|  |
| --- |
|  |
| Linux学习笔记 |
|  |

|  |
| --- |
| 姚斯元  2019-4-11 |

目录

[一、 Linux的简介 1](#_Toc5873072)

[1.1 Linux的概述 1](#_Toc5873073)

[1.2 Linux的分类 1](#_Toc5873074)

[1.2.1 Linux根据市场需求不同，基本分为两个方向： 2](#_Toc5873075)

[1.2.2 Linux根据原生程度，又分为两种： 2](#_Toc5873076)

[1.2.3 Linux发行版本不同，又可以分为n多种： 2](#_Toc5873077)

[二、 Linux的安装 2](#_Toc5873078)

[2.1 虚拟机的安装 2](#_Toc5873079)

[2.2 CentOS的安装 3](#_Toc5873080)

[三、 Linux的基本命令 4](#_Toc5873081)

[3.1 目录切换命令 4](#_Toc5873082)

[3.2 目录的操作命令（增删改查） 4](#_Toc5873083)

[3.2.1 增加目录操作（增） 4](#_Toc5873084)

[3.2.2 查看目录（查） 5](#_Toc5873085)

[3.2.3 寻找目录（查） 5](#_Toc5873086)

[3.2.4 修改目录的名称（改） 6](#_Toc5873087)

[3.2.5 移动目录的位置---剪切（改） 6](#_Toc5873088)

[3.2.6 拷贝目录（改） 6](#_Toc5873089)

[3.2.7 删除目录（删） 7](#_Toc5873090)

[3.3 文件的操作命令（增删改查） 7](#_Toc5873091)

[3.3.1 文件的创建（增） 7](#_Toc5873092)

[3.3.2 文件的查看（查） 8](#_Toc5873093)

[3.3.3 修改文件的内容（改） 10](#_Toc5873094)

[3.3.4 删除文件（删） 12](#_Toc5873095)

[3.4 压缩文件的操作命令 12](#_Toc5873096)

[3.4.1 打包并压缩文件 12](#_Toc5873097)

[3.5 其他命令 14](#_Toc5873098)

[3.5.1 显示当前所在位置 14](#_Toc5873099)

[3.5.2 搜索命令 14](#_Toc5873100)

[3.5.3 管道命令 14](#_Toc5873101)

[3.5.4 查看进程 15](#_Toc5873102)

[3.5.5 杀死进程 15](#_Toc5873103)

[3.5.6 网络通信命令 15](#_Toc5873104)

[3.6 Linux的权限命令 17](#_Toc5873105)

[四、 远程连接工具的使用 18](#_Toc5873106)

[五、 Linux的软件安装 22](#_Toc5873107)

[5.1 JDK安装 22](#_Toc5873108)

[5.2 Mysql安装 22](#_Toc5873109)

[5.3 Tomcat安装 24](#_Toc5873110)

[六、 Vim命令合集 25](#_Toc5873111)

[6.1.1 vi/vim下退出及保存命令 25](#_Toc5873112)

[6.1.2 命令历史 25](#_Toc5873113)

[6.1.3 启动vim 25](#_Toc5873114)

[6.1.4 文件命令 26](#_Toc5873115)

[6.1.5 vim的模式 26](#_Toc5873116)

[6.1.6 导航命令 26](#_Toc5873117)

[6.1.7 插入命令 27](#_Toc5873118)

[6.1.8 查找命令 27](#_Toc5873119)

[6.1.9 替换命令 28](#_Toc5873120)

[6.1.10 移动命令 28](#_Toc5873121)

[6.1.11 撤销和重做 30](#_Toc5873122)

[6.1.12 删除命令 30](#_Toc5873123)

[6.1.13 拷贝和粘贴 31](#_Toc5873124)

[6.1.14 剪切命令 31](#_Toc5873125)

[6.1.15 退出命令 32](#_Toc5873126)

[6.1.16 窗口命令 32](#_Toc5873127)

[6.1.17 执行shell命令 33](#_Toc5873128)

[6.1.18 注释命令 33](#_Toc5873129)

[6.1.19 帮助命令 33](#_Toc5873130)

[6.1.20 其他非编辑命令 34](#_Toc5873131)

# Linux的简介

## Linux的概述

Linux是基于Unix的开源免费的操作系统，由于系统的稳定性和安全性几乎成为程序代码运行的最佳系统环境。Linux是由Linus Torvalds（林纳斯·托瓦兹）起初开发的，由于源代码的开放性，现在已经衍生出了千上百种不同的Linux系统。

