目录

[一、基础 5](#_Toc16593797)

[比Windows的好处 5](#_Toc16593798)

[分区原理 5](#_Toc16593799)

[和Windows的不同 6](#_Toc16593800)

[启动过程 6](#_Toc16593801)

[登录 16](#_Toc16593802)

[查看centos版本 17](#_Toc16593803)

[图形界面 17](#_Toc16593804)

[显示电脑以及操作系统的相关信息 18](#_Toc16593805)

[显示正在运行的内核版本 18](#_Toc16593806)

[显示发行版本信息 18](#_Toc16593807)

[几种提高工作效率的方法 18](#_Toc16593808)

[别名生效 18](#_Toc16593809)

[二、文件夹 18](#_Toc16593810)

[进入文件夹 cd 18](#_Toc16593811)

[/：根目录 19](#_Toc16593812)

[查看当前目录pwd 20](#_Toc16593813)

[./ 20](#_Toc16593814)

[默认的目录 20](#_Toc16593815)

[/home 21](#_Toc16593816)

[/etc 21](#_Toc16593817)

[/bin、/usr/bin 21](#_Toc16593818)

[/boot 21](#_Toc16593819)

[/dev 21](#_Toc16593820)

[/lib、/usr/lib、/usr/local/lib 22](#_Toc16593821)

[/lost+fount 22](#_Toc16593822)

[/mnt: /media 22](#_Toc16593823)

[/opt 22](#_Toc16593824)

[/proc 22](#_Toc16593825)

[/root 22](#_Toc16593826)

[/sbin、/usr/sbin、/usr/local/sbin 23](#_Toc16593827)

[/tmp 23](#_Toc16593828)

[/srv 23](#_Toc16593829)

[/usr 23](#_Toc16593830)

[/var 23](#_Toc16593831)

[用户目录 24](#_Toc16593832)

[绝对路径 24](#_Toc16593833)

[相对路径 24](#_Toc16593834)

[.和.. 24](#_Toc16593835)

[建立目录 24](#_Toc16593836)

[目录改名 25](#_Toc16593837)

[mv命令 25](#_Toc16593838)

[目录删除 25](#_Toc16593839)

[文件权限 26](#_Toc16593840)

[访问用户 27](#_Toc16593841)

[访问权限 28](#_Toc16593842)

[三、文本 29](#_Toc16593843)

[文本编辑：vim 29](#_Toc16593844)

[文本编辑：vi 30](#_Toc16593845)

[f.搜索 31](#_Toc16593846)

[安装文本编辑器(UTF-8)gedit 32](#_Toc16593847)

[四、常用命令 32](#_Toc16593848)

[ls 32](#_Toc16593849)

[处理目录的常用命令 32](#_Toc16593850)

[内容查看 33](#_Toc16593851)

[链接 33](#_Toc16593852)

[touch 36](#_Toc16593853)

[echo 36](#_Toc16593854)

[cat 36](#_Toc16593855)

[find 36](#_Toc16593856)

[tail -10 36](#_Toc16593857)

[head -20 36](#_Toc16593858)

[压缩 36](#_Toc16593859)

[五、系统配置 37](#_Toc16593860)

[查询系统负载 37](#_Toc16593861)

[ip地址配置 38](#_Toc16593862)

[DNS服务器地址配置 39](#_Toc16593863)

[hosts主机表文件 40](#_Toc16593864)

[修改网络配置 40](#_Toc16593865)

[重启网络服务 41](#_Toc16593866)

[tracerout 41](#_Toc16593867)

[和Linux之间通过ftp交互 41](#_Toc16593868)

[字符集 42](#_Toc16593869)

[yum 43](#_Toc16593870)

[wget 43](#_Toc16593871)

[安装Nginx 43](#_Toc16593872)

[环境变量 44](#_Toc16593873)

[安装Java环境 46](#_Toc16593874)

[安装tomcat 47](#_Toc16593875)

[修改网络配置 47](#_Toc16593876)

[安装redis 47](#_Toc16593877)

[解压 48](#_Toc16593878)

[安装浏览器 48](#_Toc16593879)

[重启 48](#_Toc16593880)

[关机 49](#_Toc16593881)

[六、Shell 49](#_Toc16593882)

[Shell是什么 49](#_Toc16593883)

[七、云服务器 49](#_Toc16593884)

[登录 49](#_Toc16593885)

[连接中断 52](#_Toc16593886)

[上传文件到服务器 53](#_Toc16593887)

[创建文件夹 53](#_Toc16593888)

[安装yum 53](#_Toc16593889)

[安装java环境腾讯云默认没有FTP，安装FTP 55](#_Toc16593890)

[安装FTP： 55](#_Toc16593891)

[安装java环境 56](#_Toc16593892)

[安装tomcat环境 57](#_Toc16593893)

[五、常见错误 57](#_Toc16593894)

[swap file “\*.swp”already exists！ 57](#_Toc16593895)

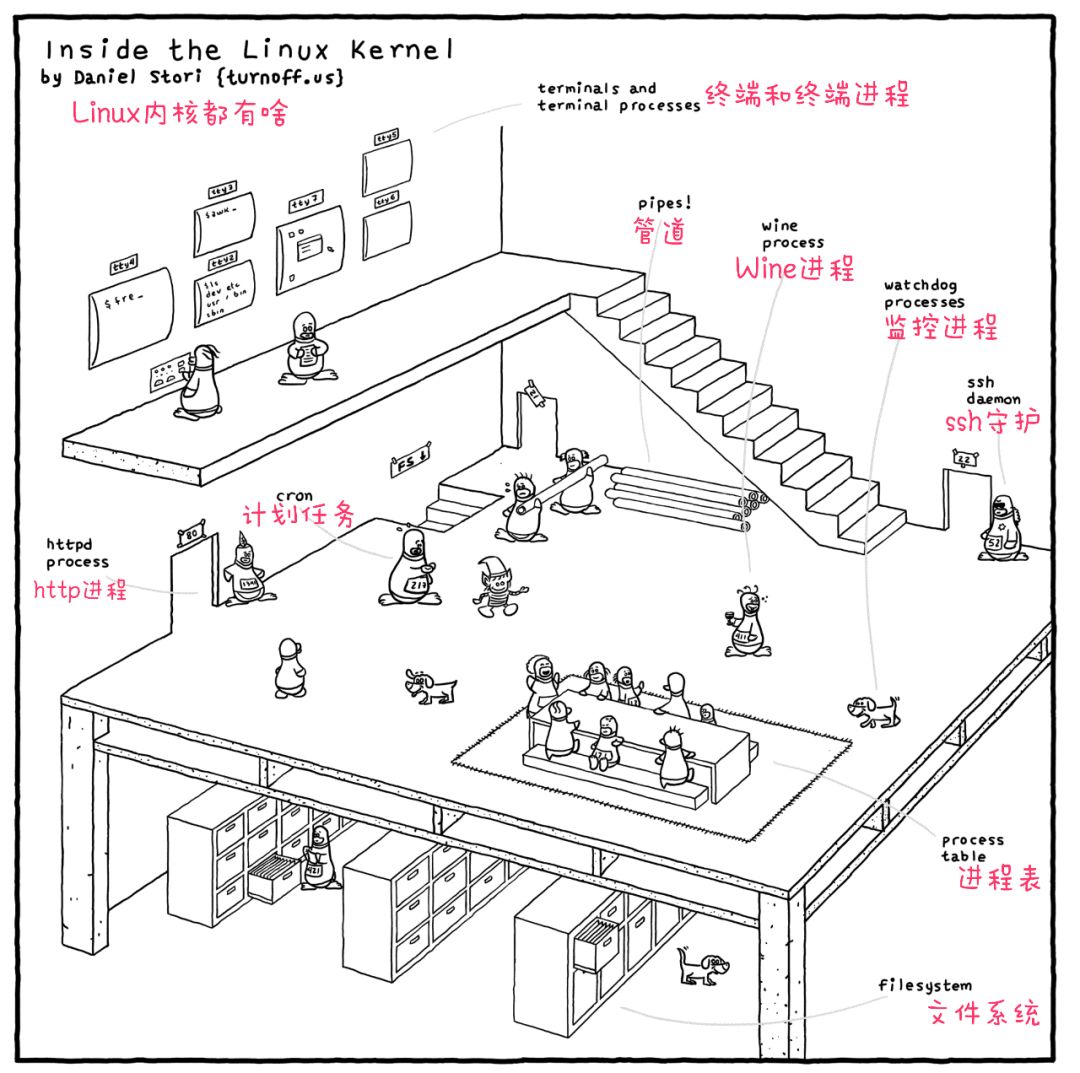
# 一、基础

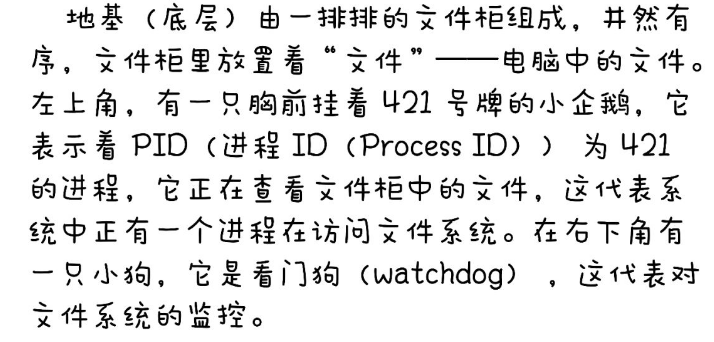
比Windows的好处

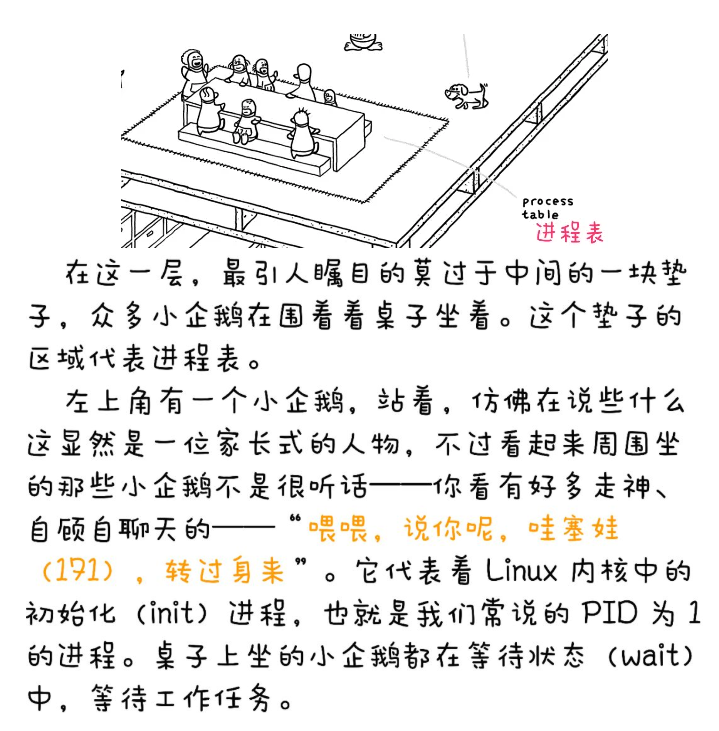
Linux所有的都是文件，分区、目录、硬件都是文件，且没有类似exe这样的扩展名，文件类型是通过权限体现的，Windows上的大量病毒在Linux上是无法识别的。

