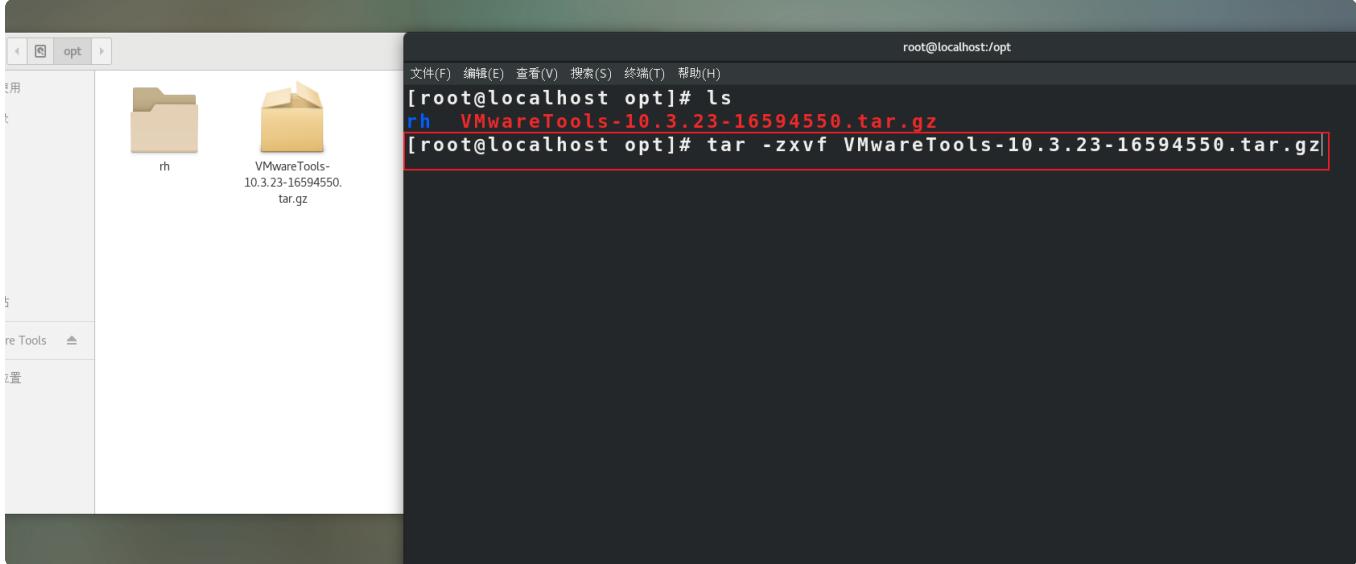
- 之后再次开启虚拟机过程中，右击找到“重新安装 VMware Tools”选项，等开启完毕后，在桌面看到一个 VMware Tools 光盘驱动图标，里面有一个工具 tar.gz 压缩包



- 复制一份该 tar.gz 压缩包，找到“其他位置”->计算机->粘贴到 opt/ 目录下



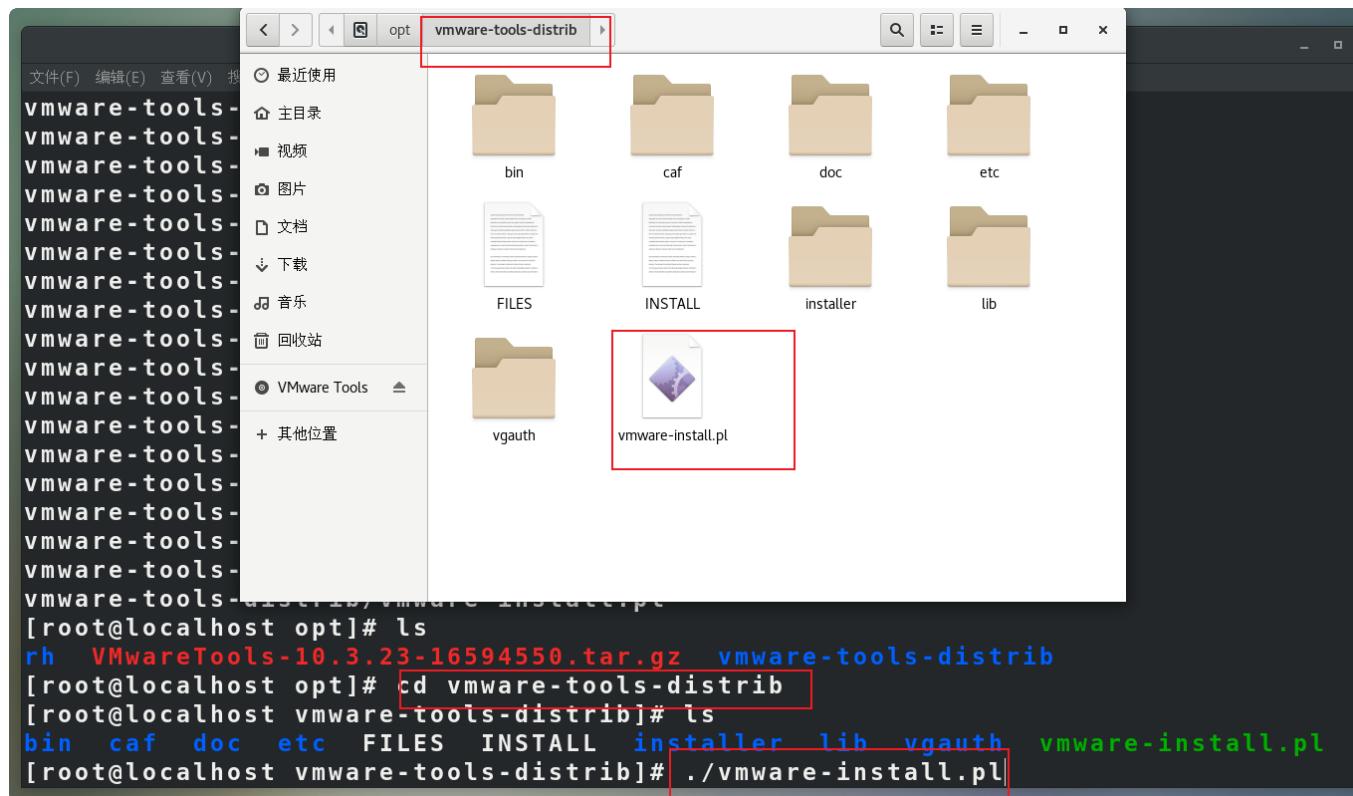
4. 在终端下进入到 opt目录下 cd /opt 解压该工具包，执行命令：tar -zxvf 指定压缩包，然后回车



5. 进入解压后的目录 cd vmware-tools-distrib/ 下执行 ./vmware-install.pl 之后一直回车安装即可。

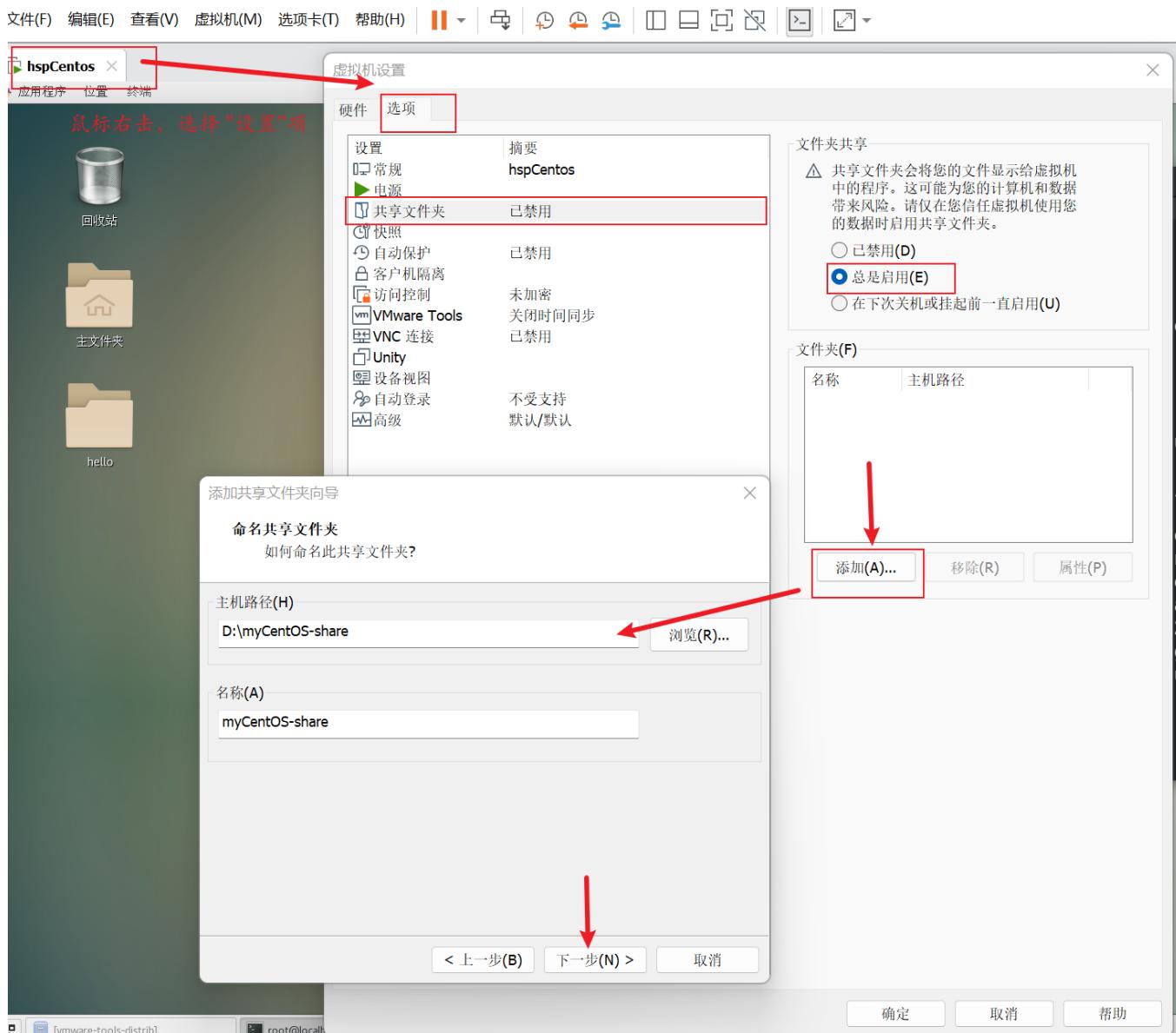
注意安装此工具需要 gcc支持，在安装虚拟机的时候已经配置了基本的程序环境

```
[root@localhost vmware-tools-distrib]# gcc -v
使用内建 specs。
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux
目标 : x86_64-redhat-linux
配置为 : ../configure --prefix=/usr --mandir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla-shared --enable-threads=posix --enable-checking=re
enable_cxa_atexit --disable-libunwind-exceptions --enable-linker-build-id --with-linker-hash-style-gnu -c,obj-c++,java,fortran,ada,go,lto --enable-plugin --enable-libgcj --with-isl=/builddir/build/BUILD/gcc-4.8.5
-linux/isl-install --with-cloog=/builddir/build/BUILD_64-redhat-linux/cloog-install --enable-gnu-indirect-c --with-arch_32=x86-64 --build=x86_64-redhat-linux
线程模型 : posix
gcc 版本 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)
[root@localhost vmware-tools-distrib]# |
```