Linux系统的应用非常广泛，不仅可以长时间的运行我们编写的程序代码，还可以安装在各种计算机硬件设备中，比如手机、平板电脑、路由器等。尤其在这里提及一下，我们熟知是Android程序最底层就是运行在linux系统上的。



图1 林纳斯·托瓦兹 图2 linux的logo

## Linux的分类

### Linux根据市场需求不同，基本分为两个方向：

图形化界面版：注重用户体验，类似window操作系统，但目前成熟度不够

服务器版：没有好看的界面，是以在控制台窗口中输入命令操作系统的，类似 于DOS，是我们假设服务器的最佳选择

### Linux根据原生程度，又分为两种：

内核版本：在Linus领导下的内核小组开发维护的系统内核的版本号

发行版本：一些组织或公司在内核版基础上进行二次开发而重新发行的版本

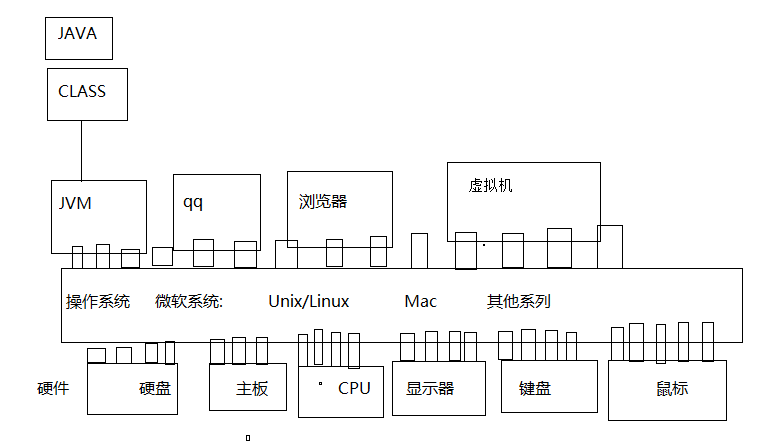
### Linux发行版本不同，又可以分为n多种：



# Linux

## 操作系统概念

操作系统作用:管理硬件,服务软件



硬件: 电路,写驱动程序,写应用程序

操作系统: 微软(windows) /Linux,Unix/苹果(mac)

Linux操作系统(世界巨头计算机,各个行业软件服务器系统)

系统开发人员,运维人员,程序开发人员

## 安装虚拟机

虚拟机(软件):通过调用系统接口模拟出一台机器

服务器(硬件/软件)