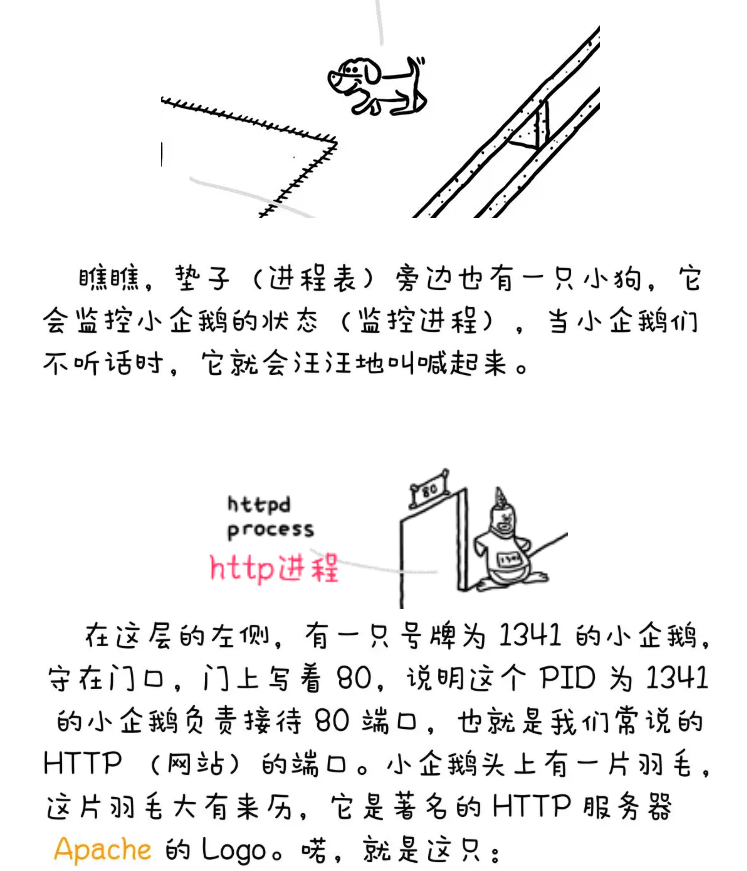
Linux可以没有图形界面，节省系统资源。

Linux全貌

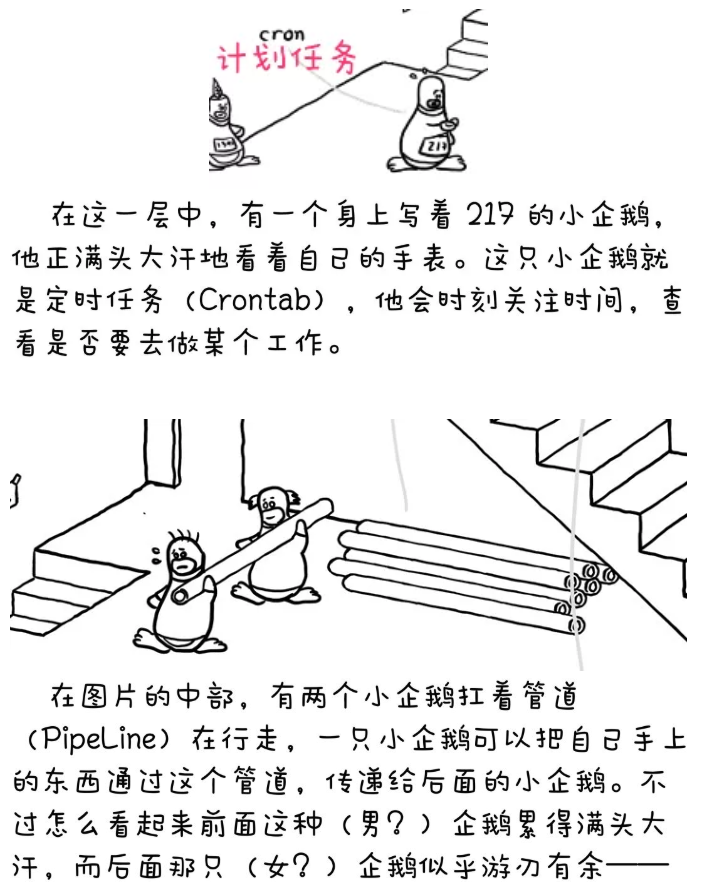








上述，nginx也是同理，当然也不一定是80端口。

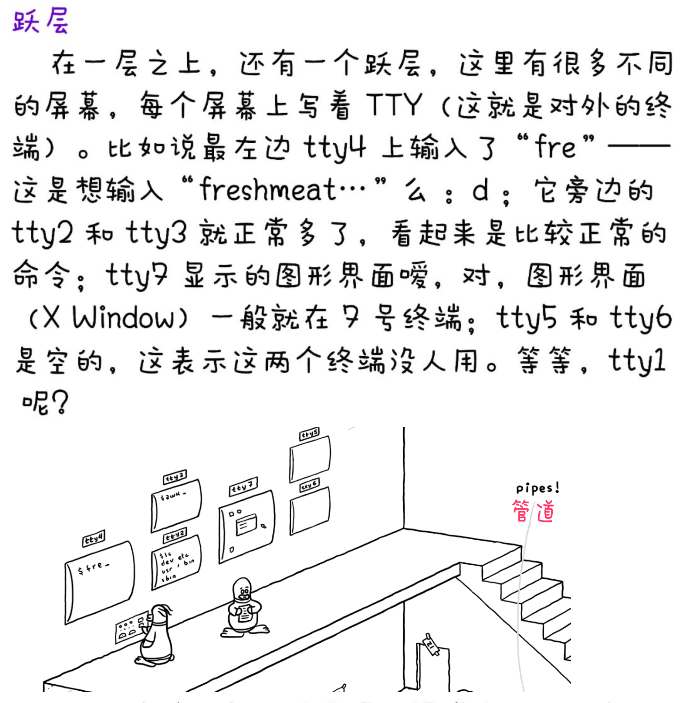


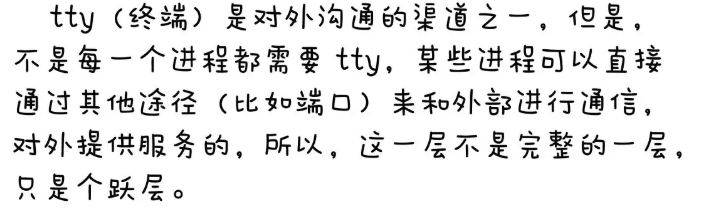
初识pipeline

从一个现象说起，有一家咖啡吧生意特别好，每天来的客人络绎不绝，客人A来到柜台，客人B紧随其后，客人C排在客人B后面，客人D排在客人C后面，客人E排在客人D后面，一直排到店面门外。老板和三个员工首先为客人A准备食物：员工甲拿了一个干净的盘子，然后员工乙在盘子里装上薯条，员工丙再在盘子里放上豌豆，老板最后配上一杯饮料，完成对客人A的服务，送走客人A，下一位客人B开始被服务。然后员工甲又拿了一个干净的盘子，员工乙又装薯条，员工丙又放豌豆，老板又配上了一杯饮料，送走客人B，客人C开始被服务。一直重复下去。

从效率方面观察这个现象，当服务客人A时，在员工甲拿了一个盘子后，员工甲一直处于空闲状态，直到送走客人A，客人B被服务。老板自然而然的就会想到如果每个人都不停的干活，就可以服务更多的客人，赚到更多的钱。老板通过不停的尝试想出了一个办法。以客户A，B为例阐述这个方法：员工甲为客户A准备好了盘子后，在员工乙开始为客户A装薯条的同时，员工甲开始为客户B准备托盘。这样员工甲就可以不停的进行生产。整个过程如下图，客户们围着咖啡吧台排队，因为有四个生产者，一个老板加三个员工，所以可以同时服务四个客户。我们将目光转向老板，单位时间从他那里出去的客户数提高了将近四倍，也就是说效率提高将近四倍。

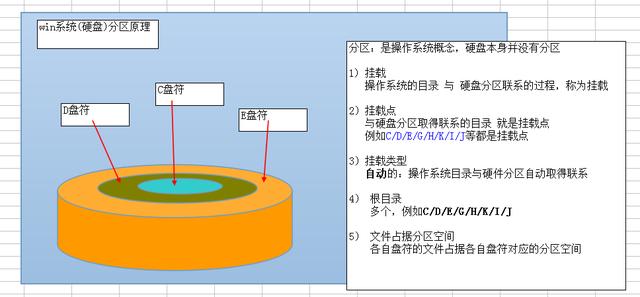
pipeline的概念可以从这里抽象出来：将一件需要重复做的事情（这里指为客户准备一份精美的食物）切割成各个不同的阶段（这里是四个阶段：盘子，薯条，豌豆，饮料），每一个阶段由独立的单元负责（四个生产者分别负责不同的环节）。所有待执行的对象依次进入作业队列（这里是所有的客户排好队依次进入服务，除了开始和结尾的一段时间，任意时刻，四个客户被同时服务）。对应到CPU中，每一条指令的执行过程可以切割成：fetch instruction、decode it、find operand、perform action、store result 5个阶段。



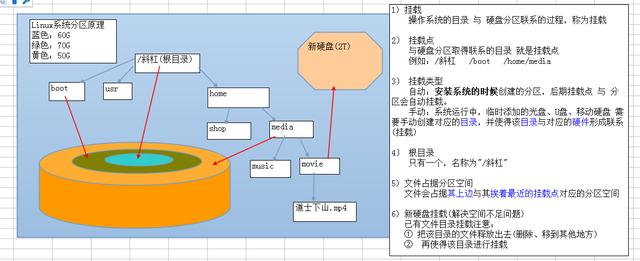


分区原理

Windows: 类似：倒着的森林



Linux: 类似：倒着的一个"大树"



和Windows的不同

Linux区分大小写；

Linux中所有内容以文件形式保存，包括硬件、用户等，有些临时的操作可能不用保存到文件，只要一直有效，就必须保存到文件；

Linux不靠扩展名区分文件类型。

启动过程

在BIOS阶段，计算机的行为基本上被写死了，可以做的事情并不多；一般就是通电、BIOS、主引导记录、操作系统这四步。所以我们一般认为加载内核是linux启动流程的第一步。



**第一步、加载内核**

操作系统接管硬件以后，首先读入 /boot 目录下的内核文件。



$ ls /boot  
　　  
　　config-3.2.0-3-amd64  
　　config-3.2.0-4-amd64  
　　grub  
　　initrd.img-3.2.0-3-amd64  
　　initrd.img-3.2.0-4-amd64  
　　System.map-3.2.0-3-amd64  
　　System.map-3.2.0-4-amd64  
　　vmlinuz-3.2.0-3-amd64  
　　vmlinuz-3.2.0-4-amd64

**第二步、启动初始化进程**

内核文件加载以后，就开始运行第一个程序 /sbin/init，它的作用是初始化系统环境。

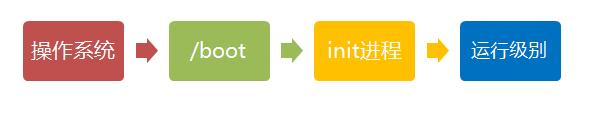


由于init是第一个运行的程序，它的进程编号（pid）就是1。其他所有进程都从它衍生，都是它的子进程。

**第三步、确定运行级别**

许多程序需要开机启动。它们在Windows叫做"服务"（service），在Linux就叫做"守护进程"（daemon）。

init进程的一大任务，就是去运行这些开机启动的程序。但是，不同的场合需要启动不同的程序，比如用作服务器时，需要启动Apache，用作桌面就不需要。Linux允许为不同的场合，分配不同的开机启动程序，这就叫做"运行级别"（runlevel）。也就是说，启动时根据"运行级别"，确定要运行哪些程序。



我们查看一下，/boot 目录下面大概是这样一些文件：

$ ls /boot  
　　  
　　config-3.2.0-3-amd64  
　　config-3.2.0-4-amd64  
　　grub  
　　initrd.img-3.2.0-3-amd64  
　　initrd.img-3.2.0-4-amd64  
　　System.map-3.2.0-3-amd64  
　　System.map-3.2.0-4-amd64  
　　vmlinuz-3.2.0-3-amd64  
　　vmlinuz-3.2.0-4-amd64

**第二步、启动初始化进程**

内核文件加载以后，就开始运行第一个程序 /sbin/init，它的作用是初始化系统环境。

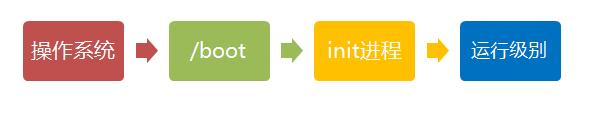


由于init是第一个运行的程序，它的进程编号（pid）就是1。其他所有进程都从它衍生，都是它的子进程。

**第三步、确定运行级别**

许多程序需要开机启动。它们在Windows叫做"服务"（service），在Linux就叫做"守护进程"（daemon）。

init进程的一大任务，就是去运行这些开机启动的程序。但是，不同的场合需要启动不同的程序，比如用作服务器时，需要启动Apache，用作桌面就不需要。Linux允许为不同的场合，分配不同的开机启动程序，这就叫做"运行级别"（runlevel）。也就是说，启动时根据"运行级别"，确定要运行哪些程序。



Linux预置七种运行级别（0-6）。一般来说，0是关机，1是单用户模式（也就是维护模式），6是重启。运行级别2-5，各个发行版不太一样，对于Debian来说，都是同样的多用户模式（也就是正常模式）。

init进程首先读取文件 /etc/inittab，它是运行级别的设置文件。如果你打开它，可以看到第一行是这样的：

id:2:initdefault:

initdefault的值是2，表明系统启动时的运行级别为2。如果需要指定其他级别，可以手动修改这个值。

那么，运行级别2有些什么程序呢，系统怎么知道每个级别应该加载哪些程序呢？回答是每个运行级别在/etc目录下面，都有一个对应的子目录，指定要加载的程序。

/etc/rc0.d  
 /etc/rc1.d  
 /etc/rc2.d  
 /etc/rc3.d  
 /etc/rc4.d  
 /etc/rc5.d  
 /etc/rc6.d

上面目录名中的"rc"，表示run command（运行程序），最后的d表示directory（目录）。下面让我们看看 /etc/rc2.d 目录中到底指定了哪些程序。

　　$ ls /etc/rc2.d  
　　  
　　README  
　　S01motd  
　　S13rpcbind  
　　S14nfs-common  
　　S16binfmt-support  
　　S16rsyslog  
　　S16sudo  
　　S17apache2  
　　S18acpid  
　　...