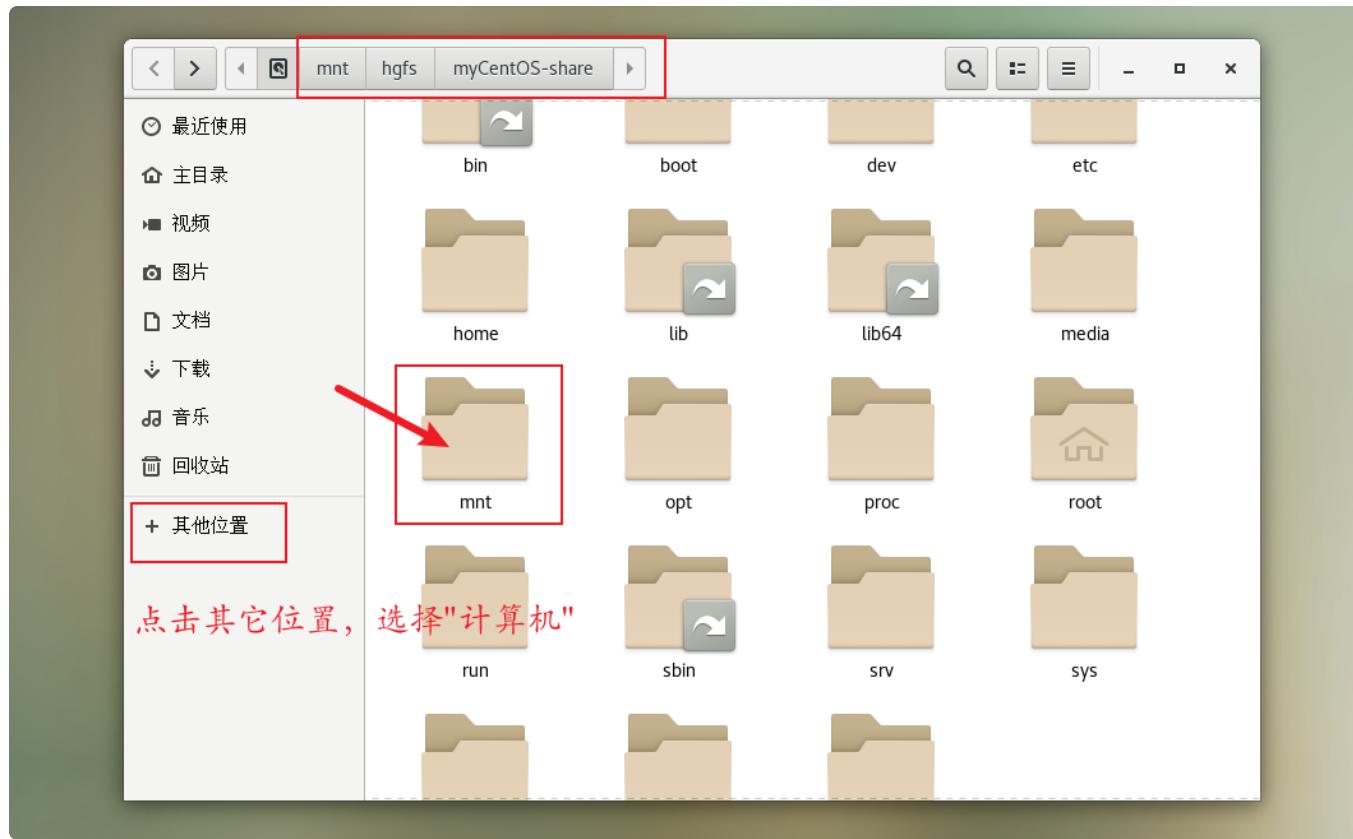
6. 设置共享文件夹：

先在windows电脑下新建一个文件用于共享，比如：D:\myCentOS-share，设置虚拟机选项，找到共享文件夹->总是启用->添加共享的本地磁盘中的文件夹->点击下一步



7. 找到 `mnt/hgfs/` 目录下可以看到和windows主机共享的文件夹了，此时在windows或者Linux虚拟机中修改文件夹中的数据都是同步变化共享的。

备注：但是在实际开发中，文件的上传和下载是需要使用远程登录的方式来完成的(后面会讲解)



1.8 Linux目录结构详解 (一切皆文件)

在Linux中，所有的硬件配置都是映射成文件形式来管理和体现 -> 一切皆文件

1. linux的文件系统采用树状目录结构，最上层是**根目录 "/"**，然后在此根目录下创建其他的目录用于存放不同的配置文件和程序指令，以下列出了常用和重要的目录：

/root (管理员权限者的用户主目录)

/sbin 、 /usr/sbin (s就是Super User存放的是系统管理员使用的系统管理程序)

/home (普通用户的主目录)

/bin (该目录存放着最经常使用的命令)

/etc (系统管理所需要的配置文件和子目录，比如安装mysql数据库 my.conf)

/usr 用户的很多应用程序和文件都存放在此目录下，类似于windows中的 programs files目录

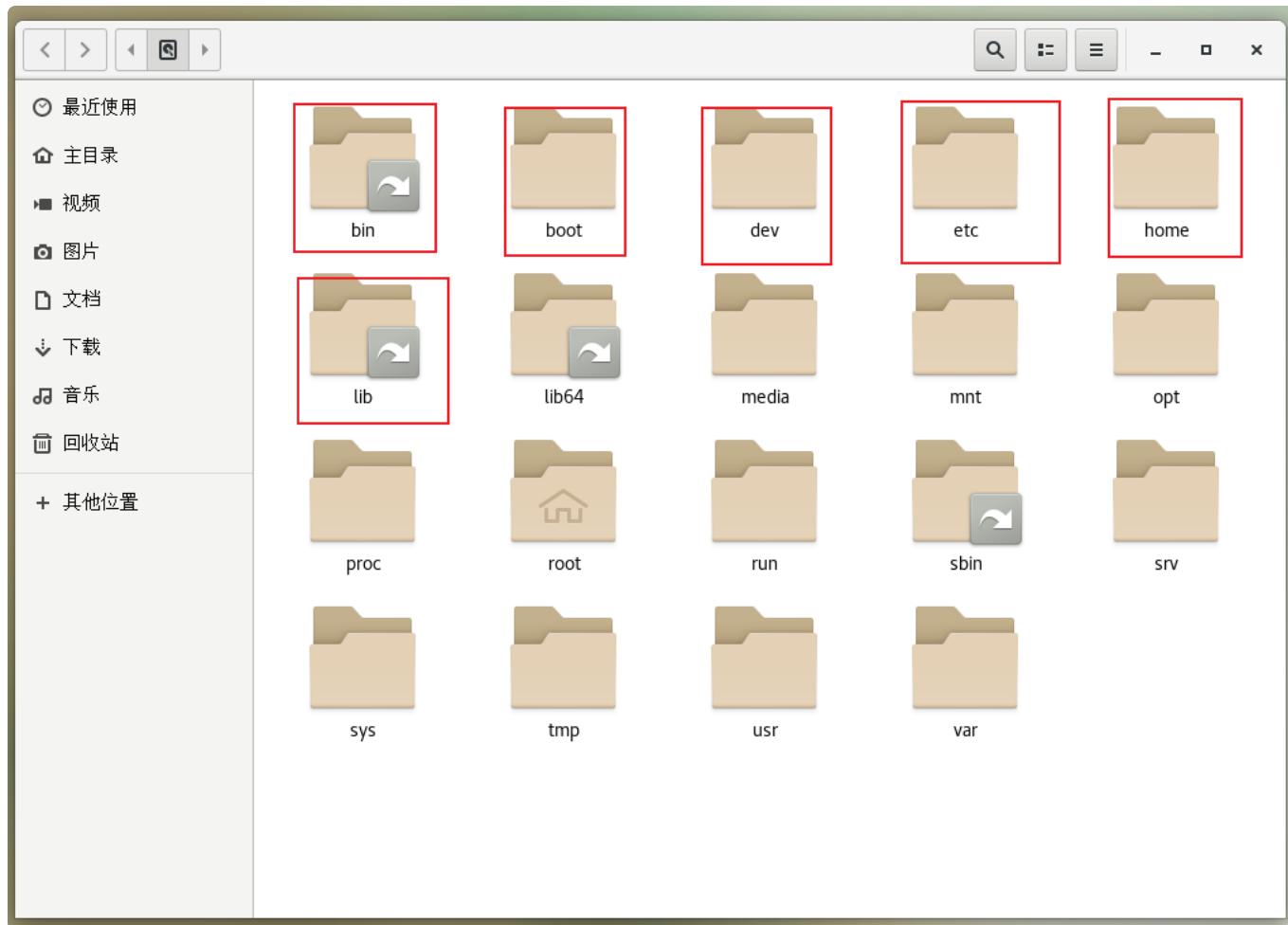
/boot 存放启动Linux时使用的一些核心文件，包括一些连接、镜像文件等

/dev 类似于windows的设备管理器

/media U盘、光驱设备挂载到该目录下

/mnt 用户临时挂载别的文件系统

/opt 给虚拟机额外安装软件程序所存放的目录，比如安装Mysql数据库、JDK、Tomcat等 就可以放到该目录下，默认为空



1.9 Linux远程登录

Xshell 、Xftp 家庭免费版下载官网：<https://www.xshell.com/zh/free-for-home-school/>

Linux虚拟机网络连接问题：

https://blog.csdn.net/weixin_58279299/article/details/118312927?ops_request_misc=&request_id=&biz_id=102&utm_term=linux%E8%99%9A%E6%8B%9F%E6%9C%BAping%E4%B8%8D%E9%80%9A%E5%A4%96%E7%BD%91&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-4-118312927.nonecase&spm=1018.2226.3001.4187

2 vim快捷键-系统关机-重启-登录-退出

2.1 vi和vim快捷键使用

正常模式：以 vim 指令打开一个文件进入的默认模式

编辑插入模式：按快捷键 INSERT 或者输入 i 进入

命令模式：先按 ESC，再输入 : 进入

wq 保存并退出

q 退出

q! 强制退出，并且不保存

看看vi和vim各个模式的切换图



快捷键使用：

- 【正常模式下输入】 复制当前行： yy， 复制当前行向下的 3行： 3yy， 粘贴： p
- 【正常模式下输入】 删除当前行： dd， 删除当前向下的5行： 5dd
- 【命令模式下输入】 文件中查找某个单词： /关键字， 回车查找， 输入 n 查找下一个
- 【命令模式下输入】 设置文件显示行号： :set nu ， 隐藏行号： :set nonu
- 【正常模式下输入】 跳转定位到文件的最末行： G 后者 \$， 最首行： gg
- 【正常模式下输入】 先进入编辑模式修改文件输入内容，然后ESC退出到正常模式，撤销这个动作： u
- 【正常模式下输入】 先输入要跳转的行号，再按 Shift + g ，比如要跳转到文件的第66行： 66 Shift g

2.2 关机&重启&数据同步命令

Shell |

```
1 shutdown -h now //立刻关机, => 等价于 shutdown 命令
2 shutdown -h 1 // 1分钟后关机
3 shutdown -r now //现在重启虚拟机
4
5 halt // 关机, 同上
6 reboot //重启
7 sync //把内存的数据同步到磁盘
8
9 注意:
10 1. 不管是重启系统还是关闭, 首先要运行 sync 命令, 把内存数据写到磁盘, 防止丢失
11 2. 目前的 shutdown/reboot/halt 命令均已在关机前进行了 sync, 但以防万一
```

2.3 用户登录和退出系统

1. 登录时尽量少用 root 登录, 权限最高, 避免操作失误
2. 执行指令: su – 用户名 来切换成系统管理员身份
3. 在提示符下输入 logout 指令 即可退出当前用户, 注意该指令在图形化运行级别下无效

3 用户管理

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

3.1 添加用户

```

1 // 添加用户指令
2 useradd 用户名
3
4 // 给新创建的用户自定义家home目录
5 useradd -d /home/目录名 用户名
6
7 useradd -d /home/test king
8
9 比如：
10 在家目录home下的test目录下，添加一个 king 用户
11 /home/test 就是该用户的工作目录(主目录、家目录)
12
13 注意：
14 1. 当创建用户成功后，会默认自动的在home下创建和用户同名的目录，作为该用户的工作目录
15 2. 也可以通过指令来给新创建的用户指定家home目录

```

```

[root@localhost ~]# cd /home    进入到当前登录用户的家目录
[root@localhost home]# ls
ripple
[root@localhost home]# useradd milan    添加用户
[root@localhost home]# ls
milan ripple
[root@localhost home]# useradd -d /home/test king
[root@localhost home]# ls    添加一个用户king，并自定义其工作目录test
milan ripple test
[root@localhost home]# pwd
/home

```

3.2 设置/修改用户密码

下次以该用户名登录时，就需要用新设置的密码

```

1 // 设置密码 注意：必须指定用户，否则默认是给当前登录的用户设置密码
2 passwd 用户名
3
4 // 显示当前用户所在的目录路径
5 pwd

```

```
[root@localhost test]# passwd milan      设置用户 milan 的登录密码
更改用户 milan 的密码。
新的 密码： 123456
无效的密码： 密码少于 8 个字符
重新输入新的 密码：
passwd: 所有的身份验证令牌已经成功更新。
[root@localhost test]#
```

```
[milan@localhost ~]$ pwd
/home/milan          登录进来默认的目录
[milan@localhost ~]$   就是该用户的工作目录
```

3.3 删 除 用户

```
1 // 1. 删除某个用户，但是保留其家目录
2 userdel 用户名
3
4 // 2. 删除用户以及该用户的主目录
5 userdel -r 用户名
6
```

当前登录用户是 milan，进入到 /home 目录下，可以看到各个用户的主目录，尝试删除 milan 用户
发现当前用户正在使用，无法删除，即使切换到 root 权限也无法删除；需要退出当前 milan 用户，并以 root 管理用户登录才可以删除用户

```
[milan@localhost ~]$ cd /home
[milan@localhost home]$ ls
milan ripple test
[milan@localhost home]$ userdel milan
userdel: user milan is currently used by process 5337
[milan@localhost home]$ su - root
密码：
上一次登录：二 10月 10 22:33:32 CST 2023 从 192.158.158.1pts/2 上
[root@localhost ~]# /home/
-bash: /home/: 是一个目录
[root@localhost ~]# /home
-bash: /home: 是一个目录
[root@localhost ~]# cd /home
[root@localhost home]# ls
milan ripple test
[root@localhost home]# userdel milan
userdel: user milan is currently used by process 5329
[root@localhost home]#
```

```
[root@localhost home]# Logout
[milan@localhost home]$ su - root
密码：
[milan@localhost home]$ logout
Connection closed. 退出当前系统

Disconnected from remote host(192.158.158.11).

Type `help' to learn how to use Xshell prompt.
[C:\~]$
```

Connecting to 192.158.158.11:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.