硬件:网吧一台公共机器,硬件服务器

软件服务器:WebServer(Tomcat) ,邮件,DNS服务器

参照笔记day21--01.VMware使用.pdf

## 安装Linux(CentOS版本)

参照笔记day21--02.Linux(CentOS)安装.pdf

## Linux历史介绍

了解Linux发展史

了解Linux应用/版本

## Linux目录结构介绍

Linux目录结构: 只有1个目录,根目录

usr:相当于program files

etc:存放系统配置文件

root:系统管理员默认目录

home:存放其他用户的目录

pwd: 打印当前目录

cd / : 切换到根目录

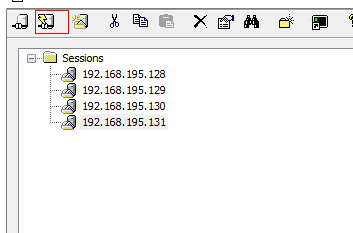
ll : 查看当前目录下的内容

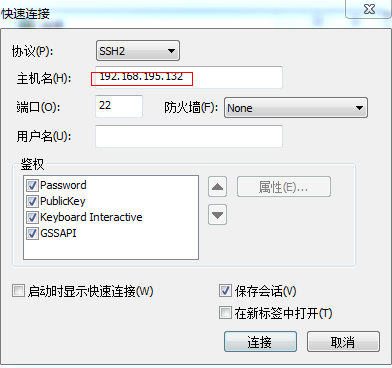
## 利用CRT连接linux

\*\_在linux获取ip: ifconfig

\*\_打开绿色版本的CRT: 点击SecureCRT.exe

\*\_点击文件--->连接--->快速连接



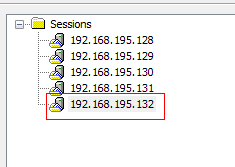


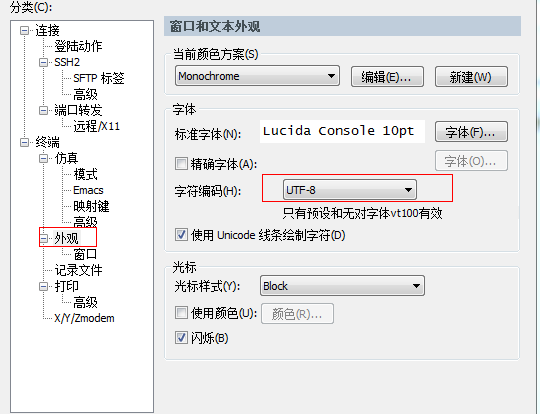
依次输入用户名,密码,连接到linux机器上.

解决CRT编码问题:

\*\_关闭CRT,重新打开

\*\_选中待连接的IP,右击--->属性





## 查找字符串

我们以samba的配置文件为例，搜索一个user的字符串。

vim /etc/samba/smb.conf

打开smb.conf

教你怎么在vi和vim上查找字符串

命令模式下，输入/user

/要搜索的字符串或者字符

按下回车之后，可以看到vim已经把光标移动到该字符处和高亮了匹配的字符串（vi没高亮，因为它没有颜色）

教你怎么在vi和vim上查找字符串

怎么查看下一个匹配呢？按下n(小写n)

教你怎么在vi和vim上查找字符串

我们一直按n到最后，红色的字提示BOTTOM（已经到底了，尽头了），说明匹配的字符串已经到此处是最后一个。再按n会回到TOP（第一个匹配成功的字符串）

教你怎么在vi和vim上查找字符串

那么如何跳转到上一个匹配呢？按下N（大写N）。

你可以按下Caps Lock键切换大小写，也可以在小写状态按下Shift + n

教你怎么在vi和vim上查找字符串

如果我们想从文件的结尾往开始处搜索呢？

使用?user

?要搜索的字符串或者字符

教你怎么在vi和vim上查找字符串

最后一个问题，搜索后，我们打开别的文件，发现也被高亮了，怎么关闭高亮？

教你怎么在vi和vim上查找字符串

命令模式下，输入:nohlsearch

PS：nohlsearch是（no highlight search缩写）

也可以:set nohlsearch

教你怎么在vi和vim上查找字符串

当然，可以简写。

noh或者set noh

## 复习

Linux:

1\_OS操作系统(operation system):

作用:控制硬件,服务软件 (3本书籍)

2\_虚拟机概念:(软件):虚拟出一台计算机环境

安装虚拟机(vmware)\_\_\_\_>配置 2个虚拟网卡

3\_在虚拟机上安装操作系统Linux(CentOS)

4\_Linux历史知识

应用:安装在各种服务器之上,用于嵌入式

版本: 内核版本,发行版本

5\_目录介绍 (1个目录)

root home usr etc

6\_核心命令

切换目录:cd cd / cd - cd ../ cd ~

查看目录下的文件/目录:list ll ls -al ls /root/

创建目录: mkdir ./b mkdir -pv /root/a/b/c/d

创建文件: touch /root/1.txt

复制目录: cp -r ./a /root/b

复制文件: cp ./install.log /root/

删除文件: rm /root/install.log rm -f /root/install.log

删除目录: rm -rf /root/a

剪切文件/目录: mv ./aaa/ /root/

重命名文件/目录: mv -f ./install.log ./test.log

查看文件: cat more less tail

搜索指定文件: find /root/ -name ‘test\*’

编辑文件: vim ./1.txt\_\_\_\_>IAO(iao)\_\_\_>录入内容\_\_\_>ESC\_\_\_>:wq

过滤: grep -i ‘sss’

管道: 不是具体的命令,而是命令组合使用的一种方式

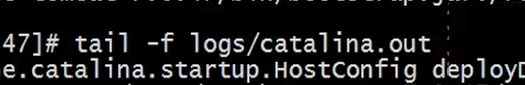
cat 1.txt | grep -i 'test'

ps -ef | grep -i 'crond'