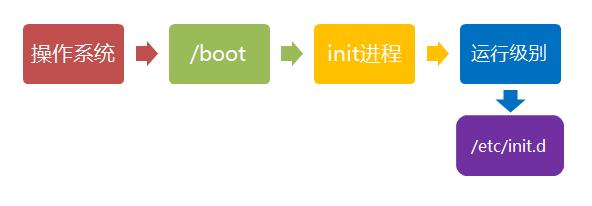
可以看到，除了第一个文件README以外，其他文件名都是"字母S+两位数字+程序名"的形式。字母S表示Start，也就是启动的意思（启动脚本的运行参数为start），如果这个位置是字母K，就代表Kill（关闭），即如果从其他运行级别切换过来，需要关闭的程序（启动脚本的运行参数为stop）。后面的两位数字表示处理顺序，数字越小越早处理，所以第一个启动的程序是motd，然后是rpcbing、nfs......数字相同时，则按照程序名的字母顺序启动，所以rsyslog会先于sudo启动。

这个目录里的所有文件（除了README），就是启动时要加载的程序。如果想增加或删除某些程序，不建议手动修改 /etc/rcN.d 目录，最好是用专门命令进行管理。

**第四步、加载开机启动程序**

前面提到，七种预设的"运行级别"各自有一个目录，存放需要开机启动的程序。不难想到，如果多个"运行级别"需要启动同一个程序，那么这个程序的启动脚本，就会在每一个目录里都有一个拷贝。这样会造成管理上的困扰：如果要修改启动脚本，岂不是每个目录都要改一遍？

Linux的解决办法，就是七个 /etc/rcN.d 目录里列出的程序，都设为链接文件，指向另外一个目录 /etc/init.d ，真正的启动脚本都统一放在这个目录中。init进程逐一加载开机启动程序，其实就是运行这个目录里的启动脚本。



下面就是链接文件真正的指向。

　　$ ls -l /etc/rc2.d  
　　  
　　README  
　　S01motd -> ../init.d/motd  
　　S13rpcbind -> ../init.d/rpcbind  
　　S14nfs-common -> ../init.d/nfs-common  
　　S16binfmt-support -> ../init.d/binfmt-support  
　　S16rsyslog -> ../init.d/rsyslog  
　　S16sudo -> ../init.d/sudo  
　　S17apache2 -> ../init.d/apache2  
　　S18acpid -> ../init.d/acpid  
　　...

这样做的另一个好处，就是如果你要手动关闭或重启某个进程，直接到目录 /etc/init.d 中寻找启动脚本即可。比如，我要重启Apache服务器，就运行下面的命令：

$ sudo /etc/init.d/apache2 restart

/etc/init.d 这个目录名最后一个字母d，是directory的意思，表示这是一个目录，用来与程序 /etc/init 区分。

**第五步、用户登录**

开机启动程序加载完毕以后，就要让用户登录了。



一般来说，用户的登录方式有三种：

（1）命令行登录（2）ssh登录（3）图形界面登录

这三种情况，都有自己的方式对用户进行认证。

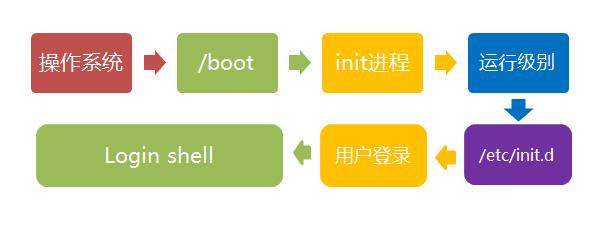
（1）命令行登录：init进程调用getty程序（意为get teletype），让用户输入用户名和密码。输入完成后，再调用login程序，核对密码（linux还会再多运行一个身份核对程序/etc/pam.d/login）。如果密码正确，就从文件 /etc/passwd 读取该用户指定的shell，然后启动这个shell。

（2）ssh登录：这时系统调用sshd程序（linux还会再运行/etc/pam.d/ssh ），取代getty和login，然后启动shell。

（3）图形界面登录：init进程调用显示管理器，Gnome图形界面对应的显示管理器为gdm（GNOME Display Manager），然后用户输入用户名和密码。如果密码正确，就读取/etc/gdm3/Xsession，启动用户的会话。

**第六步、进入 login shell**

所谓shell，简单说就是命令行界面，让用户可以直接与操作系统对话。用户登录时打开的shell，就叫做login shell。



linux默认的shell是Bash，它会读入一系列的配置文件。上一步的三种情况，在这一步的处理，也存在差异。

（1）命令行登录：首先读入 /etc/profile，这是对所有用户都有效的配置；然后依次寻找下面三个文件，这是针对当前用户的配置。

　　~/.bash\_profile  
　　~/.bash\_login  
　　~/.profile

需要注意的是，这三个文件只要有一个存在，就不再读入后面的文件了。比如，要是 ~/.bash\_profile 存在，就不会再读入后面两个文件了。

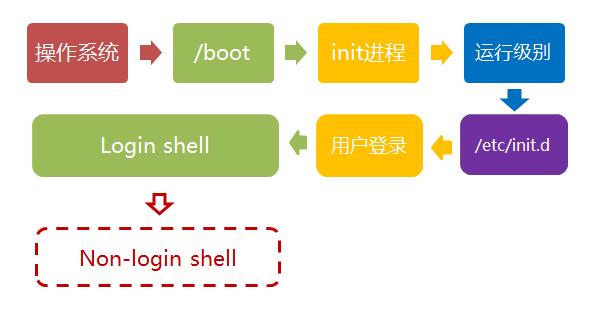
（2）ssh登录：与第一种情况完全相同。

（3）图形界面登录：只加载 /etc/profile 和 ~/.profile。也就是说，~/.bash\_profile 不管有没有，都不会运行。

**第七步，打开 non-login shell**

老实说，上一步完成以后，Linux的启动过程就算结束了，用户已经可以看到命令行提示符或者图形界面了。但是，为了内容的完整，必须再介绍一下这一步。

用户进入操作系统以后，常常会再手动开启一个shell。这个shell就叫做 non-login shell，意思是它不同于登录时出现的那个shell，不读取/etc/profile和.profile等配置文件。



non-login shell的重要性，不仅在于它是用户最常接触的那个shell，还在于它会读入用户自己的bash配置文件 ~/.bashrc。大多数时候，我们对于bash的定制，都是写在这个文件里面的。

你也许会问，要是不进入 non-login shell，岂不是.bashrc就不会运行了，因此bash 也就不能完成定制了？事实上，Debian已经考虑到这个问题了，请打开文件 ~/.profile，可以看到下面的代码：

　　if [ -n "$BASH\_VERSION" ]; then  
　　　　if [ -f "$HOME/.bashrc" ]; then  
　　　　　　. "$HOME/.bashrc"  
　　　　fi  
　　fi

上面代码先判断变量 $BASH\_VERSION 是否有值，然后判断主目录下是否存在 .bashrc 文件，如果存在就运行该文件。第三行开头的那个点，是source命令的简写形式，表示运行某个文件，写成"source ~/.bashrc"也是可以的。

因此，只要运行～/.profile文件，～/.bashrc文件就会连带运行。但是上一节的第一种情况提到过，如果存在～/.bash\_profile文件，那么有可能不会运行～/.profile文件。解决这个问题很简单，把下面代码写入.bash\_profile就行了。

　　if [ -f ~/.profile ]; then  
　　　　. ~/.profile  
　　fi

这样一来，不管是哪种情况，.bashrc都会执行，用户的设置可以放心地都写入这个文件了。

Bash的设置之所以如此繁琐，是由于历史原因造成的。早期的时候，计算机运行速度很慢，载入配置文件需要很长时间，Bash的作者只好把配置文件分成了几个部分，阶段性载入。系统的通用设置放在 /etc/profile，用户个人的、需要被所有子进程继承的设置放在.profile，不需要被继承的设置放在.bashrc。

顺便提一下，除了Linux以外， Mac OS X 使用的shell也是Bash。但是，它只加载.bash\_profile，然后在.bash\_profile里面调用.bashrc，而且不管是ssh登录还是在图形界面里启动shell窗口都是如此。

登录

输入用户名root，密码。

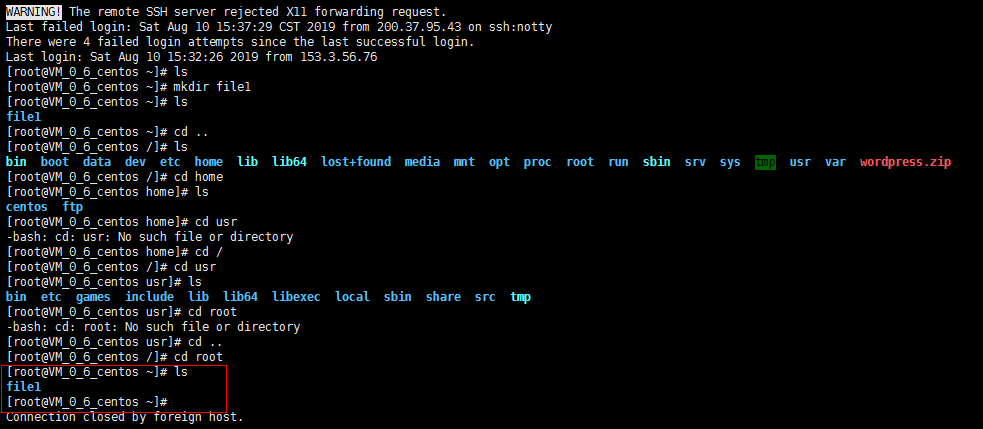
# startx

如提示command not found说明图形界面没安装。

用户登录后默认是家目录，因为root用户的家目录是根目录下的root用户，因此root用户登录后，默认是root目录。

cd ~ 进入家目录

cd /进入根目录。



查看centos版本



cat /etc/redhat-release

图形界面

a.命令行界面切换到图形界面：# starx

图形界面切换到命令行界面：ctrl+alt+F2

b.安装图形界面，可以使用yum，必须在联网状态下，以下是例子：

1. /\*\*
2. \* -----------------------------------------------------------------------------------------------------------
3. \* CentOS-6.4-minimal版中安装图形用户界面
4. \* 1)安装GNOME桌面环境
5. \*   [root@CentOS64 ~]# yum groupinstall -y "Desktop"
6. \* 2)安装XWindow
7. \*   [root@CentOS64 ~]# yum groupinstall -y "X Window System"
8. \* 3)安装中文语言
9. \*   [root@CentOS64 ~]# yum groupinstall -y "Chinese Support"
10. \* 4)启动GNOME
11. \*   [root@CentOS64 ~]# startx  (在CentOS系统上执行此命令,若命令执行后没有显示桌面,可以尝试按下Ctrl+Alt+F2进入桌面)
12. \* 5)若想在下次系统启动后直接进入桌面环境
13. \*   需修改/etc/inittab文件中的"id:3:initdefault"为"id:5:initdefault",然后重启系统即可
14. \* -----------------------------------------------------------------------------------------------------------
15. \* @create Jan 15, 2014 9:42:37 PM
16. \* @author 玄玉<http://blog.csdn.net/jadyer>

显示电脑以及操作系统的相关信息

uname -a

uname 命令用于查看系统内核与系统版本等信息，格式为“uname [-a]”。

uname -r 查看内核

uname --help 帮助

显示正在运行的内核版本

cat /proc/version

显示发行版本信息

cat /etc/issue

几种提高工作效率的方法

自动补全命令TAB键

历史命令上下箭头键

别名alias

别名生效

自动source ~/.bash\_profile

yum命令

**yum命令**是在Fedora和RedHat以及SUSE中基于[rpm](http://man.linuxde.net/rpm)的软件包管理器，它可以使系统管理人员交互和自动化地更细与管理RPM软件包，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体包，无须繁琐地一次次下载、安装。

yum -y update：升级所有包同时也升级软件和系统内核；

yum -y upgrade：只升级所有包，不升级软件和系统内核。

说明：生产环境对软件版本和内核版本要求非常精确，别没事有事随便的进行yum update操作！！！！！！！！！

设置服务开机启动

systemctl enable jenkins.service #设置jenkins服务为自启动服务

sysstemctl start  jenkins.service #启动jenkins服务

# 二、文件夹

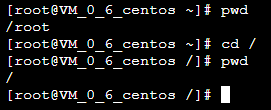
进入文件夹 cd

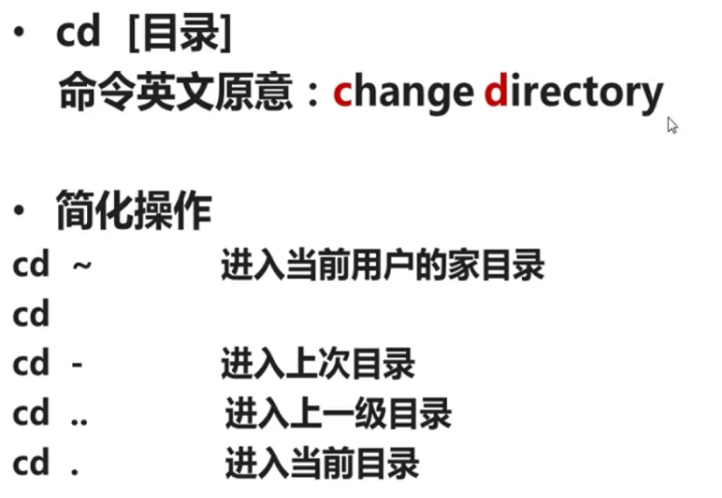
如：#cd /etc/sysconfig/

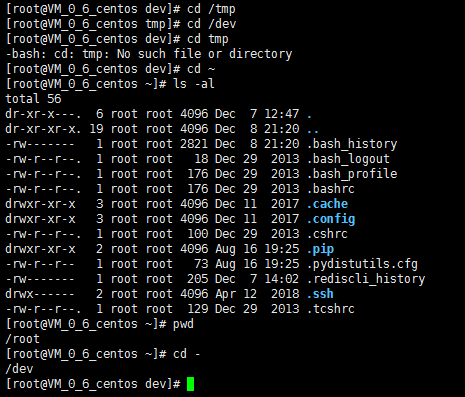
#cd network-scripts

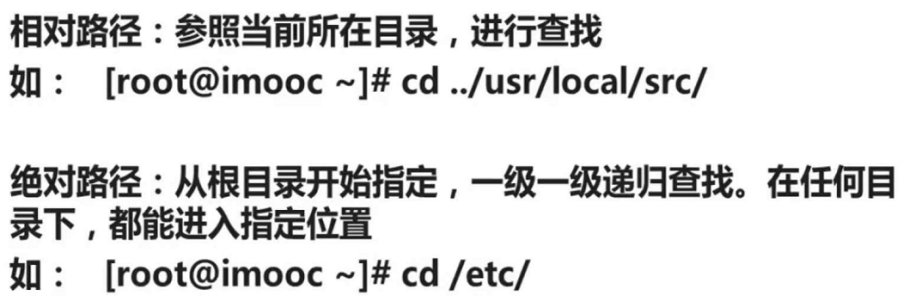
windows下可以cd..，Linux下必须加空格，cd ..