使用root用户登录，删除 milan 用户，但是其主目录保留，当再次以 milan用户去登录时：**SSH服务器拒绝了密码，请再试一次。**代表该用户已经不存在了，无法登录

```
[root@localhost ~]# pwd      当前是root用户登录  
/root  
[root@localhost ~]# cd /home  
[root@localhost home]# ls  
milan ripple test          删该用户，但保留其主目录milan  
[root@localhost home]# userdel milan  
[root@localhost home]# ls  
milan ripple test 该目录是在添加king用户时自定义的用户主目录  
[root@localhost home]# userdel -r tom  
userdel: 用户“tom”不存在  
[root@localhost home]# ls  
milan ripple test          删除king用户，包括其主目录全部删除  
[root@localhost home]# userdel -r king  
[root@localhost home]# ls  
milan ripple  
[root@localhost home]#
```

3.4 查询用户-切换用户-查看当前登录用户

```
1 // 1.查询用户指令，用户不存在时，返回无此用户  
2 id 用户名  
3  
4 // 2.切换用户指令  
5 su - 用户名  
6  
7 // 3.查看当前登录用户(是第一次登录时的用户)  
8 whoami 或者 who am i  
9  
10 注意：  
11 1. 从权限高的用户切换到权限低的用户，不需要输入密码，反之需要  
12 2. 当需要返回到原来的用户时，使用 exit 或者 logout 指令
```

```
[root@localhost ~]# id ripple
uid=1000(ripple) gid=1000(ripple) 组 =1000(ripple)
[root@localhost ~]# id root
uid=0(root) gid=0(root) 组 =0(root)
[root@localhost ~]# cd /home
[root@localhost home]# ls
milan ripple
[root@localhost home]# id milan
id: milan: no such user
[root@localhost home]# su - ripple
上一次登录：二 10月 10 21:03:42 CST 2023:0 上
[ripple@localhost ~]$ su - root
密 码：
上一次登录：一 10月 16 14:04:11 CST 2023从 192.158.158.1pts/0 上
[root@localhost ~]# whoami
root
[root@localhost ~]#
```

3.5 用户组和 etc/下相关文件信息

类似于角色，系统可以对有共性(权限)的多个用户进行统一的管理

1. /etc/passwd :用户的配置文件，记录用户的各种信息

每行含义 用户名：口令：用户标识号：组标识号：注释性描述：主目录：登录Shell

2. /etc/shadow 口令的配置文件

每行含义 登录名：加密口令：最后一次修改时间：最短时间间隔：最大时间间隔：警告时间：不活动时间：失效时间：标志

3. /etc/group 组的配置文件信息

每行含义 组名：口令：组标识号：组内用户列表

```

1 // 新增组
2 groupadd 组名
3
4 // 删除组
5 groupdel 组名
6
7 // 添加用户时可以直接指定组
8 useradd -g 用户组 用户名
9
10 注意：如果新建用户时没有指定组，那么默认会创建一个与用户同名的组并将其放在该组下面
11

```

```

[root@localhost etc]# pwd
/etc
[root@localhost etc]# cat 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
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin

```

4 运行级别-找回root密码-帮助指令

- 运行级别说明，常用级别是 3 和 5，也可以通过 init 指定运行级别

- 0 关机
- 1 单用户
- 2 多用户状态没有网络服务
- 3 多用户状态有网络服务
- 4 系统未使用保留给用户
- 5 图形界面

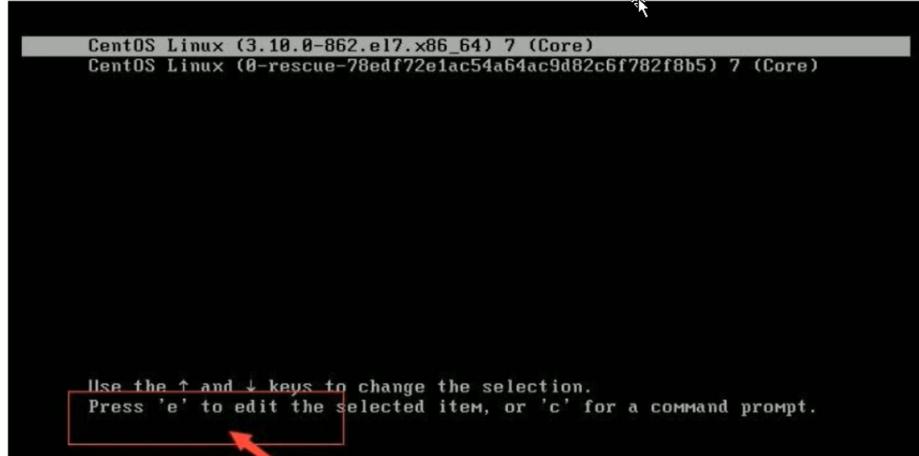
6 系统重启

```
1 // 切换不同的运行级别  
2 init 级别号
```

2. 找回 root 密码

设置运行级别，linux运行后，直接进入到命令行终端

- 首先，启动系统，进入开机界面，在界面中按“e”进入编辑界面。如图



- 进入编辑界面，使用键盘上的上下键把光标往下移动，找到以“Linux16”开头内容所在的行数，在行的最后面输入：init=/bin/sh。如图

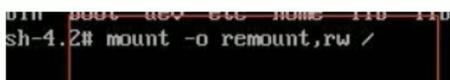
```

insmod xfs
set root='hd0,msdos1'
if [ $feature_platform_search_hint = xy 1]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint=t-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' 6cc07f9f-1
405-4c03-8b9f-9e8d
else
    search --no-floppy --fs-uuid --set=root 6cc07f9f-1405-4c03-8b9f-9e8d
fbc3d84u
fi
linux16 /vmlinuz-3.10.0-862.el7.x86_64 root=/dev/mapper/centos-root ro
crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet LANG=
zh_CN.UTF-8 init=/bin/sh
initrd16 /initramfs-3.10.0-862.el7.x86_64.img

Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
discard edits and return to the menu. Pressing Tab lists
possible completions.

```

3. 接着，输入完后，直接按快捷键：Ctrl+x 进入单用户模式。
 4. 接着，在光标闪烁的位置中输入：mount -o remount,rw /（注意：各个单词间有空格），完成后按键盘的回车键（Enter）。如图



5. 在新的一行最后面输入：passwd，完成后按键盘的回车键（Enter）。输入密码，然后再次确认密码即可(韩顺平提示：密码长度最好 8 位以上,但不是必须的)，密码修改成功后，会显示 passwd.....的样式，说明密码修改成功



6. 接着，在鼠标闪烁的位置中（最后一行中）输入：touch /.autorelabel（注意：touch 与 /后面有一个空格），完成后按键盘的回车键（Enter）
 7. 继续在光标闪烁的位置中，输入：exec /sbin/init（注意：exec 与 /后面有一个空格），完成后按键盘的回车键（Enter），等待系统自动修改密码(韩顺平提示：这个过程时间可能有点长，耐心等待)，完成后，系统会自动重启，新的密码生效了



3. 帮助指令

Shell |

```

1 // 获得帮助信息
2 man [命令或配置文件]
3
4 // 获得shell内置命令的帮助信息
5 help 命令

```

5 文件目录指令

5.1 pwd ls cd

```
1 // 显示当前工作目录的绝对路径
2 pwd
3
4 ls -a // 显示当前目录的所有文件和目录，包括隐藏的
5 ls -l // 以列表的方式显示信息
6 ls -h // 显示详细简单参数，包括文件大小，时间等
7
8 // 查看指定目录下的所有文件和目录信息，a,l,h选项可以组合使用
9 ls -lah /home
10
11 // 切换到自己的家目录
12 cd 或者 cd ~
13 // 切换到当前目录的上一级目录
14 cd ..
```

```
[root@localhost ~]# ls -lah
总用 量 120K
dr-xr-x---. 15 root root 4.0K 10月 16 19:09 .
dr-xr-xr-x. 18 root root 4.0K 10月 5 07:18 ..
-rw-----. 1 root root 2.0K 10月 5 07:18 anaconda-ks.cfg
-rw-----. 1 root root 1.1K 10月 10 23:28 .bash_history
-rw-r--r--. 1 root root 18 12月 29 2013 .bash_logout
-rw-r--r--. 1 root root 176 12月 29 2013 .bash_profile
-rw-r--r--. 1 root root 176 12月 29 2013 .bashrc
-rw-r--r--. 1 root root 37 10月 6 19:39 bobo.txt
drwx-----. 17 root root 4.0K 10月 5 07:56 .cache
drwxr-xr-x. 16 root root 4.0K 10月 6 05:29 .config
-rw-r--r--. 1 root root 100 12月 29 2013 .cshrc
drwx-----. 3 root root 4.0K 10月 5 07:20 .dbus
-rw-----. 1 root root 16 10月 5 07:24 .esd_auth
-rw-r--r--. 1 root root 263 10月 10 20:39 hello.java
-rw-----. 1 root root 2.5K 10月 6 19:22 .ICEauthority
-rw-r--r--. 1 root root 2.0K 10月 5 07:22 initial-setup-ks.cfg
drwx-----. 3 root root 4.0K 10月 5 07:24 .local
drwx-----. 5 root root 4.0K 10月 5 07:29 .mozilla
-rw-r--r--. 1 root root 129 12月 29 2013 .tcshrc
-rw-----. 1 root root 5.5K 10月 16 15:42 .viminfo
```

→ -l 以列表形式显示
-a 显示所有文件和目录
-h 显示文件大小，时间等参数
..

```
[root@localhost ~]# ls -lah /home/*.txt  
-rw-r--r--. 1 root root 123 10月 16 20:08 /home/hello.txt  
-rw-r--r--. 1 root root 563 10月 16 16:24 /home/info.txt  
[root@localhost ~]# 显示该目录下所有 .txt类型的文件
```

5.2 mkdir rmdir rm -rf

```
1 // 创建目录  
2 mkdir 目录名  
3  
4 // 创建多级目录  
5 mkdir -p 目录  
6  
7 // 删除空目录, 如果目录下有内容是无法删除的  
8 rmdir 目录  
9  
10 // 删除目录或文件  
11 rm -rf 目录  
12  
13
```

```
[root@localhost /]# cd /home  
[root@localhost home]# ls  
milan ripple  
[root@localhost home]# mkdir -p test/test2  
[root@localhost home]# ls  
milan ripple test test2  
[root@localhost home]# mkdir test3 test4  
[root@localhost home]# ls  
milan ripple test test2 test3 test4  
[root@localhost home]# mkdir test5/test6  
mkdir: 无法创建目录 "test5/test6": 没有那个文件或目录  
[root@localhost home]# mkdir -p test5/test6  
[root@localhost home]# ls  
milan ripple test test2 test3 test4 test5  
[root@localhost home]# cd test5  
[root@localhost test5]# ls  
test6
```

5.3 touch cp rm

```

1 // 1. 创建空文件
2 touch 文件名称
3
4 // 2. 拷贝文件到指定目录
5 cp [选项] 源目录 目标目录
6
7 常用选项: -r 递归复制整个文件夹, 包括子目录的内容
8 使用细节: 强制覆盖不提示: \cp
9
10 举例: 拷贝home下的整个test5目录到 /opt下, 并强制覆盖
11 \cp -r /home/test5/ /opt/
12
13 // 3. 删除文件或目录
14 rm [选项] 文件或目录
15
16 常用选项:
17 -r 递归删除整个文件夹
18 -f 强制删除不提示
19
20 举例: 删除 /home/test5 整个目录, 并强制删除
21 rm -rf /home/test5
22

```

```

[root@localhost ~]# cd /home
[root@localhost home]# ls
milan ripple test test2 test3 test4 test5
[root@localhost home]# touch hello.txt
[root@localhost home]# ls
hello.txt milan ripple test test2 test3 test4 test5
[root@localhost home]# cp /home/hello.txt /home/test
[root@localhost home]# ls -la test
总用量 8                                         → 列出test目录下的所有信息
drwxr-xr-x. 2 root root 4096 10月 16 15:23 .
drwxr-xr-x. 9 root root 4096 10月 16 15:22 ..
-rw-r--r--. 1 root root 0 10月 16 15:23 hello.txt
[root@localhost home]# cp hello.txt test2
[root@localhost home]# ls -la test2 在同级目录下可以直接拷贝文件到某个目录
总用量 8
drwxr-xr-x. 2 root root 4096 10月 16 15:23 .
drwxr-xr-x. 9 root root 4096 10月 16 15:22 ..
-rw-r--r--. 1 root root 0 10月 16 15:23 hello.txt