系统命令: ps ifconfig ping kill

## 常用命令

Tail –f 循环查看



## 常见问题



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

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

```

yum update -y

```

## wget

```

yum -y install wget

```

---

## 查看状态

```

wget -qO- bench.sh | bash

```

---

@

## 文件、文件夹操作

### 删除

linux删除目录很简单，很多人还是习惯用rmdir，不过一旦目录非空

，就陷入深深的苦恼之中，现在使用rm -rf命令即可。

直接rm就可以了，不过要加两个参数-rf 即：rm -rf 目录名字

-r 就是向下递归，不管有多少级目录，一并删除

-f 就是直接强行删除，不作任何提示的意思

## 后台执行命令

```

nohup {{command options}}

```

nohup {{java -cp WEB-INF/lib/\*:WEB-INF/classes

org.b3log.solo.Starter}}

## 解压war包

```

jar -xvf project.war

```

screen -dmS java -cp WEB-INF/lib/\*:WEB-INF/classes

org.b3log.solo.Starter

### 区别



### 更新系统

yum **update** -y

**wget**

yum -y **install** wget

### 查看状态

wget -qO- bench.sh | bash

@

**文件、文件夹操作**

**删除**

linux删除目录很简单，很多人还是习惯用rmdir，不过一旦目录非空，就陷入深深的苦恼之中，现在使用rm -rf命令即可。 直接rm就可以了，不过要加两个参数-rf 即：rm -rf 目录名字 -r 就是向下递归，不管有多少级目录，一并删除 -f 就是直接强行删除，不作任何提示的意思

### 后台执行命令

**nohup** {{command options}}

nohup {{java -cp WEB-INF/lib/\*:WEB-INF/classes org.b3log.solo.Starter}}

**解压war包**

jar -xvf project.war

screen -dmS java -cp WEB-INF/lib/\*:WEB-INF/classes org.b3log.solo.Starter

screen -r

### SSR

wget -**N** --**no**-check-certificate https://raw.githubusercontent.**com**/FunctionClub/SSR-Bash-Python/master/install.**sh** && bash install.**sh**

1 <https://github.com/FunctionClub/SSR-Bash-Python/>

**-bash: wget: command not found 解决方法**

yum -y **install** wget

### Xshell实现windows上传文件到linux主机

<http://mofansheng.blog.51cto.com/8792265/1653048>

经常有这样的需求，我们在windows下载的软件包，如何上传到远程linux主机上？还有如何从linux主机下载软件包到windows下； 之前我的做法现在看来好笨好繁琐，不过也达到了目的，笨人有本方法嘛；

我是怎么操作的： 1、打开一台本地linux虚拟机，使用mount 挂载windows的共享文件夹到linux上，然后拷贝数据到linux虚拟机里面；（经常第一步都不顺利，无法挂载windows的文件夹） 2、在本地linux虚拟机使用rsync同步拷贝的数据到远程linux主机上，需要双方都要安装rsync包、openssh-clients包；遇到大一点的文件拷贝很费时间； 3、还有一种方法就是直接使用wget直接下载，提前是有下载的网址；大部分都是下载到windows本地然后上传到远程linux主机；

下面介绍一个简单的方法，方便上传windows的文件到linux上，也可以从linux下载到windows本地； 1、使用我们常用的Xshell登录工具，新建立一个远程会话，填写ip地址及用户名密码后，选择最下面的ZMODEM，填写下载的路径，加载的路径；2个路径可以一样也可以不一样； spacer.gifwKiom1Vb7JmDUla6AAJLbBDBWAg716.jpg

2、在linux主机上，安装上传下载工具包rz及sz 如果不知道你要安装包的具体名称，可以使用yum provides \*/name 进行查找系统自带软件包的信息；

[root@localhost src]# yum provides \*/rz lrzsz-0.12.20-27.1.el6.i686 : The lrz and lsz modem communications programs Repo : base Filename : /usr/bin/rz 一般会列出软件包的名称及版本，还有安装路径；查询到软件包名后，使用yum install -y 包名 进行安装。 lrzsz包安装完成后包括上传rz、下载sz命令；只需要安装这个包即可。 1 [root@localhost src]# yum install -y lrzsz

3、从windows上传文件，上传命令为rz；在linux命令行下输入rz，上传的文件在当前命令行的目录下； 1 [root@localhost src]# rz 输入rz命令后，会弹出对话框，选择你要上传的文件，选择打开就上传到linux主机。上传完可以使用ls 查看； wKiom1Vb7e7Qg7KBAAJSEEKdVSs344.jpg

4、从linux主机下载文件，下载命令为sz ，后面跟要下载的文件名；可以选择下载的保存文件夹； 1 [root@localhost src]# sz nginx-1.6.2.tar.gz wKioL1Vb75jgJKDPAAFdIWibogc477.jpg

### xshell下上传文件速度一直为0，可以尝试安装

yum -y install lrzsz

亲测，实用！

### Loaded plugins: fastestmirror, refresh-packagekit, security

[yaosiyuan@localhost ~]$ yum install glibc.i686

Loaded plugins: fastestmirror, refresh-packagekit, security

You need to be root to perform this command.