cd /进入根目录，pwd:显示当前目录。











注意上图中 cd /tmp和cd tmp的区别。

/：根目录

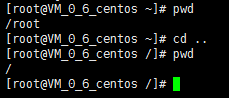
一般根目录下只存放目录，在Linux下有且只有一个根目录。所有的东西都是从这里开始。当你在终端里输入“/home”，你其实是在告诉电脑，先从/（根目录）开始，再进入到home目录。

用户登录后默认是家目录，因为root用户的家目录是根目录下的root用户，因此root用户登录后，默认是root目录。

cd ~ 进入家目录

cd /进入根目录。

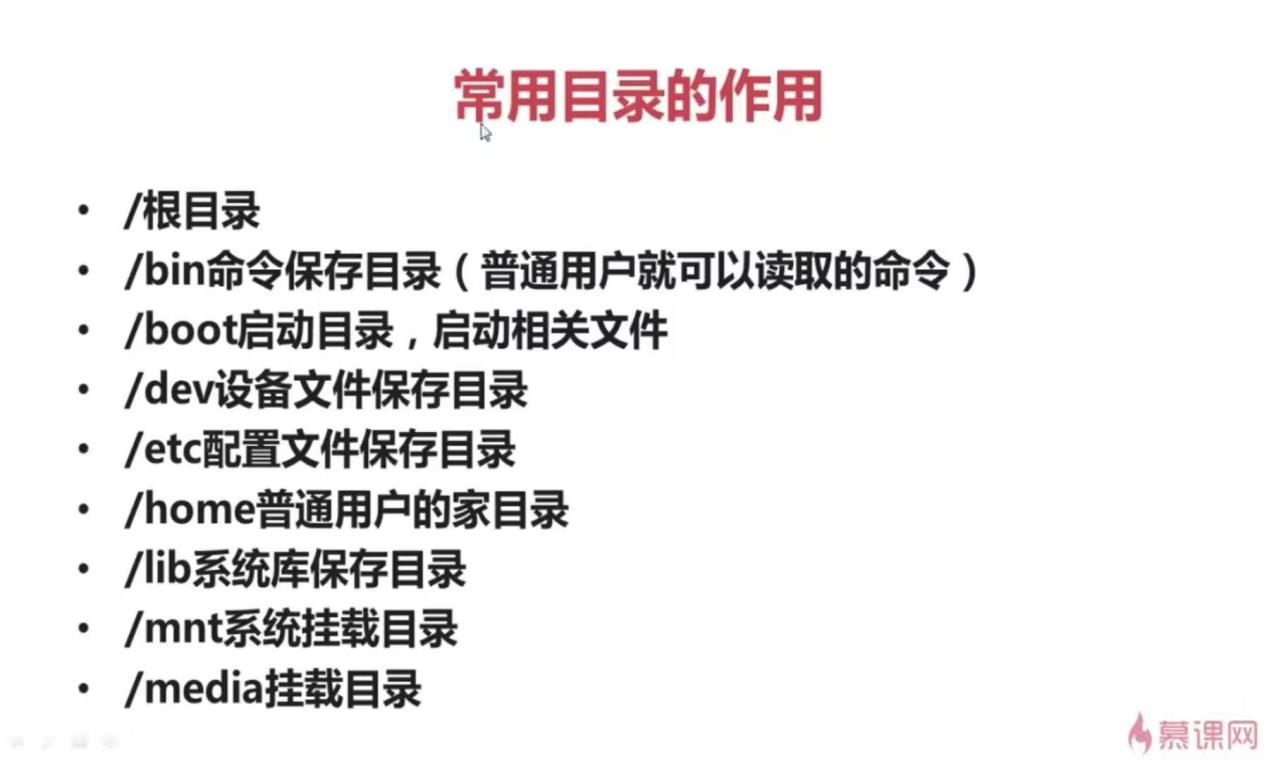
查看当前目录pwd



./

当前路径。



****

默认的目录





/home

系统默认的用户家目录，新增用户账号时，用户的家目录都存放在此目录下，~表示当前用户的家目录，~edu 表示用户 edu 的家目录。

/etc

系统配置文件存放的目录，不建议在此目录下存放可执行文件。/etc 目录存放着各种系统配置文件，其中包括了用户信息文件 /etc/passwd ，系统初始化文件 /etc/rc 等。 linux 正是靠这些文件才得以正常地运行。类似windows下的[c](https://www.baidu.com/s?wd=cc&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9uW-hPWI9n1-buhDzuyRz0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPj0znH03rj0L):\WINDOWS目录。

重要的配置文件有 /etc/inittab、/etc/fstab、/etc/init.d、/etc/X11、/etc/sysconfig、/etc/xinetd.d。

/bin、/usr/bin

可执行二进制文件的目录，如常用的命令ls、tar、mv、cat等。

/boot

放置linux系统启动时用到的一些文件，如Linux的内核文件：/boot/vmlinuz，系统引导管理器：/boot/grub。

/dev

存放linux系统下的设备文件，访问该目录下某个文件，相当于访问某个设备，常用的是挂载光驱 mount /dev/cdrom /mnt。

/lib、/usr/lib、/usr/local/lib

系统使用的函数库的目录，这个目录里存放着系统最基本的动态连接共享库，其作用类似于Windows里的DLL文件。程序在执行过程中，需要调用一些额外的参数时需要函数库的协助。

/lost+fount

系统异常产生错误时，会将一些遗失的片段放置于此目录下。

/mnt: /media

光盘默认挂载点，通常光盘挂载于 /mnt/cdrom 下，也不一定，可以选择任意位置进行挂载。

所有的存储设备都要挂载运行。

这三个目录是作为外接设备的目录。

/opt

给主机额外安装软件所摆放的目录。

/proc

此目录的数据都在内存中，如系统核心，外部设备，网络状态，由于数据都存放于内存中，所以不占用磁盘空间，比较重要的目录有 /proc/cpuinfo、/proc/interrupts、/proc/dma、/proc/ioports、/proc/net/\* 等。

/root

系统管理员root的家目录。

/sbin、/usr/sbin、/usr/local/sbin

放置系统管理员使用的可执行命令，如fdisk、shutdown、mount 等。与 /bin 不同的是，这几个目录是给系统管理员 root使用的命令，一般用户只能"查看"而不能设置和使用。

/tmp

一般用户或正在执行的程序临时存放文件的目录，任何人都可以访问，重要数据不可放置在此目录下。

/srv

服务启动之后需要访问的数据目录，如 www 服务需要访问的网页数据存放在 /srv/www 内。

/usr

应用程序存放目录，/usr/bin 存放应用程序，/usr/share 存放共享数据，/usr/lib 存放不能直接运行的，却是许多程序运行所必需的一些函数库文件。/usr/local: 存放软件升级包。/usr/share/doc

系统说明文件存放目录。/usr/share/man: 程序说明文件存放目录。

/var

放置系统执行过程中经常变化的文件，如随时更改的日志文件 /var/log，/var/log/message：所有的登录文件存放目录，/var/spool/mail：邮件存放的目录，/var/run:程序或服务启动后，其PID存放在该目录下。

用户目录

位于/home/user，称之为用户工作目录或家目录,表示方式：

/home/user

绝对路径

从/目录开始描述的路径为绝对路径，如：

cd /home

ls /usr

相对路径

从当前位置开始描述的路径为相对路径，如：

cd ../../

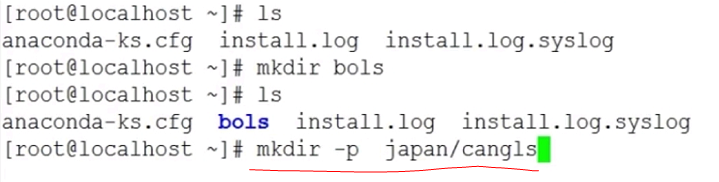
ls abc/def

.和..

每个目录下都有.和..，. 表示当前目录，.. 表示上一级目录，即父目录。根目录下的.和..都表示当前目录。

建立目录





mkdir -p的意思是本来没有japan，也没有cangls目录，-p了后，两个目录都创建好了。

目录改名

mv apache-tomcat-8.0.29 tomcat

其实mv的意思应该是移动，这个操作变成了将文件夹“apache-tomcat-8.0.29”改为了“tomcat”。

mv命令

移动文件或修改文件名，根据第二参数类型（如目录，则移动文件；如为文件则重命令该文件）。

当第二个参数为目录时，可刚多个文件以空格分隔作为第一参数，移动多个文件到参数2指定的目录中

实例：

（1）将文件test.log重命名为test1.txt

mv test.log test1.txt

（2）将文件log1.txt,log2.txt,log3.txt移动到根的test3目录中

mv llog1.txt log2.txt log3.txt /test3

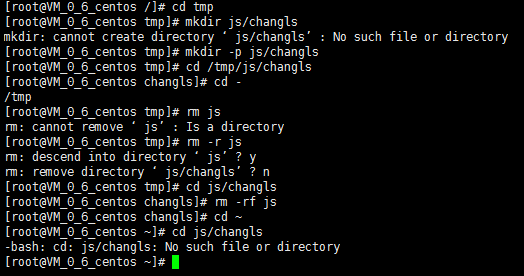
（3）将文件file1改名为file2，如果file2已经存在，则询问是否覆盖

mv -i log1.txt log2.txt

（4）移动当前文件夹下的所有文件到上一级目录

mv \* ../

目录删除



rm 删除一个目录中的一个或多个文件或目录，如果没有使用- r选项，则rm不会删除目录。如果使用rm 来删除文件，通常仍可以将该文件恢复原状。

rm -r可以删除目录

rm -rf强制删除目录

如果不用强制删除目录，目录下的文件很多的话，就会一条条让你确认删除，此时ctrl+c退出，重新输入 rm -rf 目录名即可。

rm -rf /是强制删除根目录，造成操作系统崩溃。

命令提示

输入字母u后按两遍tab键，则会将u开头的命令显示在界面。

远程登录

Linux系统中是通过ssh服务实现的远程登录功能，默认ssh服务端口号为 22。

字符界面清屏

Ctrl+L

文件权限

文件权限就是文件的访问控制权限，即哪些用户和组群可以访问文件以及可以执行什么样的操作。

Unix/Linux系统是一个典型的多用户系统，不同的用户处于不同的地位，对文件和目录有不同的访问权限。为了保护系统的安全性，Unix/Linux系统除了对用户权限作了严格的界定外，还在用户身份认证、访问控制、传输安全、文件读写权限等方面作了周密的控制。

在 Unix/Linux中的每一个文件或目录都包含有访问权限，这些访问权限决定了谁能访问和如何访问这些文件和目录。

访问用户

通过设定权限可以从以下三种访问方式限制访问权限：

只允许用户自己访问（所有者） 所有者就是创建文件的用户，用户是所有用户所创建文件的所有者，用户可以允许所在的用户组能访问用户的文件。

允许一个预先指定的用户组中的用户访问（用户组） 用户都组合成用户组，例如，某一类或某一项目中的所有用户都能够被系统管理员归为一个用户组，一个用户能够授予所在用户组的其他成员的文件访问权限。

允许系统中的任何用户访问（其他用户） 用户也将自己的文件向系统内的所有用户开放，在这种情况下，系统内的所有用户都能够访问用户的目录或文件。在这种意义上，系统内的其他所有用户就是 other 用户类。