```

5.4 mv cat less echo head tail

```
1 // 1. 移动文件与目录或重命名
2 mv 老文件(目录)名 新文件(目录)名 // 重命名
3 mv /文件目录 /目标目录 //移动文件或目录
4
5
6 // 2. 查看文件内容
7 cat -n 文件 // -n 显示行号
8 使用细节: cat指令只能浏览文件, 为了方便, 一般会带上管道命令 | more
9
10
11 // 3.less指令用来分屏查看文件内容, 它的功能与more指令类似, 支持各种显示终端
12 less 文件
13
14 less指令操作 功能说明
15 空白键space 向下翻动一页
16 /字符串 向下搜寻该字符串, n:向下查找, N: 向上查找
17 ?字符串 向上....
18 q 离开less这个程序指令
19
20
21 // 4.echo指令输出内容到控制台
22 echo [选项] 输出内容
23
24
25 // 5.head指令用于显示文件的开头部分内容, 默认显示文件前10行
26 head 文件
27 // 比如查看文件前20行内容
28 head -n 20 文件
29
30
31 // 6. 用于输出文件中尾部的内容
32 tail -n 5 文件
33
34 // 7. 实时监控某个文件的更新变化内容情况
35 tail -f 文件
36
37
```

```
[root@localhost home]# cat -n /etc/profile | more
 1 # /etc/profile
 2
 3 # System wide environment and startup programs, for login se
 4 # Functions and aliases go in /etc/bashrc
 5
 6 # It's NOT a good idea to change this file unless you know w
 7 # are doing. It's much better to create a custom.sh shell sc
 8 # /etc/profile.d/ to make custom changes to your environment
 9 # will prevent the need for merging in future updates.
10
11 pathmunge () {
12     case ":${PATH}:" in
13         *:"$1":*)
14             ;;

```

```
[root@localhost home]# echo $HOSTNAME
localhost.localdomain
[root@localhost home]# echo $PATH
/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
[root@localhost home]# cat hostname
```

```
[root@localhost home]# head -n 20 /etc/profile
# /etc/profile                               需要带上选项 -n 才可以
# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

pathmunge () {
    case ":${PATH}:" in
        *:"$1":*)
            ;;
        *)
            if [ "$2" = "after" ] ; then
                PATH=$PATH:$1
            else
                PATH=$1:$PATH
            fi

```

5.5 > 和 >> 指令

> 输出重定向(写入覆盖) 和 >> 追加

```

1 // 列表的内容写入到文件中，覆盖写
2 ls -l > 文件
3
4 // 列表的内容追加到文件的末尾
5 ls -la >> 文件
6
7 // 把文件1的内容覆盖到文件2
8 cat 文件1 > 文件2
9
10 使用说明：通过 > 与 >> 可以把前面任何指令所显示或输出的所有内容
11      以 覆盖或追加 的方式写入到指定的某个文件中。
12

```

```

[root@localhost home]# ls
hello.txt milan ripple test test2 test3 test4
[root@localhost home]# ls -la >> info.txt
[root@localhost home]# ls -la 总用量 40          把前面指令显示的内容追加写入到info.txt文件中
drwxr-xr-x.  8 root    root   4096 10月 16 16:24 .
dr-xr-xr-x. 18 root    root   4096 10月  5 07:18 ..
-rw-r--r--.  1 root    root    92 10月 16 15:42 hello.txt
-rw-r--r--.  1 root    root   563 10月 16 16:24 info.txt
drwx-----. 5 1001   1001 4096 10月 10 23:13 milan
drwx-----. 15 ripple  ripple 4096 10月 10 23:28 ripple
drwxr-xr-x.  2 root    root   4096 10月 16 15:23 test
drwxr-xr-x.  2 root    root   4096 10月 16 15:23 test2
drwxr-xr-x.  2 root    root   4096 10月 16 15:09 test3
drwxr-xr-x.  2 root    root   4096 10月 16 15:09 test4
[root@localhost home]# cat info.txt
总用量 36
drwxr-xr-x.  8 root    root   4096 10月 16 16:24 .
dr-xr-xr-x. 18 root    root   4096 10月  5 07:18 ..
-rw-r--r--.  1 root    root    92 10月 16 15:42 hello.txt
-rw-r--r--.  1 root    root     0 10月 16 16:24 info.txt
drwx-----. 5 1001   1001 4096 10月 10 23:13 milan
drwx-----. 15 ripple  ripple 4096 10月 10 23:28 ripple
drwxr-xr-x.  2 root    root   4096 10月 16 15:23 test
drwxr-xr-x.  2 root    root   4096 10月 16 15:23 test2
drwxr-xr-x.  2 root    root   4096 10月 16 15:09 test3
drwxr-xr-x.  2 root    root   4096 10月 16 15:09 test4

```

```
[root@localhost home]# cal
```

查看当前日历

十月 2023

日	一	二	三	四	五	六
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				

把日历内容追加写入到某个文件

```
[root@localhost home]# cal >> /home/mycal
```

```
[root@localhost home]# cat mycal
```

十月 2023

日	一	二	三	四	五	六
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				

```
[root@localhost home]# cat /etc/profile > /home/myprofile
```

把前面指令查看到的内容覆盖写入到指定文件中
总用 量 48

```
drwxr-xr-x. 8 root root 4096 10月 16 16:28 .
dr-xr-xr-x. 18 root root 4096 10月 5 07:18 ..
-rw-r--r--. 1 root root 92 10月 16 15:42 hello.txt
-rw-r--r--. 1 root root 563 10月 16 16:24 info.txt
drwx-----. 5 1001 1001 4096 10月 10 23:13 milan
-rw-r--r--. 1 root root 145 10月 16 16:27 mycal
-rw-r--r--. 1 root root 1819 10月 16 16:28 myprofile
drwx-----. 15 ripple ripple 4096 10月 10 23:28 ripple
drwxr-xr-x. 2 root root 4096 10月 16 15:23 test
drwxr-xr-x. 2 root root 4096 10月 16 15:23 test2
drwxr-xr-x. 2 root root 4096 10月 16 15:09 test3
drwxr-xr-x. 2 root root 4096 10月 16 15:09 test4
[root@localhost home]# cat -n myprofile
 1 # /etc/profile
 2
 3 # System wide environment and startup programs, for login setup
 4 # Functions and aliases go in /etc/bashrc
 5
 6 # It's NOT a good idea to change this file unless you know what you
 7 # are doing. It's much better to create a custom.sh shell script in
 8 # /etc/profile.d/ to make custom changes to your environment, as this
 9 # will prevent the need for merging in future updates.
10
11 pathmunge () {
```

5.6 ln history date cal

ln软链接也称为符号链接，类似于windows里的快捷键方式，主要存放了链接其他文件的路径

```
1 // 1. 给原文件创建一个软链接
2 ln -s 原文件或目录 软链接名
3
4 案例1：在 /home 目录下创建一个软链接 myroot，链接到 /root目录
5 ln -s /root/ myroot
6
7 删除软连接
8 rm /home/myroot
9
10
11 // 2. 查看已经执行过的历史命令，也可以执行历史命令
12 history //显示所有的历史命令
13 history 10 //显示最近使用过的10个命令
14 !历史编号          //执行某个历史指令
15
16
17 // 3. date指令，显示当前日期
18 date //显示当前时间
19 date +%Y //显示年份
20 date +%m //显示月份
21 date +%d //显示当前是那一天
22
23 // 4. 设置日期
24 date -s "2022-11-11 20:05:19" //设置系统当前时间
25
26 // 5. 查看日历指令
27 cal //显示当前日历
28 cal 2020 //显示2020年日历
29
```

Shell |

```
[root@localhost ~]# cd /home
[root@localhost home]# ln -s /root/ myroot
[root@localhost home]# ll
总用 量 40
-rw-r--r--. 1 root root 92 10月 16 15:42 hello.txt
-rw-r--r--. 1 root root 563 10月 16 16:24 info.txt
drwx-----. 5 1001 1001 4096 10月 10 23:13 milan
-rw-r--r--. 1 root root 145 10月 16 16:27 mycal
-rw-r--r--. 1 root root 1819 10月 16 16:28 myprofile
lrwxrwxrwx. 1 root root 6 10月 16 19:20 myroot -> /root/
drwx-----. 15 ripple ripple 4096 10月 10 23:28 ripple
drwxr-xr-x. 2 root root 4096 10月 16 15:23 test
drwxr-xr-x. 2 root root 4096 10月 16 15:23 test2
drwxr-xr-x. 2 root root 4096 10月 16 15:09 test3
drwxr-xr-x. 2 root root 4096 10月 16 15:09 test4
```

5.7 查找指令find which grep

```

1 // 1. find指令将从指定目录向下递归地遍历其各个子目录，将满足条件的文件或者目录显示在终端
2 -> find [搜索范围] [选项]
3
4 选项说明：
5   -name 按照指定的文件名查找
6   -user 查找属于指定用户的所有文件
7   -size 按照指定的文件大小查找
8 案例1：按照文件名查找 /home 目录下的hello.txt文件
9 find /home -name hello.txt
10 案例2：按照用户拥有者查找 /opt目录下，用户名为 root的文件
11 find /opt -user root
12 案例3：查找整个linux系统下大于200M的文件 (+n大于 -n小于 n等于，单位有：k M G)
13 find / -size +200M
14
15
16 //2. 查看某个指令在哪个目录下，比如 ls 指令
17 which ls
18
19
20 //3. grep过滤查找，管道符 |，表示将前一个命令的处理结果输出传递给后面的命令处理
21 -> grep [选项] 查找内容 源文件
22
23 选项说明 -ni 可以组合使用：
24   -n 显示匹配以及行号
25   -i 忽略字母大小写
26 案例：在hello.txt文件中，查找 "yes"所在行，并显示行号
27 写法1：cat /home/hello.txt | grep -n "yes"
28 写法2：grep -n "yes" /home/hello.txt
29

```

```

[root@localhost ~]# cat /home/hello.txt | grep -n "yes"
11:yes
14:yesyes
15:yes
16:yes1738271
[root@localhost ~]#

```

管道符，把前面指令的内容
交给后面指令处理

5.8 压缩与解压缩 gzip gunzip

gzip 用于压缩文件，gunzip 用于解压

```

1 // 压缩文件，只能将文件压缩为 *.gz 类型
2 gzip 文件
3
4 // 解压缩
5 gunzip 文件.gz
6
7 案例1：将 /home 下的hello.txt 文件进行压缩
8 gzip /home/hello.txt
9
10 案例2：将 /home 下的hello.txt.gz 文件进行解压缩
11 gunzip /home/hello.txt.gz

```

```

[root@localhost ~]# gzip /home/hello.txt    压缩
[root@localhost ~]# ls -lah /home    显示 /home 目录下的所有内容
总用量 48K
drwxr-xr-x.  8 root    root    4.0K 10月 16 20:19 .
dr-xr-xr-x. 18 root    root    4.0K 10月  5 07:18 ..
-rw-r--r--.  1 root    root    142 10月 16 20:08 hello.txt.gz
-rw-r--r--.  1 root    root    563 10月 16 16:24 info.txt
drwx-----.  5 1001   1001 4.0K 10月 10 23:13 milan
-rw-r--r--.  1 root    root    145 10月 16 16:27 mycal
-rw-r--r--.  1 root    root    1.8K 10月 16 16:28 myprofile
lrwxrwxrwx.  1 root    root     6 10月 16 19:20 myroot -> /root/
drwx-----. 15 ripple  ripple  4.0K 10月 10 23:28 ripple
drwxr-xr-x.  2 root    root    4.0K 10月 16 15:23 test
drwxr-xr-x.  2 root    root    4.0K 10月 16 15:23 test2
drwxr-xr-x.  2 root    root    4.0K 10月 16 15:09 test3
drwxr-xr-x.  2 root    root    4.0K 10月 16 15:09 test4
[root@localhost ~]# gunzip /home/hello.txt.gz    解压缩
[root@localhost ~]# ls -lah /home
总用量 48K
drwxr-xr-x.  8 root    root    4.0K 10月 16 20:20 .
dr-xr-xr-x. 18 root    root    4.0K 10月  5 07:18 ..
-rw-r--r--.  1 root    root    123 10月 16 20:08 hello.txt
-rw-r--r--.  1 root    root    563 10月 16 16:24 info.txt
drwx-----.  5 1001   1001 4.0K 10月 10 23:13 milan

```

5.9 压缩与解压缩 zip unzip

zip 用于压缩文件，unzip 用于解压缩，这个在项目打包发布中常用

```
1 // 压缩文件或目录的命令
2 zip -r 压缩名.zip 要压缩的文件或目录
3
4 // 解压缩
5 unzip -d 指定解压的目录 要解压的文件或目录.zip
6
7 zip常用选项说明:
8   -r 递归压缩, 即压缩目录
9 unzip选项说明:
10  -d 指定解压缩后文件的存放目录
11
12 案例1: 把 /home下的所有文件进行压缩成 myhome.zip
13 zip -r myhome.zip /home/
14 案例2: 把 myhome.zip 解压到指定 /op/tmp 目录下
15 unzip -d /opt/tmp /home/myhome.zip
16
```