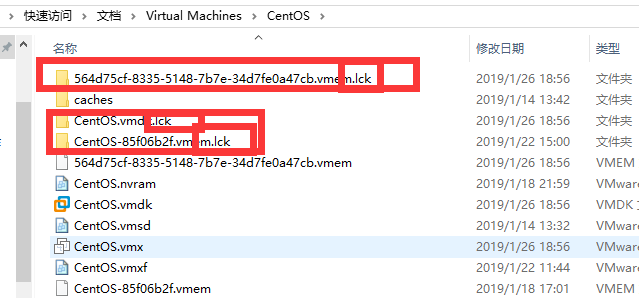
需要切换到root用户

su

然后输入密码 6 root用户时 #,普通用户时$

### VM

这主要是非正常关闭虚拟机造成的，未能锁定文件

虚拟机为了防止有多虚拟机共用一个虚拟磁盘（就是后 缀为.vmdk那个文件）造成数据的丢失和性能的削弱，每次启动虚拟机时会给每个虚拟磁盘加一个磁盘锁（也就是后缀为.lck的那个文件夹）对虚拟磁盘文件 进行锁定保护在关掉虚拟机时又会自动删除那个磁盘锁文件。当虚拟机非正常关闭，就会出现一个文件夹带有缀.lck，不会自动删除，才会引起这样报错。这样下次启动虚拟机时就会弹出上面的那个 错误。按照上面的解释，在虚拟机报的那个路径下删除了那个.lck文件夹全部删掉，再点启动该虚拟机即可。

## Xshell

### xshell5下 linux终端中常用的快捷键 CTRL + ALT + T: 打开终端 TAB: 自动补全命令或文件名 CTRL + SHIFT + V: 粘贴（Linux中不需要复制的动作，文本被选择就自动被复制） CTRL + SHIFT + T: 新建标签页 CTRL + D: 关闭标签页==logout CTRL + L: 清楚屏幕==clear命令 CTRL + R + 文本: 在输入历史中搜索 CTRL + A: 移动到行首 CTRL + E: 移动到行末 CTRL + C: 终止当前任务 CTRL + Z: 把当前任务放到后台运行（相当于运行命令时后面加&） ~: 表示用户目录路径

# 命令

服务器版的Linux安装完毕了，但没有图形化界面，我们只能通过控制台去操作系统，我们就要使用类似DOS命令的Linux命令去操作系统，那么下面我们就进行Linux的命令的学习。Linux命令成百上千，课程不能一一涉及，也没有必要一一涉及，根据我们程序员在日常工作中的需求，总结出如下几种供学习。

## 核心命令学习

linux有2种用户,一种管理员root,另外一种普通: zhangsan lisi

Root用户登录到linux默认所在的目录是 /root/

其他用户zhangsan登录到linux默认所在的目录是 /home/zhangsan/

### 查看目录结构

list :列出

ll :列出当前目录下的文件信息

ls -al :列出当前目录下的文件信息(包括隐藏文件,特殊目录)

ll /bin/ :列出根目录下bin目录下的文件信息

### 切换目录命令cd

touch 1.txt :在当前目录创建一个文件1.txt

clear :清除屏幕

cd ~

cd /

cd -

cd ..