这有点类似于 QQ 空间的访问权限：这个 QQ 空间是属于我的，我相当于管理者（也就是“所有者”），我想怎么访问就怎么访问。同时，我可以设置允许 QQ 好友访问，而这些 QQ 好友则类似于“用户组”。 当然，我可以允许所有人访问，这里的所有人则类似于“其他用户”。

访问权限

用户能够控制一个给定的文件或目录的访问程度，一个文件或目录可能有读、写及执行权限：

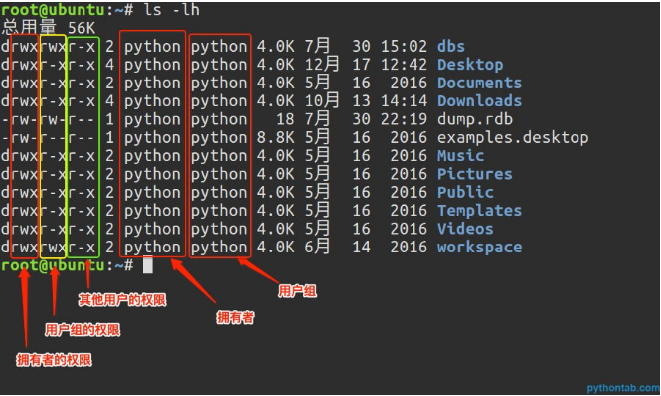
读权限（r） 对文件而言，具有读取文件内容的权限；对目录来说，具有浏览目录的权限。

写权限（w） 对文件而言，具有新增、修改文件内容的权限；对目录来说，具有删除、移动目录内文件的权限。

可执行权限（x） 对文件而言，具有执行文件的权限；对目录了来说该用户具有进入目录的权限。

注意：通常，Unix/Linux系统只允许文件的属主(所有者)或超级用户改变文件的读写权限。

示例说明：



第1个字母代表文件的类型：“d” 代表文件夹、“-” 代表普通文件、“c” 代表硬件字符设备、“b” 代表硬件块设备、“s”表示管道文件、“l” 代表软链接文件。 后 9 个字母分别代表三组权限：文件所有者、用户者、其他用户拥有的权限。

每一个用户都有它自身的读、写和执行权限。

第一组权限控制访问自己的文件权限，即所有者权限。

第二组权限控制用户组访问其中一个用户的文件的权限。

第三组权限控制其他所有用户访问一个用户的文件的权限。

这三组权限赋予用户不同类型（即所有者、用户组和其他用户）的读、写及执行权限就构成了一个有9种类型的权限组。

# 三、文本

文本编辑：vim

如提示command not found说明vim没安装。

输入rpm -qa|grep vim 命令, 如果 vim 已经正确安裝,会返回下面的三行代码:

root@server1 [~]# rpm -qa|grep vim

vim-enhanced-7.0.109-7.el5

vim-minimal-7.0.109-7.el5

vim-common-7.0.109-7.el5

如果少了其中的某一条,比如 vim-enhanced 的,就用命令 yum -y install vim-enhanced 来安裝:

yum -y install vim-enhanced

如果上面的三条一条都沒有返回, 可以直接用 yum -y install vim\* 命令

yum -y install vim\*

文本编辑：vi

可以用vi代替vim进行编辑：# vi 文件名

如：# vi ifcfg-eth0

a.进入后，键盘上下左右键移动。

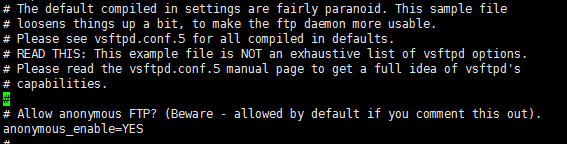
b.需要删除，移动光标到需要删除的地方，通过键盘上的x键。

c.需要增加/编辑，移动光标到需要插入的地方，通过键盘上的i键，此时右下角会出现insert字样，直接输入需要插入的文字即可，按Backspace键可以删除文字。

具体操作可参见：记事宝CentOS中的《CentOS6.4之文本编辑器Vi/Vim》。

d.复制粘贴：

1）把光标移动到要复制的行上  
2）按yy  
3）把光标移动到要复制的位置  
4）按p



会复制到高亮显示的下一行。

e.退出vi

按ESC，然后输入:wq 意思是保存并退出vi。

其他具体的命令如下：

:w 保存文件但不退出vi

:w file 将修改另外保存到file中，不退出vi

:w! 强制保存，不推出vi

:wq 保存文件并退出vi

:wq! 强制保存文件，并退出vi

:q 不保存文件，退出vi

:q! 不保存文件，强制退出vi

:e! 放弃所有修改，从上次保存文件开始再编辑

f.搜索

在打开的vi中输入/，输入需要搜索的文字，回车即可，如需继续搜索，则按键盘n即可。

安装文本编辑器(UTF-8)gedit

# yum install gedit

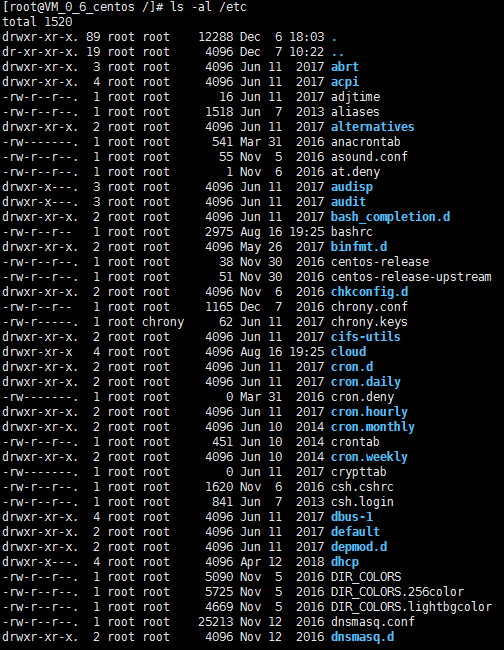
# 四、常用命令

ls

查看目录和文件

命令可叠加，如-a -l可以写成：ls -al /etc

相当于通过ls -al命令查看/etc目录下详情。



处理目录的常用命令

ls: 列出目录；

cd：切换目录；

pwd：显示目前的目录；

mkdir：创建一个新的目录；

rmdir：删除一个空的目录；

cp: 复制文件或目录；

rm: 移除文件或目录；

用 root 身份，将 root 目录下的 .bashrc 复制到 /tmp 下，并命名为 bashrc：

[root@www ~]# cp ~/.bashrc /tmp/bashrc

内容查看

使用以下命令来查看文件的内容：

* cat  由第一行开始显示文件内容
* tac  从最后一行开始显示，可以看出 tac 是 cat 的倒著写！
* nl   显示的时候，顺道输出行号！
* more 一页一页的显示文件内容
* less 与 more 类似，但是比 more 更好的是，他可以往前翻页！
* head 只看头几行
* tail 只看尾巴几行

链接

Linux 链接分两种，一种被称为硬链接（Hard Link），另一种被称为符号链接（Symbolic Link）。默认情况下，ln 命令产生硬链接。

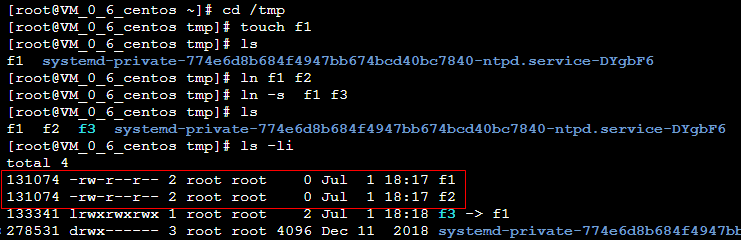
硬连接：

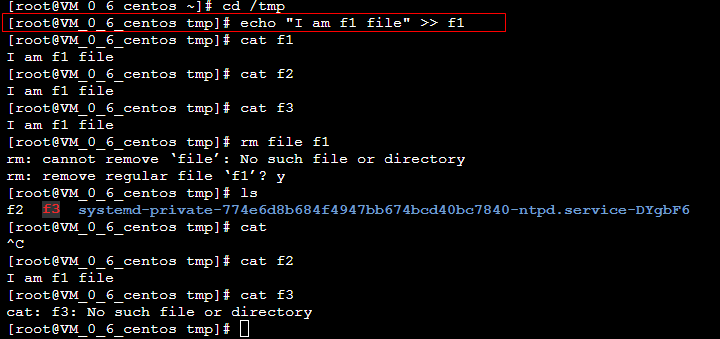
硬连接指通过索引节点来进行连接。在 Linux 的文件系统中，保存在磁盘分区中的文件不管是什么类型都给它分配一个编号，称为索引节点号(Inode Index)。在 Linux 中，多个文件名指向同一索引节点是存在的。比如：A 是 B 的硬链接（A 和 B 都是文件名），则 A 的目录项中的 inode 节点号与 B 的目录项中的 inode 节点号相同，即一个 inode 节点对应两个不同的文件名，两个文件名指向同一个文件，A 和 B 对文件系统来说是完全平等的。删除其中任何一个都不会影响另外一个的访问。

硬连接的作用是允许一个文件拥有多个有效路径名，这样用户就可以建立硬连接到重要文件，以防止“误删”的功能。其原因如上所述，因为对应该目录的索引节点有一个以上的连接。只删除一个连接并不影响索引节点本身和其它的连接，只有当最后一个连接被删除后，文件的数据块及目录的连接才会被释放。也就是说，文件真正删除的条件是与之相关的所有硬连接文件均被删除。

软连接：

另外一种连接称之为符号连接（Symbolic Link），也叫软连接。软链接文件有类似于 Windows 的快捷方式。它实际上是一个特殊的文件。在符号连接中，文件实际上是一个文本文件，其中包含的有另一文件的位置信息。比如：A 是 B 的软链接（A 和 B 都是文件名），A 的目录项中的 inode 节点号与 B 的目录项中的 inode 节点号不相同，A 和 B 指向的是两个不同的 inode，继而指向两块不同的数据块。但是 A 的数据块中存放的只是 B 的路径名（可以根据这个找到 B 的目录项）。A 和 B 之间是“主从”关系，如果 B 被删除了，A 仍然存在（因为两个是不同的文件），但指向的是一个无效的链接。





可以得到以下全部结论：

* 1).删除符号连接f3,对f1,f2无影响；
* 2).删除硬连接f2，对f1,f3也无影响；
* 3).删除原文件f1，对硬连接f2没有影响，导致符号连接f3失效；
* 4).同时删除原文件f1,硬连接f2，整个文件会真正的被删除。

touch

生成一个空文件或更改文件的时间。

echo

把内容重定向到指定的文件中 ，有则打开，无则创建。

重定向 > 是覆盖模式，>> 是追加模式，例如：echo "Java3y,zhen de hen xihuan ni" > qingshu.txt把左边的输出放到右边的文件里去。

cat

查看文本文件内容。

find

查找文件。

find -name xx.xlsx 搜索文件名为xx.xlsx的文件

tail -10

查看文件的尾部的10行。

head -20

查看文件的头部20行。

压缩

压缩的方式也是有好几种，我们常用的有下面这三种：

gzip

bzip2

tar

常用的压缩的命令就有：

gzip filename

bzip2 filename

tar -czvf filename

常用的解压命令有：

gzip -d filename.gz

bzip2 -d filename.bz2

tar -xzvf filename.tar.gz

# 五、系统配置

查询系统负载

[root@localhost ~]# uptime

10:59:01 up 14 days, 20 min, 2 users, load average: 0.04, 0.04, 0.05

[root@localhost ~]# df -h

文件系统 容量 已用 可用 已用% 挂载点

/dev/mapper/cl-root 48G 25G 23G 52% /

devtmpfs 7.8G 0 7.8G 0% /dev

tmpfs 7.8G 0 7.8G 0% /dev/shm

tmpfs 7.8G 226M 7.6G 3% /run

tmpfs 7.8G 0 7.8G 0% /sys/fs/cgroup

/dev/vda1 1014M 223M 792M 22% /boot

/dev/mapper/cl-home 24G 2.6G 21G 11% /home

tmpfs 1.6G 12K 1.6G 1% /run/user/42

tmpfs 1.6G 0 1.6G 0% /run/user/0

[root@localhost ~]# free

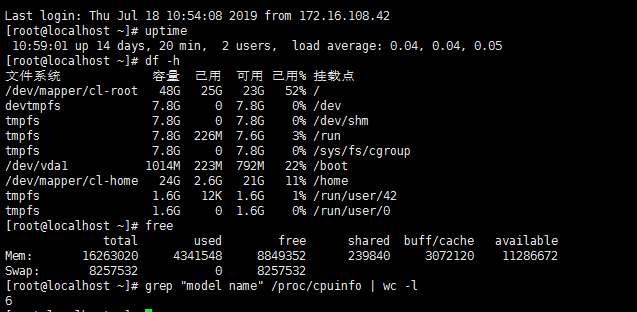
total used free shared buff/cache available