5.10 压缩与解压缩 tar

tar指令是打包指令，最后打包后的文件是 .tar.gz 类型文件

```

1 // 打包压缩目录
2 tar [选项] 压缩名.tar.gz 要压缩的目录 -C 指定解压的目录
3
4 选项说明:
5   -c 产生 .tar 打包文件
6   -v 显示详细信息
7   -f 指定压缩后的文件名
8   -z 打包同时压缩
9   -x 解包 .tar 文件
10
11 案例1: 压缩多个文件, 将 /home/pig.txt 和 /home/cat.txt 压缩成 pc.tar.gz
12 tar -zcvf pc.tar.gz /home/pig.txt /home/cat.txt
13
14 案例2: 将 /home 的文件夹压缩成 myhome.tar.gz
15 tar -zcvf myhome.tar.gz /home/
16
17 案例3: 将 pc.tar.gz 解压到当前目录, 切换到 /opt/
18 tar -zxvf pc.tar.gz
19
20 案例4: 将 myhome.tar.gz 解压到 /opt/tmp2 目录下
21 tar -zxvf /home/myhome.tar.gz -C /opt/tmp2
22

```

6 组和rwx权限 P40–P62 集未看

7 网络配置–进程–服务管理

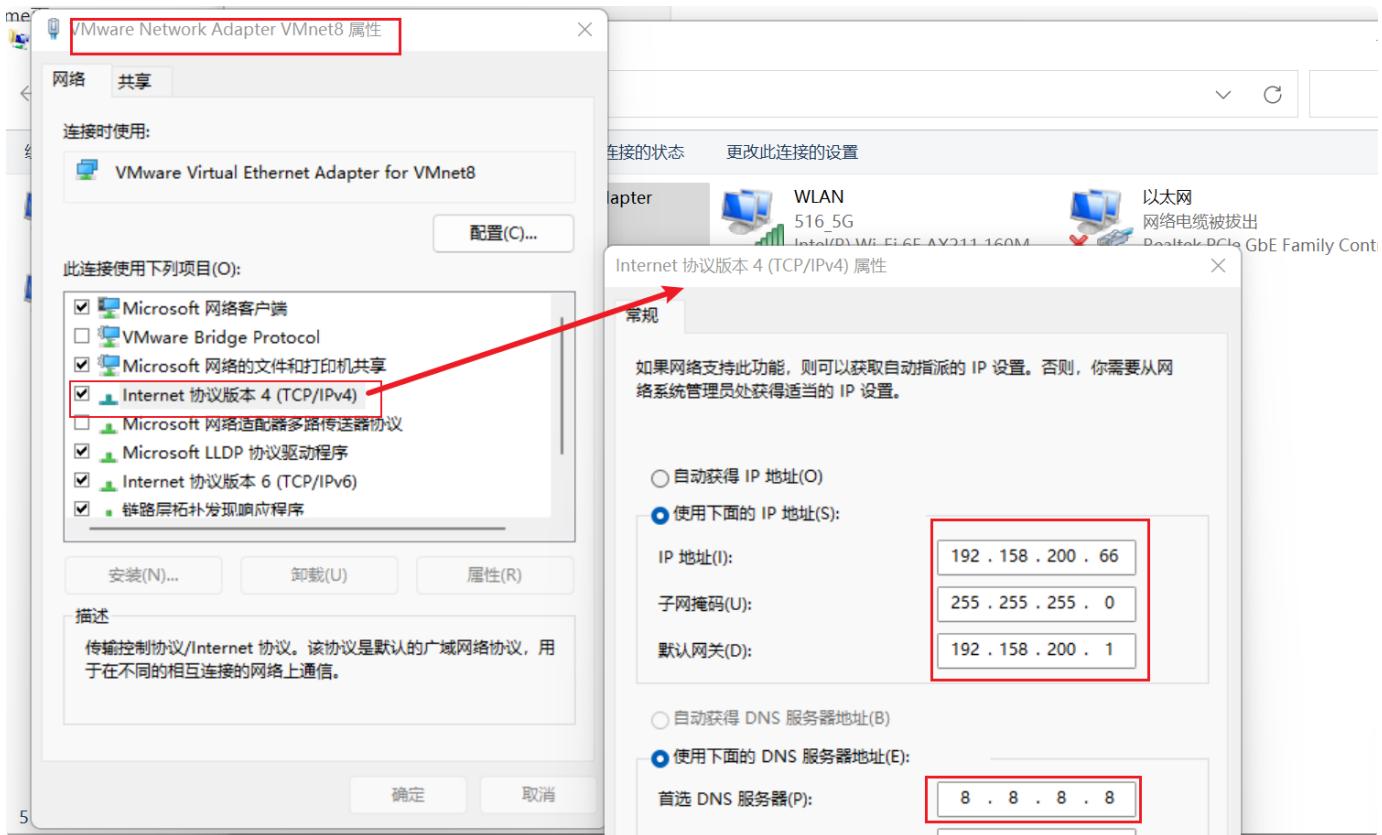
7.1 修改网络配置文件实例

```

1 // 1. 打开配置文件
2 vim sysconfig/network-scripts/ifcfg-ens33
3
4 // 2. 重启网络服务生效 或者 reboot 重启虚拟机
5 service network restart

```

1. 手动设置windows宿主机的 VMwarenet8的虚拟网卡配置信息



2. 在cmd下执行 ipconfig 查看本机网卡信息

```
以太网适配器 VMware Network Adapter VMnet1:
```

```
连接特定的 DNS 后缀 . . . . . :  
本地链接 IPv6 地址. . . . . : fe80::3bd7:be86:252:b2b8%10  
IPv4 地址 . . . . . : 192.168.203.1  
子网掩码 . . . . . : 255.255.255.0  
默认网关. . . . . :
```

```
以太网适配器 VMware Network Adapter VMnet8:
```

```
连接特定的 DNS 后缀 . . . . . :  
本地链接 IPv6 地址. . . . . : fe80::365e:81a9:3711:45d8%4  
IPv4 地址 . . . . . : 192.158.200.66 可以手动指定宿主机的 ip和网关  
子网掩码 . . . . . : 255.255.255.0  
默认网关. . . . . : 192.158.200.1
```

```
无线局域网适配器 WLAN:
```

```
连接特定的 DNS 后缀 . . . . . :  
IPv6 地址 . . . . . : fdd1:5503:61ce:4d6c:813e:f08e:169c:e78b  
临时 IPv6 地址. . . . . : fdd1:5503:61ce:4d6c:dda3:a683:e521:5fd2  
本地链接 IPv6 地址. . . . . : fe80::b3a2:e4ec:c509:dc7f%7  
IPv4 地址 . . . . . : 192.168.60.165  
子网掩码 . . . . . : 255.255.255.0  
默认网关. . . . . : fe80::222:93ff:fe54:6441%7  
fe80::222:93ff:fe54:6440%7  
192.168.60.1
```

```
以太网适配器 以太网:
```

```
媒体状态 . . . . . : 媒体已断开连接  
连接特定的 DNS 后缀 . . . . . :
```

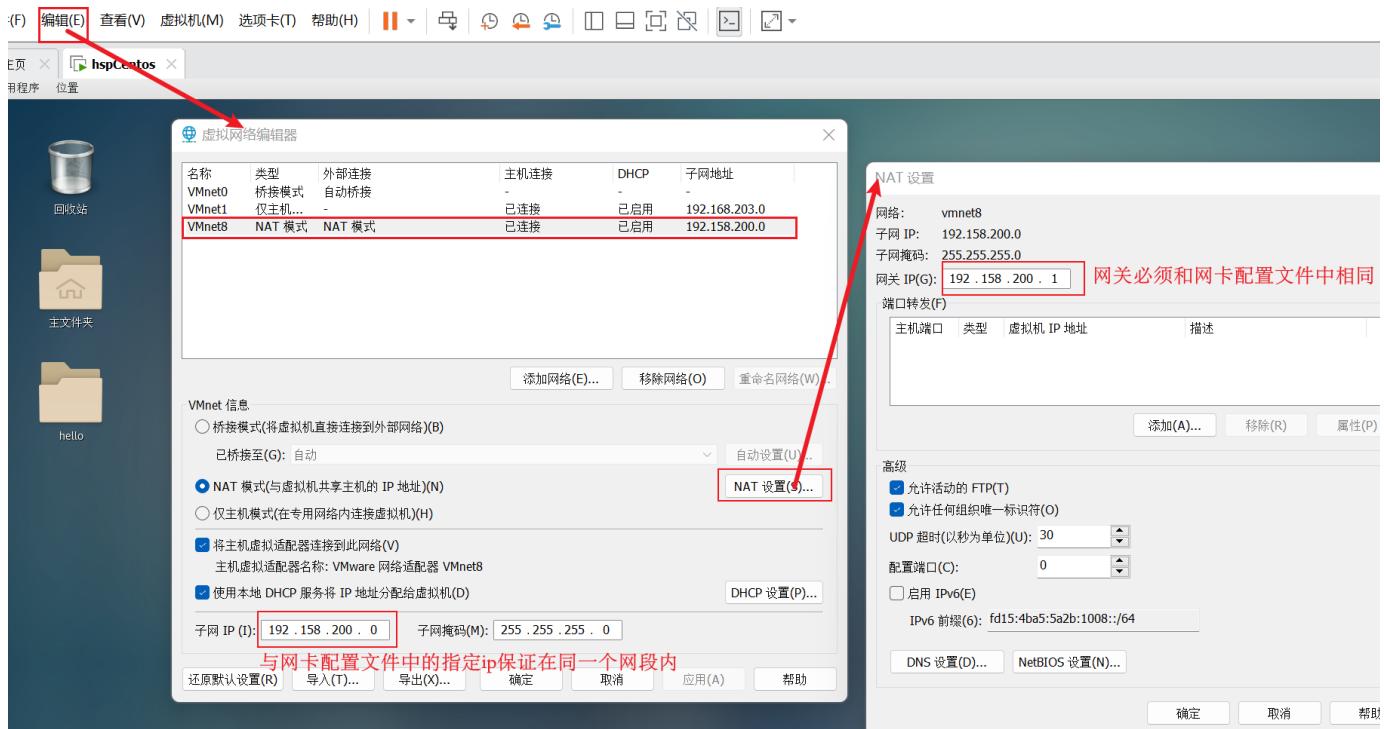
```
C:\Users\花海>■
```

- 在Linux虚拟机中的网络配置文件中修改如下信息， vim sysconfig/network-scripts/ifcfg-ens33

```

● 1 192.158.200.101 (hspCent... × + 
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens33
UUID=69954612-e669-4f29-ba43-3075da555f14
DEVICE=ens33
ONBOOT=yes
HWADDR=00:0C:29:8D:E1:9B
IPADDR=192.158.200.101
PREFIX=24
GATEWAY=192.158.200.1
DNS1=8.8.8.8
~ 1. 网卡配置文件中网关必须要和虚拟网络编辑器中的网关相同
~ 2. 并且两者的ip地址要在同一个网段内，即 192.158.200.xxx
~
```

4. 修改虚拟网络编辑器的配置信息，最后点击应用，确定



5. 最后在虚拟机终端命令行执行 service network restart 刷新网络服务生效 或者 reboot 重启虚拟

7.2 设置主机名和hosts映射

hosts: 用于记录 ip 和 主机名的映射关系

DNS: 域名系统，是作为域名和 IP地址相互映射的一个分布式数据库

```
▼ Shell |  
1 // 设置虚拟机主机名  
2 vim /etc/hostname  
3  
4 // 修改虚拟机主机配置  
5 vim /etc/hosts  
6  
7  
8 // 案例：设置hosts映射，如何通过主机名找到（比如 ping）某个Linux虚拟机系统？  
9 1. windows 中 C:\Windows\System32\drivers\etc 目录下找到 hosts 文件，添加如下：  
10 192.158.200.101 hsp  
11  
12 2. Linux中在 /etc/hosts 文件中添加如下：  
13 192.158.200.66 LAPTOP-0QA935GA
```

```
[root@hsp ~]# cat /etc/hostname  
hsp  
[root@hsp ~]# vim /etc/hosts  
[root@hsp ~]# cat /etc/hosts  
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  
192.158.200.66 LAPTOP-0QA935GA windows宿主机的VMware网卡的ip 和 电脑设备名  
[root@hsp ~]#
```

在宿主机中ping虚拟机

C:\Users\花海>ping hsp 设置主机之间的映射后，可以直接ping主机名

```
正在 Ping hsp [192.158.200.101] 具有 32 字节的数据:  
来自 192.158.200.101 的回复: 字节=32 时间<1ms TTL=64  
来自 192.158.200.101 的回复: 字节=32 时间<1ms TTL=64  
来自 192.158.200.101 的回复: 字节=32 时间<1ms TTL=64  
来自 192.158.200.101 的回复: 字节=32 时间<1ms TTL=64
```

```
192.158.200.101 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 = 0ms, 最长 = 0ms, 平均 = 0ms
```

C:\Users\花海>

在Linux虚拟机中ping宿主机

```
[root@hsp ~]# ping LAPTOP-0QA935GA  
PING LAPTOP-0QA935GA (192.158.200.66) 56(84) bytes of data.  
64 bytes from LAPTOP-0QA935GA (192.158.200.66): icmp_seq=1 ttl=128 time=0.378 ms  
64 bytes from LAPTOP-0QA935GA (192.158.200.66): icmp_seq=2 ttl=128 time=0.662 ms  
64 bytes from LAPTOP-0QA935GA (192.158.200.66): icmp_seq=3 ttl=128 time=0.556 ms  
64 bytes from LAPTOP-0QA935GA (192.158.200.66): icmp_seq=4 ttl=128 time=0.753 ms  
^C  
--- LAPTOP-0QA935GA ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3002ms  
rtt min/avg/max/mdev = 0.378/0.587/0.753/0.140 ms  
[root@hsp ~]#
```