cd

### 创建目录

pwd:打印工作目录

mkdir: 创建目录

mkdir a :当前目录下创建文件夹a

mkdir /root/b :在根目录下的root下创建目录b

mkdir -pv /root/c/e/d :在根目录下的root下创建目录结构c/e/d

### 文件浏览命令

cat命令: 如果文件中的内容很少,一页之内可以显示完毕

cat /root/文件名称

more命令: 如果文件中内容比较多,一页之内显示不完

more 文件名称

less 命令: 如果文件中内容比较多,一页之内显示不完

less -mN 文件名称

less比more多了搜索功能

tail 命令: 查看文件末尾内容

tail -10 /root/install.log 查看install.log文件中后10行内容

### 文件管理

cd /root/

**复制文件**

cp install.log install02.log

cp -b install.log install02.log :如果覆盖文件时,可以将源文件做一个备份

cp ./install.log ../ : 将当前目录下的install.log复制到父级目录

cp /root/install.log /bin/1.txt

:将/root下的install.log文件复制到根目录下的bin目录下的1.txt中

**复制目录**(无论是单层目录还是多层目录都可以复制):

cp -r ./a ./b

cp -r /root/a /root/z

**mv剪切文件**

mv /root/install.log /root/a/

**重命名文件**

mv /root/install.log /root/test.log

**删除文件**

rm -f 文件名称

rm -f /root/文件名称

**删除目录**

rm -rf 目录名称

rm -rf /root/目录名称

**在某个目录下搜索文件/目录**

find /root/ -name ‘in\*’ :

在根目录下的root目录下寻找名称中包含in而且是以in开头的文件/目录

### Vim编辑器

vim /root/1.txt :利用vim编辑器打开文件 ====> 录入 i (从一般模式转变为插入模式)

===>录入各种信息 ===>esc(从插入模式转换为一般模式)====>:wq(保存并退出)

### 系统命令

grep: 过滤任务 经常和其他命令联合使用

grep -i 名称

cat /root/install.log

cat /root/install.log | grep -i mysql :查看install.log文件中包含mysql内容信息

ps -ef :相当于任务管理器,对进程的一次查看

ps -ef | grep -i crond : 查看系统中crond的进程信息

管道命令: 命令的一种使用方式

命令1 | 命令2

命令1的输出是命令2的输入

ps aux|grep redis

cat /root/install.log | grep -i mysql

ps -ef | grep -i crond

**杀死进程 kill -9 进程id**

**查看IP: ifconfig**

**查看机器是否连接通畅: ping 对方机器ip**

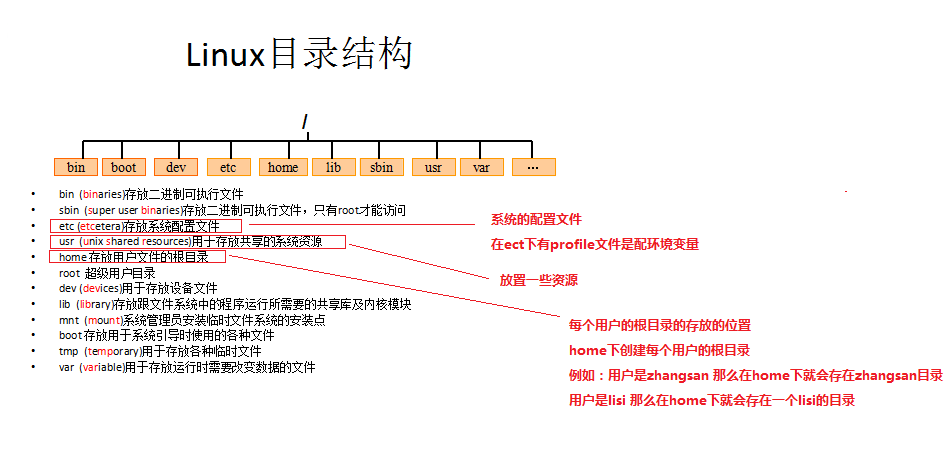
service network restart

重启网络服务。这样更改完虚拟机网络后不需要重启，一条命令就可以生效。

要求: 虚拟机\_\_\_>linux \_\_\_>镜像\_\_\_\_>crt\_\_>练习Linux命令

## 目录切换命令

window中通过图形化界面和鼠标可以任意切换需要进入的目录，但Linux必须通过cd命令切换目录。但在学习cd命令之前，我们有必要分清Linux的目录结构（[linux目录结构详细说明](资料/03-Linux各目录及每个目录的详细介绍.docx)）



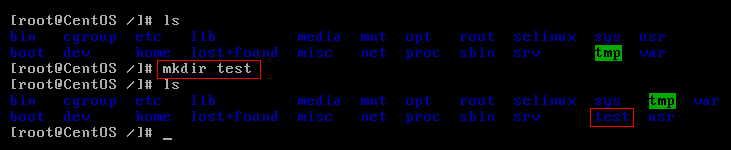
cd usr 切换到该目录下usr目录 cd ../ 切换到上一层目录 cd / 切换到系统根目录 cd ~ 切换到用户主目录 cd - 切换到上一个所在目录