Mem: 16263020 4341548 8849352 239840 3072120 11286672

Swap: 8257532 0 8257532

[root@localhost ~]# grep "model name" /proc/cpuinfo | wc -l

6

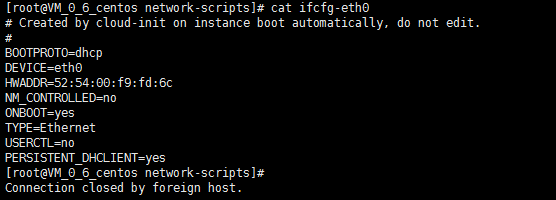


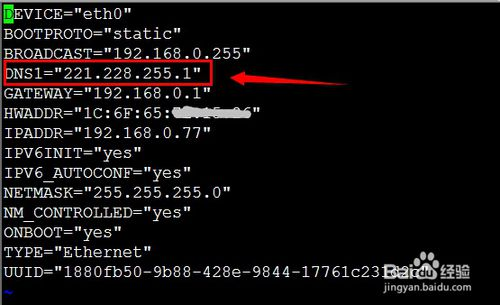
ip地址配置

进入网络配置目录：cd  /etc/sysconfig/network-scripts/



查看网卡配置：cat ifcfg-eth0





BOOTPROTO：采用static或者dhcp或者none。

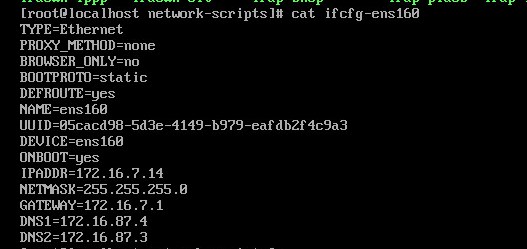
其他就是IP地址, 以及网关, 掩码, DNS信息。

操作完千万记得要重启网络。

centos 6:service network restart

或

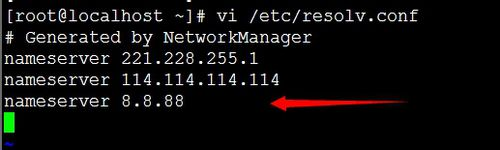
centos 7:systemctl restart network



DNS服务器地址配置

在Linux下面，有一个默认的DNS服务器地址配置文件的设置，存放在 /etc/resolv.conf

设置方法很简单，通过编辑 vi /etc/resolv.conf 设置首选DNS和次要DNS。如下图，排在前面的就是首选DNS，后面一行就是次要的DNS服务器DNS



hosts主机表文件

跟Windows系统一样，Linux系统也有主机表文件。同样可以通过设置主机表地址进行特定主机的解析，优先DNS服务器地址。

配置文件，在/etc/hosts



修改网络配置

参见“技术-系统-Linux”中《VirtualBox linux虚拟机如何实现“桥接”上网(已验证)》。

此时ping www.baidu.com就可以ping通了。

注：ping 不加其它参数的话，会一直运行了相当于Windows的ping –t

可以按Ctrl+c中断

也可以用-c参数指定要求回应次数，如：

ping -c 4 [www.163.com](http://www.163.com/)

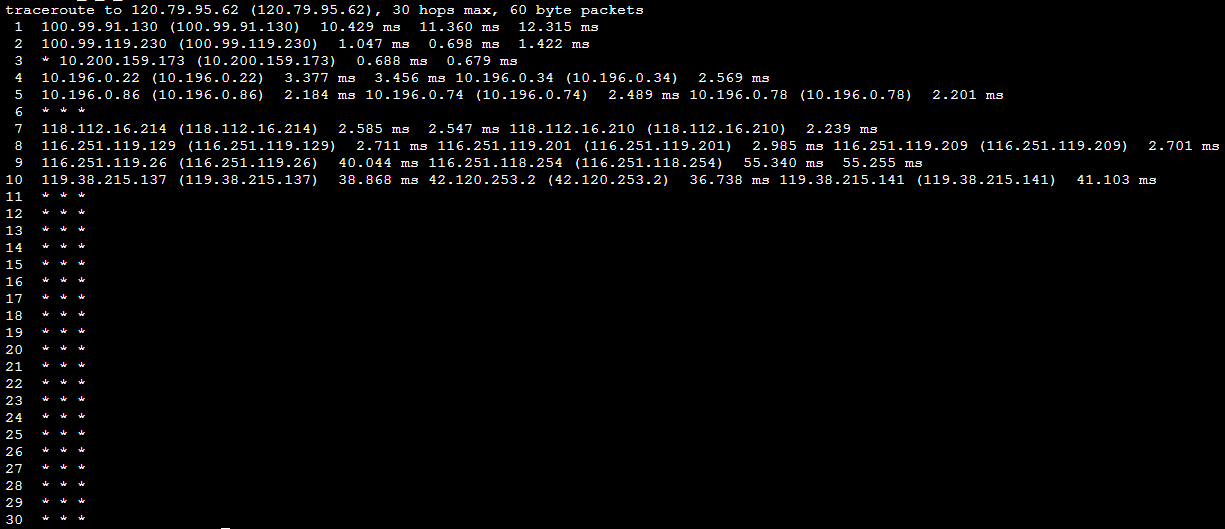
查看本机ip地址可通过：# ifconfig -a

重启网络服务

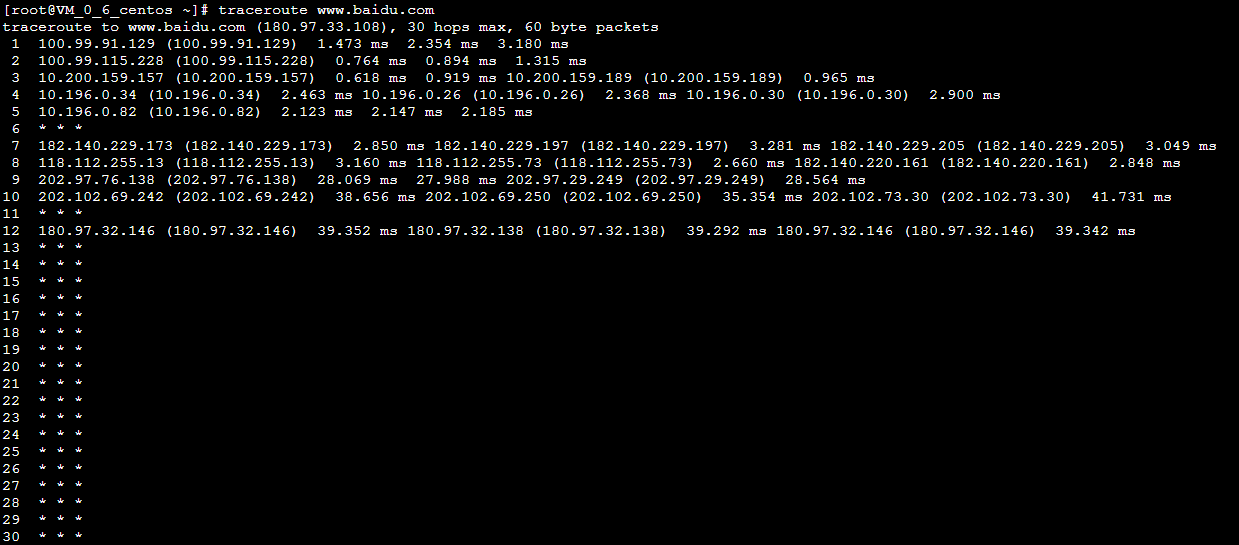
# service network restart

tracerout

测试与其它主机的网络链接路径



也可以：



和Linux之间通过ftp交互

可通过Xmanager Enterprise中的Xftp链接,Linux服务器端不需要执行任何操作。貌似服务器端不需要开图形界面，使用Xstart也是基于图形界面的。

具体使用可参见“技术-系统-Linux-CentOS”中《使用Xftp连接Centos服务器(已验证)》。

字符集

在终端输入:locale，查看是否是中文编码：

LANG=zh\_CN.UTF-8

LC\_CTYPE="zh\_CN.UTF-8"

LC\_NUMERIC="zh\_CN.UTF-8"

LC\_TIME="zh\_CN.UTF-8"

LC\_COLLATE="zh\_CN.UTF-8"

LC\_MONETARY="zh\_CN.UTF-8"

LC\_MESSAGES="zh\_CN.UTF-8"

LC\_PAPER="zh\_CN.UTF-8"

LC\_NAME="zh\_CN.UTF-8"

LC\_ADDRESS="zh\_CN.UTF-8"

LC\_TELEPHONE="zh\_CN.UTF-8"

LC\_MEASUREMENT="zh\_CN.UTF-8"

LC\_IDENTIFICATION="zh\_CN.UTF-8"

LC\_ALL=

需要中文，将/etc/locale.conf文件修改：

vi /etc/locale.conf //编辑locale.cnf

i //进入命令行模式

---

//把内容修改成如下

LANG="zh\_CN.UTF-8"

LANGUAGE="zh\_CN.zh\_CN.UTF-8:zh\_CN.zh\_CN.UTF-8:zh\_CN"

SUPPORTED="zh\_CN.zh\_CN.UTF-8:zh\_CN:zh"

SYSFONT="latarcyrheb-sun16"

---

esc //进入命令行模式

:wq //保存退出

重新启动服务器

yum

Yum（全称为 Yellow dog Updater, Modified）是一个在Fedora和RedHat以及CentOS中的[Shell](https://baike.baidu.com/item/Shell)前端软件包管理器。基于[RPM](https://baike.baidu.com/item/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，无须繁琐地一次次下载、安装。

yum安装

yum -y install vsftpd

curl-测试网址是否可访问

网址是否可访问：

curl <http://weixin.jsptpd.com:8081/WeixinCp/queryOPperson?weiling=506725>

如果是POST:

curl https://weixin.jsptpd.com/WeixinCp/Token -X POST -d "op=get\_userid&code=587108"

windows下：

<https://www.cnblogs.com/zhuzhenwei918/p/6781314.html>

curl https://m.cicdi.com/WeixinCp/Token -k -X POST -d "op=get\_userid&code=904885"

https记得加-k

wget

wget是一个从网络上自动下载文件的自由工具，支持通过HTTP、HTTPS、FTP三个最常见的TCP/IP协议下载，并可以使用HTTP代理。

# yum -y install wget

安装Nginx

可参见“技术-系统-Linux-CentOS”中《Linux（CentOS）下，下载安装Nginx并配置(已验证)》。

如果本地电脑无法访问虚拟机里nginx发布的站点，且ip能够ping通，多半是因为防火墙的问题，请参见“技术-Nginx”中《[nginx开启后主机无法访问虚拟机的nginx解决方案](http://www.cnblogs.com/yomho/p/6074815.html)》

具体内容也可参见Nginx知识体系。

简单地说，有如下几步：

1、安装编译安装所需要的依赖

yum install -y gcc-c++ pcre pcre-devel zlib zlib-devel openssl openssl-devel

2、下载Nginx

yum install -y wget

wget <https://nginx.org/download/nginx-1.14.0.tar.gz>

在哪个目录下执行wget，就下载在哪个目录。

3、解压

tar zxf nginx-1.14.0.tar.gz

cd nginx-1.14.0

4、编译安装（默认设置）

./configure

make && make install

5、查看安装目录

whereis nginx

nginx: /usr/local/nginx

6、启动nginx

cd /usr/local/nginx/sbin

./nginx

7、查看nginx是否运行

ps -ef | grep -v grep | grep nginx

root 4772 1 0 18:27 ? 00:00:00 nginx: master process ./nginx

nobody 4773 4772 0 18:27 ? 00:00:00 nginx: worker process

8、启动nginx并设置开机启动

systemctl start nginx

systemctl enable nginx

# 启动nginx出现Failed to start nginx.service:unit not found

<https://blog.csdn.net/qq_17054659/article/details/77186249>

为了方便，将nginx服务添加至systemctl:

环境变量

查看环境变量

echo $PATH



修改环境变量

vim /etc/environment

在PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games"中加入“:/usr/local/mongodb/bin”

然后:wq

#vi ~/.bash\_profile

此时，只要在export PATH前面加入自己的路径即可，例如我的修改后如下：

[www.linuxidc.com @localhost bin]$ cat ~/.bash\_profile  
# .bash\_profile

# Get the aliases and functions

if [ -f ~/.bashrc ]; then

        . ~/.bashrc

Fi

# User specific environment and startup programs

PATH=$PATH:/usr/local/programs/crosstool/gcc-4.0.2-glibc-2.3.6/arm-linux-gnu/bin/

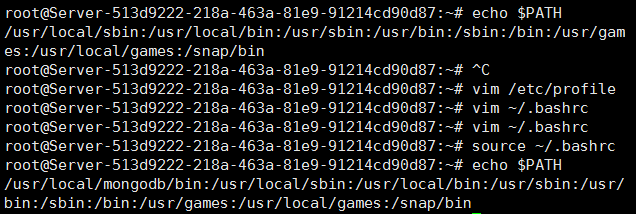
export PATH

注：PATH 声明，其格式为：

　　PATH=$PATH:<PATH 1>:<PATH 2>:<PATH 3>:------:<PATH N>

你可以自己加上指定的路径，中间用冒号隔开。环境变量更改后，在用户下次登陆时生效，如果想立刻生效，则可执行下面的语句：$ source .bash\_profile