7.3 进程-ps-kill-pstree指令

在Linux中，每个进程都分配一个ID号，即进程号pid。每个进程都可能以2种方式存在，即前台与后台，所谓前台是指用户目前在屏幕上可以进行操作的，后台则是实际在操作，但是屏幕上看不到，一般的系统服务都是以后台进程的方式存在，直到关机才结束。

```
1 // 显示当前终端的所有进程信息
2 ps -a
3
4 // 以用户的格式显示进程信息
5 ps -u
6
7 // 显示后台进程运行参数信息
8 ps -x
9
10 // 1. 进程指令选项可以组合使用，显示更多参数
11 ps -aux
12
13 // 以全格式显示当前所有的进程信息
14 ps -ef
15
16 // 2. 终止进程
17 kill [选项] 进程号
18 killall 进程名称
19
20 选项说明：
21     -9 : 表示强迫进程立即停止
22
23 案例1：踢掉某个非法登录的用户
24 kill 对应的进程号
25
26 案例2：终止远程登录服务sshd，在适当的时候再次重启sshd服务
27 kill 进程号
28 执行 /bin/systemctl start sshd.service 重启
29
30 案例3：终止多个 gedit
31 killall 进程名称
32
33 案例4：强制杀死一个终端
34 kill -9 bash对应的进程号
35
36
37 // 3. 查看进程树
38 pstree -p // 显示进程的PID
39 pstree -u // 显示进程的所属用户
40
41 案例1：以树状形式显示进程的pid
42 案例2：以树状形式显示用户id
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	终端信息		执行的开始时间	占用的CPU时间	该进程的执行指令
							STAT	START			
root	1	0.0	0.3	193952	7120	?	Ss	21:07	0:01	/usr/lib/systemd/sys	
root	2	0.0	0.0	0	0	?	S	21:07	0:00	[kthreadd]	
root	4	0.0	0.0	0	0	?	运行状态<	21:07	0:00	[kworker/0:0H]	
root	6	0.0	0.0	0	0	?	s 表示睡眠	21:07	0:00	[ksoftirqd/0]	
root	7	0.0	0.0	0	0	?	r 表示正在运行	21:07	0:00	[migration/0]	
root	8	0.0	0.0	0	0	?	S	21:07	0:00	[rcu_bh]	
root	9	0.0	0.0	0	0	?	S	21:07	0:00	[rcu_sched]	
root	10	0.0	0.0	0	0	?	S<	21:07	0:00	[lru-add-drain]	
root	11	0.0	0.0	0	0	?	S	21:07	0:00	[watchdog/0]	
root	12	0.0	0.0	0	0	?	S	21:07	0:00	[watchdog/1]	
root	13	0.0	0.0	0	0	?	S	21:07	0:00	[migration/1]	
root	14	0.0	0.0	0	0	?	S	21:07	0:00	[ksoftirqd/1]	
root	15	0.0	0.0	0	0	?	S	21:07	0:00	[kworker/1:0]	
root	16	0.0	0.0	0	0	?	S<	21:07	0:00	[kworker/1:0H]	

```
[root@hsp ~]# ps -aux | grep sshd 配合管道符指令过滤
root    1171  0.0  0.2 112900  4308 ?          Ss   21:08   0:00 /usr/sbin/sshd -D
root    2149  0.0  0.3 160988  5632 ?          Ss   21:13   0:00 sshd: root@pts/0
root    2625  0.0  0.0 112828   976 pts/0       S+   21:56   0:00 grep --color=auto sshd
[root@hsp ~]#
```

杀死父进程后，其子进程也会被杀死

UID	PID	PPID	C	STIME	TTY	TIME			CMD
						进程ID	父进程ID	C: 数值越小，执行优先级会越高	
root	1	0	0	21:07	?			00:00:01	/usr/lib/systemd/systemd --switched-r
root	2	0	0	21:07	?			00:00:00	[kthreadd]
root	4	2	0	21:07	?			00:00:00	[kworker/0:0H]
root	6	2	0	21:07	?			00:00:00	[ksoftirqd/0]
root	7	2	0	21:07	?			00:00:00	[migration/0]
root	8	2	0	21:07	?			00:00:00	[rcu_bh]
root	9	2	0	21:07	?			00:00:00	[rcu_sched]
root	10	2	0	21:07	?			00:00:00	[lru-add-drain]
root	11	2	0	21:07	?			00:00:00	[watchdog/0]
root	12	2	0	21:07	?			00:00:00	[watchdog/1]
root	13	2	0	21:07	?			00:00:00	[migration/1]
root	14	2	0	21:07	?			00:00:00	[ksoftirqd/1]

案例2：终止远程登录服务sshd，在适当的时候再次重启sshd服务

```
[root@hsp ~]# ps -aux | grep sshd 显示sshd相关的进程信息
root 1171 0.0 0.2 112900 4308 ? Ss 21:08 0:00 /usr/sbin/sshd -D
root 2149 0.0 0.3 160988 5632 ? Ss 21:13 0:00 sshd: root@pts/0
root 2835 0.0 0.0 112828 976 pts/0 S+ 22:15 0:00 grep --color=auto sshd
[root@hsp ~]# kill 1171 杀死远程登录服务进程
[root@hsp ~]# ps -aux | grep sshd
root 2149 0.0 0.3 160988 5632 ? Ss 21:13 0:00 sshd: root@pts/0
root 2872 0.0 0.0 112824 976 pts/0 S+ 22:19 0:00 grep --color=auto sshd
[root@hsp ~]# service sshd start
Redirecting to /bin/systemctl start sshd.service
[root@hsp ~]# /bin/systemctl start sshd.service 重启sshd远程登录服务
[root@hsp ~]# ps -aux | grep sshd
root 2149 0.0 0.3 160988 5632 ? Ss 21:13 0:00 sshd: root@pts/0
root 2903 0.0 0.2 112900 4308 ? Ss 22:20 0:00 /usr/sbin/sshd -D
root 2926 0.0 0.0 112824 976 pts/0 S+ 22:20 0:00 grep --color=auto sshd
[root@hsp ~]#
```

以树状形式显示进程信息

```
—xug-permission— z' [xug-permission-]
[root@hsp ~]# pstree -pu
systemd(1)─ ModemManager(750) └─ {ModemManager}(790)
              └─ {ModemManager}(820)
      └─ NetworkManager(859) └─ {NetworkManager}(864)
              └─ {NetworkManager}(868)
      └─ VGAuthService(745)
      └─ abrt-watch-log(779)
      └─ abrt-watch-log(781)
      └─ abrtd(778)
      └─ accounts-daemon(748) └─ {accounts-daemon}(755)
              └─ {accounts-daemon}(821)
      └─ alsactl(776)
      └─ at-spi-bus-laun(1691,gdm) └─ dbus-daemon(1696) └─ {dbus-daemon}(1697)
              └─ {at-spi-bus-laun}(1692)
              └─ {at-spi-bus-laun}(1693)
              └─ {at-spi-bus-laun}(1695)
      └─ at-spi2-registr(1699,gdm) └─ {at-spi2-registr}(1701)
              └─ {at-spi2-registr}(1702)
      └─ atd(1192)
      └─ auditd(714) └─ audispd(716) └─ sedispatch(718)
              └─ sedisndl(719)
```

7.4 服务管理–systemctl指令

服务本质就是进程，但是是运行在后台的，通常都会监听某个端口，等待其它程序的请求，比如mysql、sshd、防火墙等，又称为守护进程

```

1 // 服务管理指令,Centos7.0后不再使用service,
2 // service服务指令管理信息在 /etc/init.d/ 下查看
3 // systemctl指令管理的服务在 /usr/lib/systemd/system 查看
4
5 // 查看服务
6 ls -l /usr/lib/systemd/system | grep 服务名
7
8 //管理服务
9 - systemctl [start|stop|restart|status] 服务名
10
11 // 以窗口形式查看所有服务, tab键切换选项, 空格键选择自启
12 setup
13
14 // 查看服务开启启动状态, grep可以过滤
15 systemctl list-unit-files | grep 服务名
16
17 // 设置服务开机启动
18 systemctl enable 服务名
19
20 // 关闭服务开机启动
21 systemctl disable 服务名
22
23 // 查看某个服务是否为开机自启
24 systemctl is-enabled 服务名
25
26 //查看网络端口情况
27 netstat -anp | more
28
29
30 案例1: 查看当前防火墙服务的状态, 关闭或重启防火墙
31 systemctl status firewalld
32 systemctl stop firewalld
33 systemctl restart firewalld

```

```

[root@hsp ~]# ll /etc/init.d/
总用量 40
-rw-r--r--. 1 root root 18281 5月 22 2020 functions
-rwxr-xr-x. 1 root root 4569 5月 22 2020 netconsole
-rwxr-xr-x. 1 root root 7928 5月 22 2020 network
-rw-r--r--. 1 root root 1160 10月 2 2020 README
[root@hsp ~]#

```

哪些服务应该自动启动？

```
[ ] saslauthd.service  
[*] smartd.service  
[ ] speech-dispatcherd.service  
[*] sshd.service * 开机自动启动  
[ ] sshd.socket  
[ ] sssd-autofs.socket  
[ ] sssd-nss.socket  
[ ] sssd-pac.socket  
[ ] sssd-pam-priv.socket  
[ ] sssd-pam.socket  
[ ] sssd-secrets.socket
```

确定

取消

```
[root@hsp ~]# ls -l /usr/lib/systemd/system | grep sshd  
-rw-r--r--. 1 root root 574 10月 2 2020 anaconda-sshd.service  
-rw-r--r--. 1 root root 313 8月 9 2019 sshd-keygen.service  
-rw-r--r--. 1 root root 373 8月 9 2019 sshd.service  
-rw-r--r--. 1 root root 260 8月 9 2019 sshd@.service  
-rw-r--r--. 1 root root 181 8月 9 2019 sshd.socket  
[root@hsp ~]#
```

```
[root@hsp ~]# systemctl status firewalld    查看防火墙的服务状态  
● firewalld.service - firewalld - dynamic firewall daemon  
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor  
  Active: active (running) since 五 2023-10-27 18:17:07 CST; 27min ago  
    Docs: man:firewalld(1)  
   Main PID: 837 (firewalld)  
     Tasks: 2  
    CGroup: /system.slice/firewalld.service  
           └─837 /usr/bin/python2 -Es /usr/sbin/firewalld --nofork --nopid  
  
10月 27 18:17:06 hsp systemd[1]: Starting firewalld - dynamic firewall daemon  
10月 27 18:17:07 hsp systemd[1]: Started firewalld - dynamic firewall daemon.  
10月 27 18:17:07 hsp firewalld[837]: WARNING: AllowZoneDrifting is enabled.  
Hint: Some lines were ellipsized, use -l to show in full.  
[root@hsp ~]#
```

查看网络端口的情况

查看网络端口状态						
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	805/rpcbind
tcp	0	0	192.168.122.1:53	0.0.0.0:*	LISTEN	1693/dnsmasq
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	1206/sshd
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN	1204/cupsd
tcp	0	0	127.0.0.1:25	0.0.0.0:*	LISTEN	1478/master
tcp	0	0	127.0.0.1:6010	0.0.0.0:*	LISTEN	2098/sshd: root@pts
tcp	0	0	192.158.200.101:22	192.158.200.66:52694	ESTABLISHED	2098/sshd: root@pts
tcp6	0	0	:::111	:::*	LISTEN	805/rpcbind
tcp6	0	0	:::22	:::*	LISTEN	1206/sshd
tcp6	0	0	:::1:631	:::*	LISTEN	1204/cupsd
tcp6	0	0	:::1:25	:::*	LISTEN	1478/master
tcp6	0	0	:::1:6010	:::*	LISTEN	2098/sshd: root@pts
tcp6	0	0	:::2:660	:::*	LISTEN	805/rpcbind

7.5 打开或关闭指定端口-firewall指令

如果打开防火墙，那么外部请求数据包就不能跟服务监听的端口通讯，此时需要打开指定的端口，比如80、22、8080等