以下是经过验证的：



vim ~/.bashrc

//在最后一行添上：

export PATH=/usr/local/mongodb/bin:$PATH

Esc,:wq保存并退出

输入“source ~/.bashrc”命令，立即生效

如上图所示，查看环境变量，变动就加入进去了。

安装Java环境

http://download.[Oracle](http://www.linuxidc.com/topicnews.aspx?tid=12).com/

使用上述网址下载jre。

jre和jdk的区别：jre(java runtime environment),就是java程序的运行环境。你所写的java程序在经过javac编译后会形成字符行文件，这个java的中间文件，计算机是看不懂的，而这个文件是给jvm（java的虚拟机）用的，由jvm进行解释后形成[机器语言](https://www.baidu.com/s?wd=%E6%9C%BA%E5%99%A8%E8%AF%AD%E8%A8%80&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dWuAnYPjn4nh7BnHT3rAD40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWfYP10snWDd)给计算机。而这个jvm就在jre中，也就是说java程序的运行要有jre来担当。所以jdk和jre的区别就是一个用于开发，一个用于运行，联系就是他们都是java开发者所必备的（除非你就像运行java程序，那么用jre就足够了）。

# java –version 查看系统中java版本

参照上方#vi ~/.bash\_profile，设置环境变量：

  ########## JDK #############

export JAVA\_HOME=/usr/java/jdk1.8.0\_25

export PATH=$PATH:$JAVA\_HOME/

export CLASSPATH=.:$JAVA\_HOME/lib/tools.jar:$JAVA\_HOME/lib/dt.jar

########## Jre #############

PATH=$PATH:/usr/java/jre1.8.0\_25/bin

export JAVA\_HOME=/usr/java/jre1.8.0\_25

export CLASSPATH=$JAVA\_HOME/lib

：wq 保存

重启linux完成安装。运行 Java -version，测试jre是否安装成功。

参见“技术-系统-Linux-CentOS”中《CentOS下jdk和jre的安装与配置(已验证)》。

安装tomcat

<http://tomcat.apache.org/download-80.cgi>

参见“技术-系统-Linux-CentOS”中《Linux(Centos)之安装tomcat并且部署Java Web项目(已验证)》。

修改网络配置

参见“技术-系统-Linux”中《VirtualBox linux虚拟机如何实现“桥接”上网(已验证)》。

防火墙

首先查看防火墙状态：

firewall-cmd –state

开启防火墙：

systemctl start firewalld.service

设置开机自启：

systemctl enable firewalld.service

重启防火墙：

systemctl restart firewalld.service

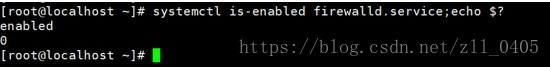
修改配置后需要重新加载才能生效：

firewall-cmd –reload

查看防火墙设置开机自启是否成功：

systemctl is-enabled firewalld.service;echo $?

如图所示，即为成功：



firewall-cmd --add-port=80/tcp --permanent     ##永久添加80端口

firewall-cmd --zone=public --remove-port=80/tcp --permanent            //关闭端口

firewall-cmd --reload                                                                   //重新载入，使其生效

查看所有开启的端口：

firewall-cmd --zone=public --list-ports

更多命令，使用  firewall-cmd --help 查看帮助文件：

<https://blog.csdn.net/zll_0405/article/details/81208606>

<https://blog.csdn.net/wangchaox123/article/details/88805182>

修改网络配置

参见“技术

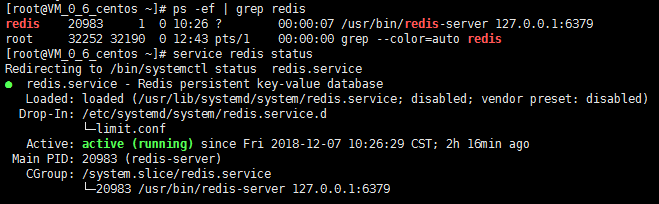
安装redis

yum install redis

yum install epel-release

安装完毕后，使用下面的命令启动redis服务

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | # 启动redis  service redis start  # 停止redis  service redis stop  # 查看redis运行状态  service redis status  # 查看redis进程  ps -ef | grep redis |



5、设置redis为开机自动启动

|  |  |
| --- | --- |
| 1 | chkconfig redis on |

进入redis服务

|  |  |
| --- | --- |
| 1  2  3  4 | # 进入本机redis  redis-cli  # 列出所有key  keys \* |