```
1 //查看网络端口情况
2 netstat -anp | more
3
4 // 1.打开端口
5 firewall-cmd --permanent --add-port=端口号/协议
6
7 // 2.关闭端口
8 firewall-cmd --permanent --remove-port=端口号/协议
9
10 // 3.打开或者关闭端口后，需要重新加载才能生效
11 firewall-cmd --reload
12
13 // 查看某个端口是否开放
14 firewall-cmd --query-port=端口号/协议
```

established 代表主机之间已经建立连接关系，通过端口进行通信

Active Internet connections (servers and established) 外网主机					
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	0.0.0.0:111	0.0.0.0:*	LISTEN
tcp 网络协议	0	0	192.168.122.1:53	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:22 主机ip和端口号	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:25	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:6010	0.0.0.0:*	LISTEN
tcp	0	36	192.158.200.101:22	192.158.200.66:52694	ESTABLISHED
tcp6	0	0	:::111	:::*	LISTEN
tcp6	0	0	:::22	:::*	LISTEN
tcp6	0	0	::1:631	:::*	LISTEN
tcp6	0	0	::1:25	:::*	LISTEN
tcp6	0	0	::1:6010	:::*	LISTEN
udp	0	0	0.0.0.0:960	0.0.0.0:*	805/rpcbind
udp	0	0	192.168.122.1:53	0.0.0.0:*	1693/dnsmasq
udp	0	0	0.0.0.0:67	0.0.0.0:*	1693/dnsmasq
udp	0	0	0.0.0.0:111	0.0.0.0:*	805/rpcbind
udp	0	0	0.0.0.0:5353	0.0.0.0:*	780/avahi-daemon: r

打开111端口，重新加载生效

[root@hsp ~]# firewall-cmd --permanent --add-port=111/tcp	打开111端口
success	
[root@hsp ~]# firewall-cmd --reload	重新加载生效
success	
[root@hsp ~]# netstat -anp more	
Active Internet connections (servers and established)	
Proto Recv-Q Send-Q Local Address	Foreign Address
tcp 0 0 0.0.0.0:111	0.0.0.0:*
tcp 0 0 192.168.122.1:53	0.0.0.0:*
tcp 0 0 0.0.0.0:22	0.0.0.0:*
tcp 0 0 127.0.0.1:631	0.0.0.0:*
tcp 0 0 127.0.0.1:25	0.0.0.0:*
tcp 0 0 127.0.0.1:6010	0.0.0.0:*
tcp 0 0 192.158.200.101:22	192.158.200.66:52694
tcp6 0 0 :::111	:::*
tcp6 0 0 :::22	:::*
tcp6 0 0 ::1:631	:::*

查看端口是否打开

[root@hsp ~]# firewall-cmd --query-port=111/tcp	查看某个端口是否打开
yes	
[root@hsp ~]# firewall-cmd --query-port=22/tcp	

7.6 动态监控系统-top指令

top 与 ps 命令相似，都用来显示正在执行的进程，top不同之处在于执行中可以更新正在运行的进程。

```

1 // 动态监控进程
2 top -d 秒数
3
4 top -i
5
6 top -p 进程id
7
8 选项说明:
9   -d 指定top命令每隔几秒更新, 默认是3秒
10  -i 让top不显示任何闲置或者僵死的进程
11  -p 通过进程id来监控

```

7.7 查看系统网络状态–netstat

```

1 // 查看网络情况
2 netstat -anp
3
4 选项说明:
5   -an 按照一定顺序排列输出
6   -p 显示哪个进程在调用

```

```

[root@hsp ~]# netstat -anp | grep sshd  查看远程登录服务的网络情况
tcp        0      0 0.0.0.0:22              0.0.0.0:*
tcp        0      0 127.0.0.1:6010            0.0.0.0:*
tcp        0      36 192.158.200.101:22       192.158.200.66:52694 ESTABLISHED 2098/sshd: root@pts
tcp6       0      0 :::22                  ::::*
tcp6       0      0 ::1:6010                ::::*
unix  3      [ ]          STREAM     CONNECTED     26914    1206/sshd
unix  2      [ ]          DGRAM                    34878    2098/sshd: root@pts
[root@hsp ~]#

```

8 软件包管理–rpm–yum–安装JDK–Tomcat

8.1 rpm 与 yum 包管理

```
1 // 查询已安装的所有软件包
2 rpm -qa | more
3 rpm -qa | grep 软件包名
4
5 // 查询软件包是否安装
6 rpm -q firefox
7
8 // 查询软件包信息
9 rpm -qi 软件包名
10
11 // 查询软件包中的文件
12 rpm -ql 软件包名
13
14 // 查询某个文件所属的软件包
15 rpm -qf /etc/passwd //比如 passwd指令
16 rpm -qf /root/install.log
17
18 // 卸载软件包
19 rpm -e 软件包
20
21 注意：如果其它软件包依赖于你要卸载的软件包，卸载时会产生错误信息，比如 foo,
22 可以增加参数 --nodeps进行强制删除，一般不推荐，会导致该依赖的软件包无法运行
23 rpm -e --nodeps foo
24
25 // 安装rpm包软件
26 rpm -ivh 目录下rpm包全路径名称
27
```

```
yum-plugin-upgrade-helper.noarch  
yum-plugin-verify.noarch  
yum-plugin-versionlock.noarch  
yum-rhn-plugin.noarch  
yum-updateonboot.noarch  
zenity.x86_64  
zlib.i686  
zlib.x86_64  
zlib-devel.i686  
zlib-devel.x86_64  
zlib-static.i686  
zlib-static.x86_64  
zsh.x86_64  
zsh-html.x86_64  
zziplib.i686  
zziplib.x86_64  
zziplib-devel.i686  
zziplib-devel.x86_64  
zziplib-utils.x86_64  
[root@hsp ~]#
```

rpm包名说明：
i686、i386表示32位系统
noarch 表示通用

yum基于rpm包管理，能够从指定的服务器自动下载rpm包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软件包。

```
1 // 查询yum服务器是否有需要安装的软件  
2 yum list | grep 软件包名  
3  
4 // 安装指定的yum包  
5 yum install 软件包名  
6  
7
```

Shell

```
[root@hsp ~]# yum list | grep jdk] 查看yum服务器上已存在的jdk软件包
Repodata is over 2 weeks old. Install yum-cron? Or run: yum makecache fast
copy-jdk-configs.noarch 3.3-10.el7_5 @anaconda
java-1.7.0-openjdk.x86_64 1:1.7.0.261-2.6.22.2.el7_8 @anaconda
java-1.7.0-openjdk-headless.x86_64 1:1.7.0.261-2.6.22.2.el7_8 @anaconda
java-1.8.0-openjdk.x86_64 1:1.8.0.262.b10-1.el7 @anaconda
java-1.8.0-openjdk-headless.x86_64 1:1.8.0.262.b10-1.el7 @anaconda
copy-jdk-configs.noarch 3.3-11.el7_9 updates
java-1.6.0-openjdk.x86_64 1:1.6.0.41-1.13.13.1.el7_3 base
java-1.6.0-openjdk-demo.x86_64 1:1.6.0.41-1.13.13.1.el7_3 base
java-1.6.0-openjdk-devel.x86_64 1:1.6.0.41-1.13.13.1.el7_3 base
java-1.6.0-openjdk-javadoc.x86_64 1:1.6.0.41-1.13.13.1.el7_3 base
java-1.6.0-openjdk-src.x86_64 1:1.6.0.41-1.13.13.1.el7_3 base
java-1.7.0-openjdk-accessibility.x86_64 1:1.7.0.261-2.6.22.2.el7_8 base
java-1.7.0-openjdk-demo.x86_64 1:1.7.0.261-2.6.22.2.el7_8 base
java-1.7.0-openjdk-devel.x86_64 1:1.7.0.261-2.6.22.2.el7_8 base
java-1.7.0-openjdk-javadoc.noarch 1:1.7.0.261-2.6.22.2.el7_8 base
java-1.7.0-openjdk-src.x86_64 1:1.7.0.261-2.6.22.2.el7_8 base
java-1.8.0-openjdk.i686 1:1.8.0.382.b05-1.el7_9 updates
java-1.8.0-openjdk.x86_64 1:1.8.0.382.b05-1.el7_9 updates
java-1.8.0-openjdk-accessibility.i686 1:1.8.0.382.b05-1.el7_9 updates
```

8.2 安装JDK8

华为镜像下载: <https://repo.huaweicloud.com>

JDK下载: <https://repo.huaweicloud.com/java/jdk/>

1. mkdir /opt/jdk
2. 把下载好的 jdk.tar.gz 软件包通过 XFTP 上传到Linux虚拟机的目录下
3. cd /opt/jdk
4. 解压 tar -zxvf 压缩包
5. mkdir /usr/local/java
6. mv /opt/jdk/jdk1.8.0_202 /usr/local/java/ 把解压的目录移动到 /local/java目录下
7. cd /usr/local/java/jdk1.8.0_202
8. cd bin/ 进入bin目录下
9. 执行 ./java -version 其中 ./ 表示在当前目录下执行脚本程序

此时可以看到jdk版本信息，但是切换到其它目录下发现没有了，因为没有配置环境变量，需要在 /etc/profile 文件中添加配置，最后需要重新刷新配置文件生效

10. vim /etc/profile 配置java环境变量，在文件最后添加如下：

```
export JAVA_HOME=/usr/local/java/jdk1.8.0_202
```

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

11. source /etc/profile 刷新配置文件生效

```
[root@hsp home]# echo $PATH  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin  
[root@hsp home]# vim /etc/profile  
[root@hsp home]# cd /usr/local
```

```
# java8  
export JAVA_HOME=/usr/local/java/jdk1.8.0_202  
export PATH=$JAVA_HOME/bin:$PATH
```

注意后面必须拼接上之前的环境变量
否则原来的程序指令就无法使用了

8.3 安装配置Tomcat

tomcat下载: <https://mirrors.huaweicloud.com/apache/tomcat/>

1. mkdir /opt/tomcat 创建tomcat目录并进入
2. cd /opt/tomcat
3. 把下载好的 tomcat.tar.gz 软件包通过 XFTP 上传到Linux虚拟机的目录下
4. tar -zxvf 压缩包名 解压
5. 进入解压后的目录
6. 进入 bin 目录下
7. 执行 ./startup.sh 命令启动tomcat服务

此时在windows宿主机浏览器地址栏中输入 Linux主机ip:8080 发现无法访问，那是因为虚拟机的防火墙服务是处于开启状态的，导致8080端口的请求被拦截下了。需要打开8080服务端口，重新加载后生效，再次访问可以看到tomcat欢迎页面。

8. firewall-cmd --permanent --add-port=8080/tcp 打开8080端口
9. firewall-cmd --reload 重新加载生效
10. firewall-cmd --query-port=8080/tcp 查看8080端口是否开放

```
[root@hsp ~]# mkdir /opt/tomcat  
[root@hsp ~]# cd /opt/tomcat  
[root@hsp tomcat]# ll  
总用 量 10188  
-rw-r--r--. 1 root root 10432020 10月 27 21:10 apache-tomcat-8.5.59.tar.gz  
[root@hsp tomcat]# tar -zxvf apache-tomcat-8.5.59.tar.gz
```

解压，按Tab键可以补全压缩包名

```

[root@hsp tomcat]# ll
总用 量 10192
drwxr-xr-x. 9 root root 4096 10月 27 21:13 apache-tomcat-8.5.59 解压后的目录
-rw-r--r--. 1 root root 10432020 10月 27 21:10 apache-tomcat-8.5.59.tar.gz
[root@hsp tomcat]# cd apache-tomcat-8.5.59/ 进入该解压目录
[root@hsp apache-tomcat-8.5.59]# ls
bin      conf      lib      logs      README.md      RUNNING.txt      webapps
BUILDING.txt  CONTRIBUTING.md  LICENSE  NOTICE  RELEASE-NOTES  temp      work
[root@hsp apache-tomcat-8.5.59]# cd bin 进入其bin 目录下
[root@hsp bin]# ls
bootstrap.jar      ciphers.sh      daemon.sh      shutdown.bat      tomcat-native.tar.gz
catalina.bat       commons-daemon.jar digest.bat      shutdown.sh      tool-wrapper.bat
catalina.sh        commons-daemon-native.tar.gz digest.sh       startup.bat      tool-wrapper.sh
catalina-tasks.xml configtest.bat  setclasspath.bat [startup.sh]  version.bat
ciphers.bat        configtest.sh   setclasspath.sh  tomcat-juli.jar  version.sh
[root@hsp bin]# ./startup.sh 启动tomcat服务
Using CATALINA_BASE: /opt/tomcat/apache-tomcat-8.5.59  ./ 表示在当前目录下去执行某个脚本程序
Using CATALINA_HOME: /opt/tomcat/apache-tomcat-8.5.59
Using CATALINA_TMPDIR: /opt/tomcat/apache-tomcat-8.5.59/temp
Using JRE_HOME: /usr/local/java/jdk1.8.0_202
Using CLASSPATH: /opt/tomcat/apache-tomcat-8.5.59/bin/bootstrap.jar:/opt/tomcat/apache-tomcat-8.5.59
Using CATALINA_OPTS:
Tomcat started.
[root@hsp bin]#

```

查看Linux虚拟机ip

```

[root@hsp bin]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.158.200.101 netmask 255.255.255.0 broadcast 192.158.200.255
        inet6 fe80::db95:87a1:8c51:4711 prefixlen 64 scopeid 0x20<link>
            ether 00:0c:29:8d:e1:9b txqueuelen 1000 (Ethernet)
            RX packets 160218 bytes 213938819 (204.0 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 15154 bytes 4514655 (4.3 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 32 bytes 2592 (2.5 KiB)

```

打开8080网络端口，并重新加载生效