https://www.cnblogs.com/rslai/p/8249812.html

解压

tar –xvjf nginx-latest-x86\_64.tar.bz2

参见“技术-系统-Linux-CentOS”中《CentOS下tar解压 gz解压 bz2等各种解压文件使用方法》

安装浏览器

(1)firefox

# yum –y install firefox

(2)chrome

参见“技术-系统-Linux-CentOS”中《CentOS 7 下安装Chrome浏览器(已验证)》

重启

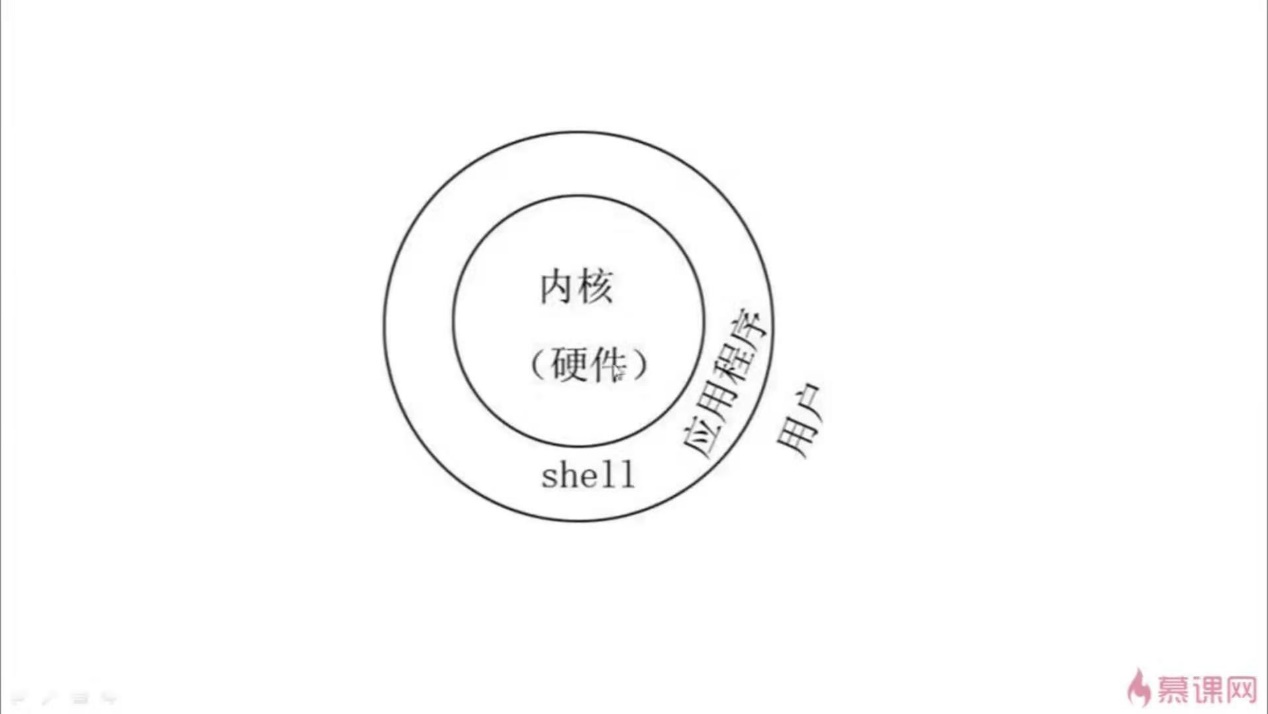
# reboot

关机

# poweroff

注：虚拟机的安装软件如无意外，我都放在了/usr/local/下。

# 六、Shell



shell将用户敲的命令翻译成0101传递给内核。

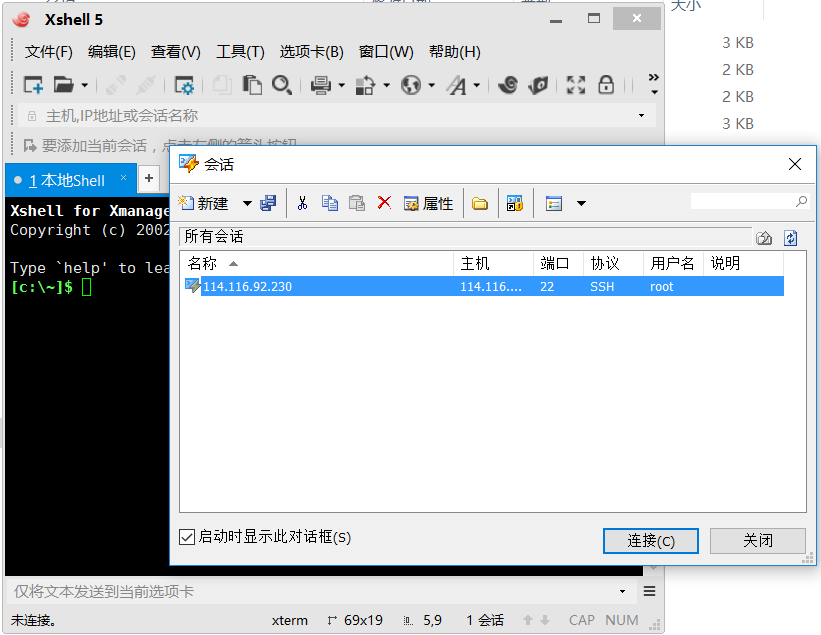
Shell是什么

Shell是一个命令行解释器。Windows的图形界面也是Shell。通俗一点，就是操作界面。

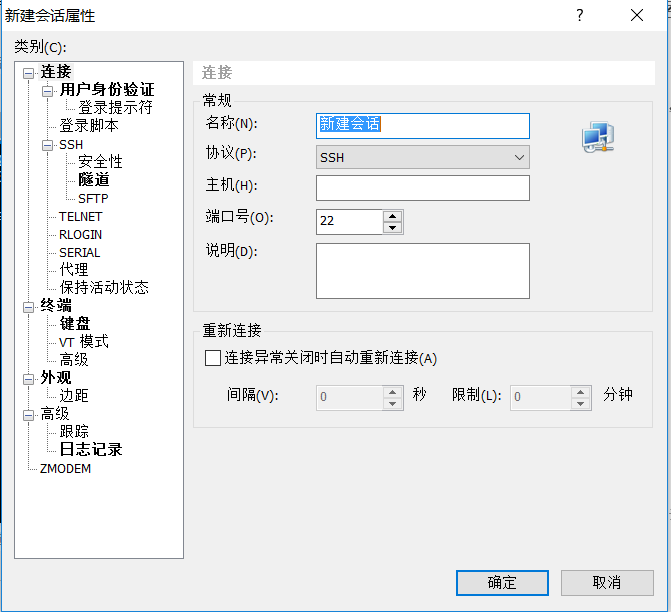
# 七、云服务器

登录

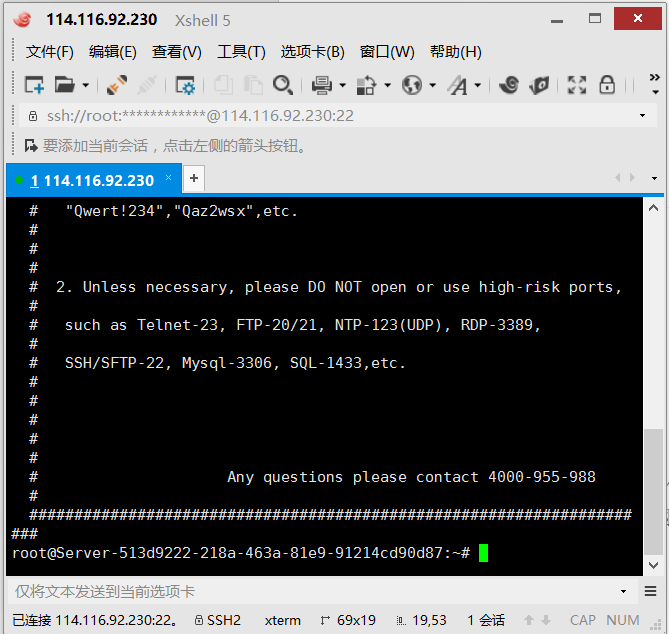
可使用XShell(Xmanager Enterprise 5里有)。



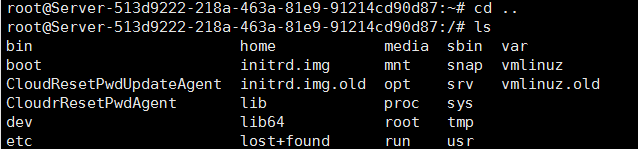
可以点击“新建”：



用户名(root)和密码(云服务器可重置)点“确定”后会让输。

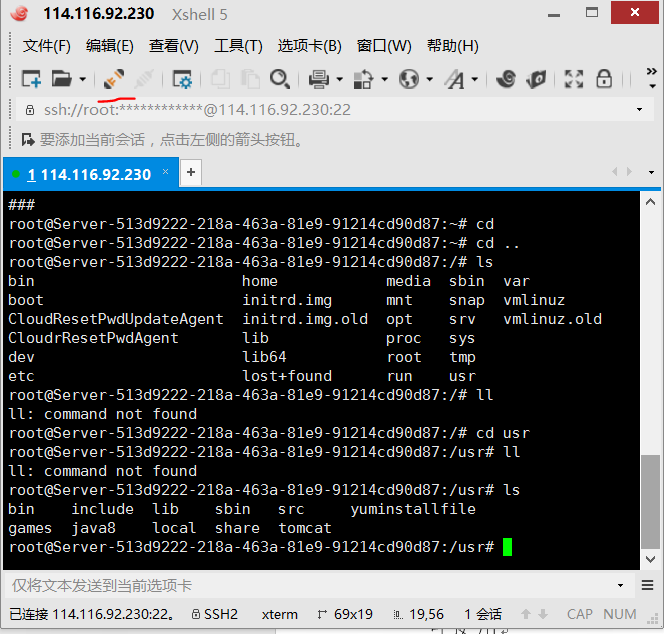


出现上图的界面就算成功了，Server后跟的GUID是云服务器的ID。



连接中断

过段时间不操作，云服务器就会中断连接，点击下图中的按钮即可重新连接：



上传文件到服务器

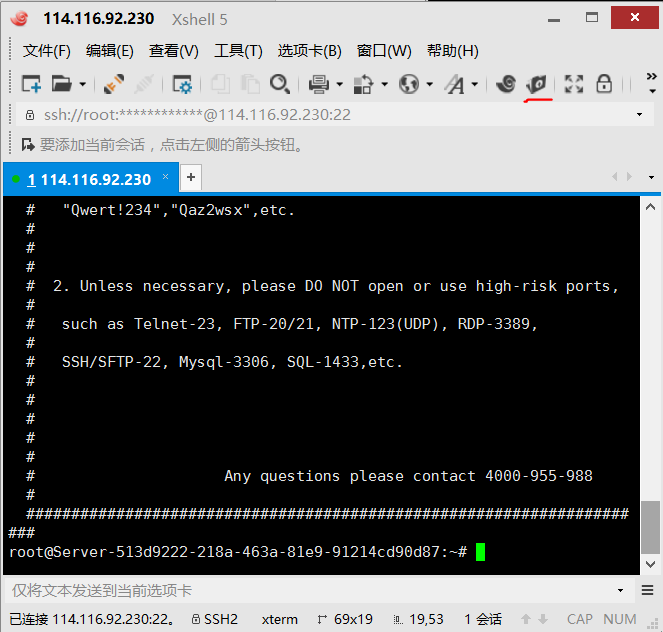
有时需要将文件传输到云服务器上，点击下图中的按钮，就可以点开类似FTP传输的界面。上图画红线那排，从右往左数第3个按钮。

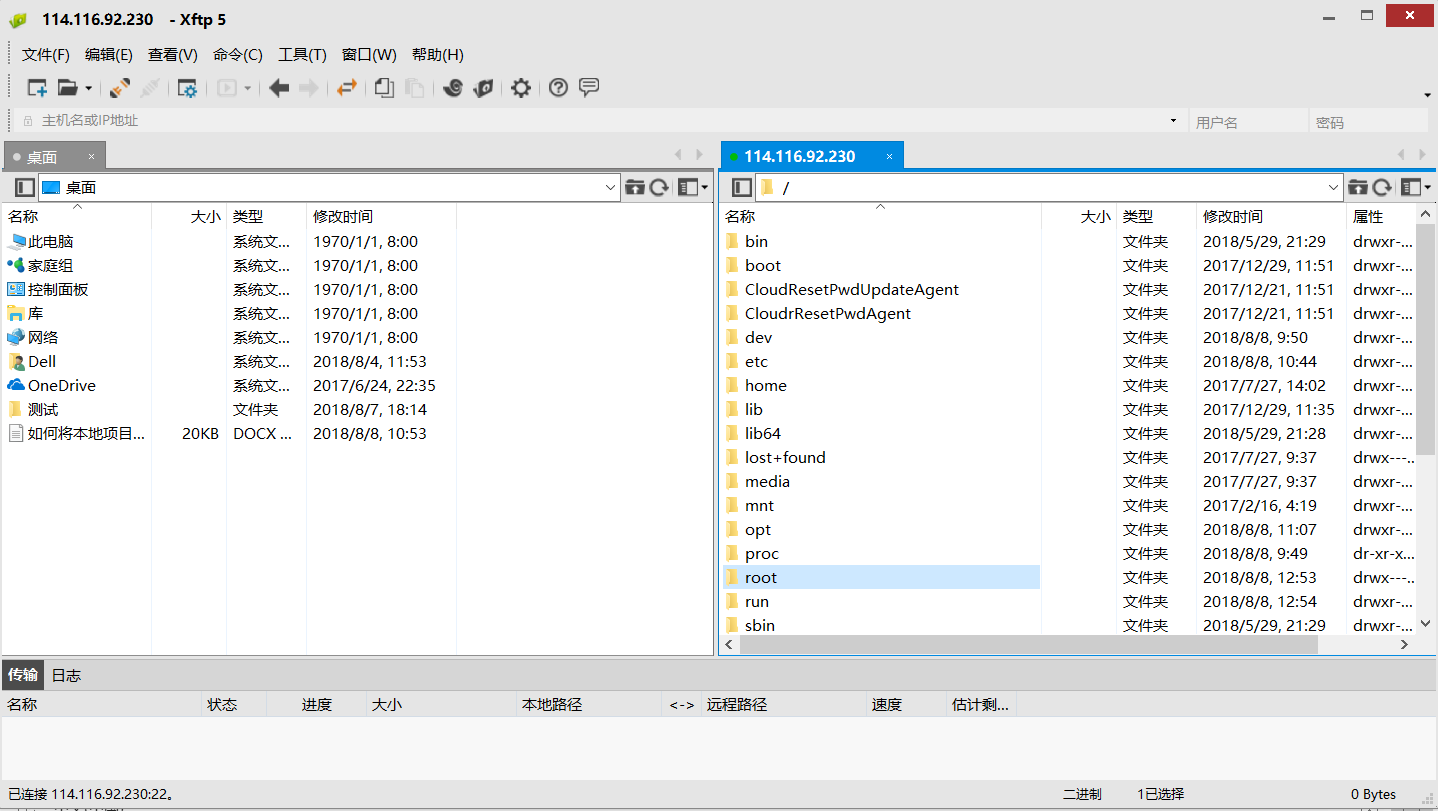
创建文件夹

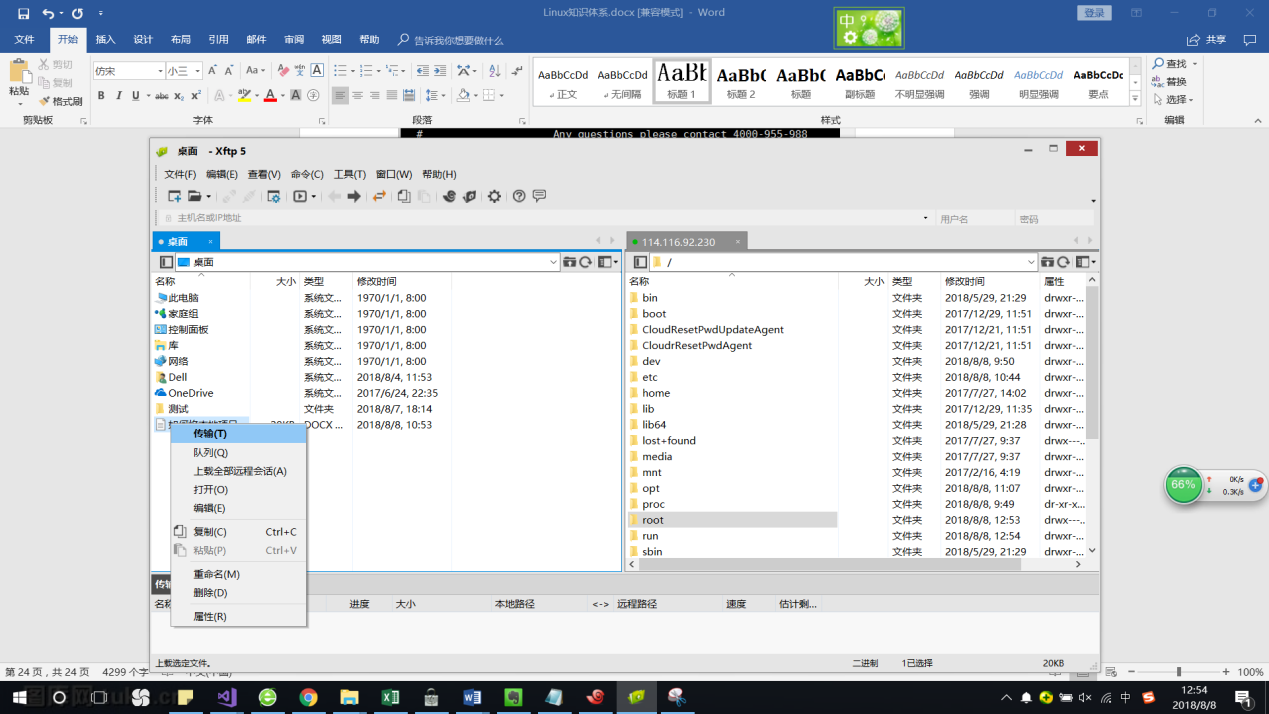
mkdir xx1 在当前目录下创建新文件夹。

安装yum

apt install yum







Ubuntu不该用yum 该用apt-get

安装java环境腾讯云默认没有FTP，安装FTP

查看是否已经安装vsftpd，如果没有就安装，查看命令：rpm -qa | grep vsftpd



# 安装FTP：

sudo yum install vsftpd

vi /etc/vsftpd/vsftpd.conf

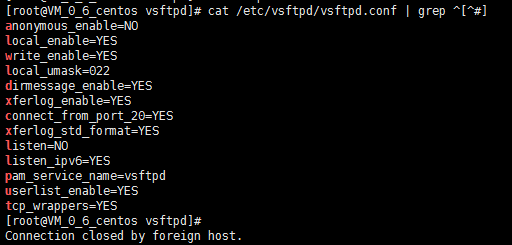
进入到这个文件中，找到如下这几行：anonymous\_enable=YES，将其中的YES换成NO。

我们还需要设置user\_list，将其他所有的敏感账号给注释掉，然后添加自己的username（见下，你自己添加的登陆账号），之后再执行vi /etc/vsftpd/vsftpd.conf，在其中添加一行userlist\_deny=NO，表示不禁用user\_list里的账号。这个时候其中的账号只有你下面的步骤中添加的账号，也就是运行你用自己创建的账号登陆。

改了这三项，其他暂时不需要更改了，我们的目的是要配置好ftp服务，让自己通过设定的ftp账户登陆，上传文件到服务器上

使用如下命令查看是否设置成功：

cat /etc/vsftpd/vsftpd.conf | grep ^[^#]



关闭防火墙

ftp默认的端口为21，而centos默认是没有开启的。所以需要修改防火墙文件，添加开启21端口配置。本文以直接关闭防火墙为例。命令如下：

systemctl stop firewalld.service #停止firewall

systemctl disable firewalld.service #禁止firewall开机启动

firewall-cmd --state



安装java环境

sudo apt-get install default-jre

sudo apt-get install default-jdk

此时输入java -version就可以查询当前的jdk版本了。

安装tomcat环境

将下载的.tar.gz格式的文件下载到电脑。

tar zxvf apache-tomcat-8.5.32.tar.gz //解压文件

sudo mv apache-tomcat-8.5.32/ /opt/tomcat-8.5.32 //移动文件夹

/opt/tomcat-8.5.32/bin/startup.sh //启动成功。

参见“技术-系统-Linux”

# 五、常见错误

# swap file “\*.swp”already exists！

说明在 Linux 下的 vim 编辑过程中，由于某种原因异常退出正在编辑的文件，再次编辑该文件时，会出现这样的提示。

使用vim编辑文件实际是先 copy 一份临时文件并映射到内存给你编辑， 编辑的是临时文件， 当执行：w 后才保存临时文件到原文件，执行：q 后才删除临时文件。

可以通过rm .proxy.conf.swp删除的方式解决。

****

# SFTP 子系统申请已拒绝 请确保ssh连接的sftp子系统设置有效

vi /etc/ssh/sshd\_config将 /etc/ssh/sshd\_config 中的

Subsystem      sftp    /usr/libexec/openssh/sftp-server

改为（没有这一行，就新添加）  
Subsystem       sftp    internal-sftp

重启sshd后，sftp正常工作了

sudo  /etc/init.d/sshd reload  
sudo  /etc/init.d/sshd restart

或：

service sshd restart