```

[root@hsp bin]# firewall-cmd --permanent --add-port=8080/tcp 打开8080端口网络服务
success
[root@hsp bin]# firewall-cmd --reload 重新加载生效
success
[root@hsp bin]# netstat -anp | more 查看网络端口情况
Active Internet connections (servers and established)
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     805/rpcbind
tcp        0      0 192.168.122.1:53        0.0.0.0:*          LISTEN     1693/dnsmasq
tcp        0      0 0.0.0.0:22           0.0.0.0:*          LISTEN     1206/sshd
tcp        0      0 127.0.0.1:631          0.0.0.0:*          LISTEN     1204/cupsd
tcp        0      0 127.0.0.1:25           0.0.0.0:*          LISTEN     1478/master
tcp        0      0 127.0.0.1:6010         0.0.0.0:*          LISTEN     2098/sshd: root@pts
tcp        0      0 192.158.200.101:22      192.158.200.66:56923 ESTABLISHED 3968/sshd: root@not
tcp        0      36 192.158.200.101:22      192.158.200.66:52694 ESTABLISHED 2098/sshd: root@pts
tcp6       0      0 :::111                ::::*               LISTEN     805/rpcbind
tcp6       0      0 :::8080              ::::*               LISTEN     4650/java
tcp6       0      0 :::22                ::::*               LISTEN     1206/sshd
tcp6       0      0 :::1:631             ::::*               LISTEN     1204/cupsd

```

此时可以看到8080端口已经开放，其它外部主机就可以访问该网络端口了

```

[root@hsp bin]# firewall-cmd --query-port=8080/tcp
yes
[root@hsp bin]# 查看8080端口是否开放

```

浏览器访问，欢迎页面

The screenshot shows a web browser window with the URL `192.158.200.101:8080` in the address bar. The page title is "Linux虚拟机IP + 端口号8080访问". The main content area displays the Apache Tomcat 8.5.59 welcome message: "If you're seeing this, you've successfully installed Tomcat. Congratulations!". Below this, there's a section titled "Developer Quick Start" with links to "Tomcat Setup", "First Web Application", "Realms & AAA", "JDBC DataSources", "Examples", and "Servlet Specifications" and "Tomcat Versions". There are also sections for "Managing Tomcat", "Documentation", and "Getting Help". A sidebar on the right includes links for "Server Status", "Manager App", and "Host Manager". The Apache logo is visible in the top right corner.

8.4 安装 nginx

官网下载：<http://nginx.org/en/download.html>

参考文章：https://blog.csdn.net/qq_45752401/article/details/122660965?

The screenshot shows the NGINX download page. At the top, it says "Join the [NGINXCommunity Slack](#) to ask and answer questions, discuss NGINX, and share useful advice and resources.". Below this, there are three main sections: "Mainline version", "Stable version", and "Legacy versions". Under "Mainline version", there are links for "CHANGES" and "nginx-1.25.3 ppg" and "nginx/Windows-1.25.3 ppg". Under "Stable version", there are links for "CHANGES-1.24", "nginx-1.24.0 ppg", and "nginx/Windows-1.24.0 ppg". Under "Legacy versions", there are links for "CHANGES-1.22", "CHANGES-1.20", "CHANGES-1.18", "nginx-1.22.1 ppg", "nginx-1.20.2 ppg", "nginx-1.18.0 ppg", "nginx/Windows-1.22.1 ppg", "nginx/Windows-1.20.2 ppg", and "nginx/Windows-1.18.0 ppg". On the right side, there is a sidebar with links for "english", "русский", "news", "about", "download", "security", "documentation", "faq", "books", "support", and "trac". The NGINX logo is prominently displayed at the top right.

1. 把下载好的资源包上传到 Linux服务器的某个目录下，进行解压后： `tar -zxvf 资源包`

```
[root@hsp ~]# cd services
[root@hsp services]# ls
nginx-1.24.0  nginx-1.24.0.tar.gz  user-center-backend  user-center-frontend
[root@hsp services]#
```

2. 进入解压后的目录 cd nginx-1.24.0, 执行如下命令配置, 可以看到目录下, 多了一个 Makefile文件

```
1 # 配置configure
2 # --prefix 代表安装的路径,
3 # --with-http_ssl_module 安装ssl,
4 # --with-http_stub_status_module查看nginx的客户端状态
5
6 ./configure --prefix=/usr/local/nginx-1.24.0 --with-http_ssl_module --with-
http_stub_status_module
7
```

如果你执行的时候出现以下问题:

问题1:

```
[root@localhost nginx-1.20.2]# ./configure --prefix=/usr/local/nginx-1.20.2 --with-http_ssl_module --with-http_stub_status_module
checking for OS
+ Linux 3.10.0-514.el7.x86_64 x86_64
checking for C compiler ... not found

./configure: error: C compiler cc is not found

[root@localhost nginx-1.20.2]#
```

CSDN @子非我鱼

- 输入以下命令

```
1 yum -y install gcc gcc-c++ autoconf automake make
```

问题2:

```
./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.
```

CSDN @子非我鱼

- 输入以下命令

```
1 yum -y install openssl openssl-devel
```

3. 进行编译安装 nginx

```
1 // 编译安装
2 make & make install
```

4. 进入到刚才指定的安装位置目录下，进入sbin目录下，执行一下命令启动

```
1 cd /usr/local/nginx-1.24.0/sbin
2
3 // 启动nginx服务
4 ./nginx
```

```
[root@hsp services]# cd nginx-1.24.0/
[root@hsp nginx-1.24.0]# ls
auto  CHANGES  CHANGES.ru  conf  configure  contrib  html  LICENSE  Makefile  man  objs  README  src
[root@hsp nginx-1.24.0]# cd /usr/local
[root@hsp local]# ls
bin  etc  games  include  java  lib  lib64  libexec  mysql  nginx-1.20.2  sbin  share  src
[root@hsp local]# cd nginx-1.20.2/
[root@hsp nginx-1.20.2]# ls
client_body_temp  conf  fastcgi_temp  html  logs  proxy_temp  sbin  scgi_temp  uwsgi_temp
[root@hsp nginx-1.20.2]# cd sbin
[root@hsp sbin]# ll
总用 量 5928
-rwxr-xr-x. 1 root root 6068712 11月 12 18:59 nginx
[root@hsp sbin]# ./nginx
```

5. 配置环境变量，这样在任意目录下都可以使用 nginx 命令了

```
1 // 编辑配置文件
2 vim /etc/profile
3
4 // 刷新文件生效
5 source /etc/profile
```

```
done

unset i
unset -f pathmunge

# java8
export JAVA_HOME=/usr/local/java/jdk1.8.0_202
export PATH=$JAVA_HOME/bin:$PATH
export PATH=/usr/local/nginx-1.20.2/sbin:$PATH

~
~
~
```

6. 访问： 虚拟机IP:80 （如果浏览器访问不了， 可能是被防火墙拦截网络请求了， 可以直接关闭防火墙， 但不推荐， 最好是通过 firewall 命令来开放指定的端口）

```
1 # 1. 添加 --permanent永久生效，没有此参数重启后失效 这里的80为nginx服务的端口，若为
2     其他服务设置，需要对应端口
3     firewall-cmd --zone=public --add-port=6379/tcp --permanent
4
5 # 2. 重新载入，使得配置生效
6     firewall-cmd --reload
7
8 # 查看所有打开的端口
9     firewall-cmd --zone=public --list-ports
10
11 # 防火墙的关闭：
12     systemctl stop firewalld //一般不用
13
14 # 防火墙的启动
15     systemctl start firewalld
```

7. 如果需要修改 nginx 配置，在安装目录 conf 下找到 nginx.conf 文件，修改之后需要重新加载生效

```
1 // 启动 nginx 服务
2 nginx
3
4 // 重新加载配置文件生效
5 nginx -s reload
```

```
[root@hsp services]# cd /usr/local
[root@hsp local]# cd nginx-1.20.2/
[root@hsp nginx-1.20.2]# ls
client_body_temp  conf  fastcgi_temp  html  logs  proxy_temp  sbin  scgi_temp
[root@hsp nginx-1.20.2]# cd conf
[root@hsp conf]# ll
总用量 68
-rw-r--r--. 1 root root 1077 11月 12 18:59 fastcgi.conf
-rw-r--r--. 1 root root 1077 11月 12 18:59 fastcgi.conf.default
-rw-r--r--. 1 root root 1007 11月 12 18:59 fastcgi_params
-rw-r--r--. 1 root root 1007 11月 12 18:59 fastcgi_params.default
-rw-r--r--. 1 root root 2837 11月 12 18:59 koi-utf
-rw-r--r--. 1 root root 2223 11月 12 18:59 koi-win
-rw-r--r--. 1 root root 5349 11月 12 18:59 mime.types
-rw-r--r--. 1 root root 5349 11月 12 18:59 mime.types.default
-rw-r--r--. 1 root root 2684 11月 12 20:44 nginx.conf
-rw-r--r--. 1 root root 2656 11月 12 18:59 nginx.conf.default
-rw-r--r--. 1 root root 636 11月 12 18:59 scgi_params
-rw-r--r--. 1 root root 636 11月 12 18:59 scgi_params.default
-rw-r--r--. 1 root root 664 11月 12 18:59 uwsgi_params
-rw-r--r--. 1 root root 664 11月 12 18:59 uwsgi_params.default
-rw-r--r--. 1 root root 3610 11月 12 18:59 win-utf
[root@hsp conf]#
```

```
server {
    listen      80;
    server_name localhost;

    #charset koi8-r;

    #access_log  logs/host.access.log  main;
                                            指定服务的资源目录
    location / {
        root  /root/services/user-center-frontend;
        index index.html index.htm;
    }

    #error_page  404          /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
```

```

user    root;      如果访问nginx 出现 403页面，需要修改
worker_processes 1; 此处的 root 用户

#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;

#pid        logs/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include      mime.types;
    default_type application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "'.$http_version'"';
}

```

8. 查看所有启动的服务情况

```

1 netstat -ntlp
2
3 netstat -anp | more
4
5 netstat -anp | grep nginx

[root@hsp ~]# nginx 启动 nginx
[root@hsp ~]# netstat -ntlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp        0      0 192.168.122.1:53       0.0.0.0:*
tcp        0      0 0.0.0.0:22            0.0.0.0:*
tcp        0      0 127.0.0.1:631         0.0.0.0:*
tcp        0      0 127.0.0.1:6010        0.0.0.0:*
tcp        0      0 0.0.0.0:111          0.0.0.0:*
tcp        0      0 0.0.0.0:80           0.0.0.0:*
tcp6       0      0 :::22                 :::*
tcp6       0      0 ::1:631              :::*
tcp6       0      0 ::1:6010             :::*
tcp6       0      0 ::::111              :::*
[root@hsp ~]#

```

8.5 安装 mysql 5.7

参考文章：https://blog.csdn.net/qq_20780541/article/details/122035569?

官网下载：<https://downloads.mysql.com/archives/community/>

MySQL5.7资源包：(mysql-5.7.30-linux-glibc2.12-x86_64.tar.gz)

Please note that these are old versions. New releases will have recent bug fixes and features!
To download the latest release of MySQL Community Server, please visit [MySQL Downloads](#).

Product Version: 5.7.30
Operating System: Linux - Generic
OS Version: Linux - Generic (glibc 2.12) (x86, 64-bit)

Compressed TAR Archive	Mar 23, 2020	629.4M	Download
(mysql-5.7.30-linux-glibc2.12-x86_64.tar.gz)			MDS: 611be3b18a30498b705db773293ad341 Signature
Compressed TAR Archive, Test Suite	Mar 23, 2020	30.7M	Download
(mysql-test-5.7.30-linux-glibc2.12-x86_64.tar.gz)			MDS: 0F76e92eebbf9be40b798477f74516b Signature
TAR	Mar 23, 2020	660.2M	Download
(mysql-5.7.30-linux-glibc2.12-x86_64.tar)			MDS: bbc988b3acd6949da677da16f49dc16c Signature

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

```
[root@hsp ~]# service mysql start 启动 mysql 服务
Starting MySQL SUCCESS!
[root@hsp ~]# 2023-11-13T07:32:47.126165Z mysqld_safe A mysqld process already exists

[root@hsp ~]# netstat -ntlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp      0      0 192.168.122.1:53        0.0.0.0:*           LISTEN    1621/dnsmasq
tcp      0      0 0.0.0.0:22            0.0.0.0:*           LISTEN    1209/sshd
tcp      0      0 127.0.0.1:631          0.0.0.0:*           LISTEN    1208/cupsd
tcp      0      0 127.0.0.1:6010          0.0.0.0.*          LISTEN    2142/sshd: root@pts
tcp      0      0 0.0.0.0:111           0.0.0.0.*          LISTEN    753/rpcbind
tcp      0      0 0.0.0.0:80            0.0.0.0.*          LISTEN    2551/nginx: master
tcp6     0      0 :::22                 ::::*               LISTEN    1209/sshd
tcp6     0      0 ::1:631              ::::*               LISTEN    1208/cupsd
tcp6     0      0 ::1:6010             ::::*               LISTEN    2142/sshd: root@pts
tcp6     0      0 ::::3306             ::::*               LISTEN    2949/mysqld
tcp6     0      0 ::::111              ::::*               LISTEN    753/rpcbind
[root@hsp ~]#
```

```
[root@hsp ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.30 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

9 P91–P153集未看