## 目录的操作命令（增删改查）

### 增加目录操作（增）

命令：mkdir 目录名称

示例：在根目录 / 下 mkdir test，就会在根目录 / 下产生一个test问目录

mi

mkdir [选项] 目录

通过 mkdir 命令可以实现在指定位置创建以 DirName(指定的文件名)命名的文件夹或目录。

要创建文件夹或目录的用户必须对所创建的文件夹的父文件夹具有写权限。

并且，所创建的文件夹(目录)不能与其父目录(即父文件夹)中的文件名重名，

即同一个目录下不能有同名的(区分大小写)。

命令参数：

-m, --mode=模式，设定权限<模式> (类似 chmod)，而不是 rwxrwxrwx 减 umask

-p, --parents 可以是一个路径名称。

此时若路径中的某些目录尚不存在,加上此选项后,

系统将自动建立好那些尚不存在的目录,即一次可以建立多个目录;

-v, --verbose 每次创建新目录都显示信息

--help 显示此帮助信息并退出

--version 输出版本信息并退出

创建一个空目录

mkdir file1

递归创建多个目录

mkdir -p file1/file2

创建权限为777(rwxrwxrwx)的目录

mkdir -m 777 file3

创建新目录都显示信息

mkdir -v file4

一个命令创建项目的目录结构

mkdir -vp scf/{lib/,bin/,doc/{info,product},logs/{info,product},service/deploy/{info,product}}

mkdir: 已创建目录 “scf”

mkdir: 已创建目录 “scf/lib”

mkdir: 已创建目录 “scf/bin”

mkdir: 已创建目录 “scf/doc”

mkdir: 已创建目录 “scf/doc/info”

mkdir: 已创建目录 “scf/doc/product”

mkdir: 已创建目录 “scf/logs”

mkdir: 已创建目录 “scf/logs/info”

mkdir: 已创建目录 “scf/logs/product”

mkdir: 已创建目录 “scf/service”

mkdir: 已创建目录 “scf/service/deploy”

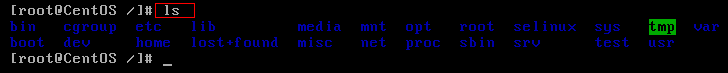
mkdir: 已创建目录 “scf/service/deploy/info”

mkdir: 已创建目录 “scf/service/deploy/product”

### 查看目录（查）

命令：ls [-al] 父目录

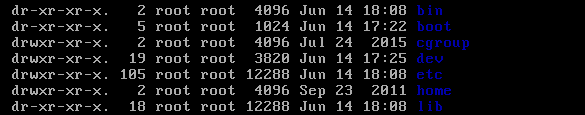
示例：在根目录 / 下使用ls，可以看到该目录下的所有的目录和文件



示例：在根目录 / 下使用ls -a，可以看到该目录下的所有文件和目录，包括隐藏的



示例：在根目录 / 下使用ls -l，可以看到该目录下的所有目录和文件的详细信息



**注意：ls -l 可以缩写成ll**

### 寻找目录（查）

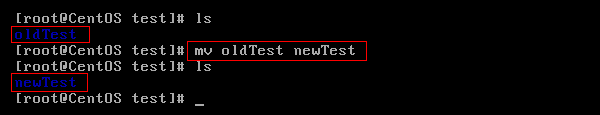
命令：find 目录 参数

示例：查找/root下的与test相关的目录(文件) find /root -name ‘test\*’

### 修改目录的名称（改）

命令：mv 目录名称 新目录名称

示例：test目录下有一个oldTest目录，使用mv oldTest newTest命令修改

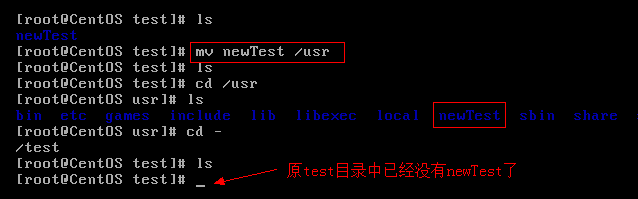


**注意：mv的语法不仅可以对目录进行重命名而且也可以对各种文件，压缩包等进行 重命名的操作**

### 移动目录的位置---剪切（改）

命令：mv 目录名称 目录的新位置

示例：在test下将newTest目录剪切到 /usr下面，使用mv newTest /usr

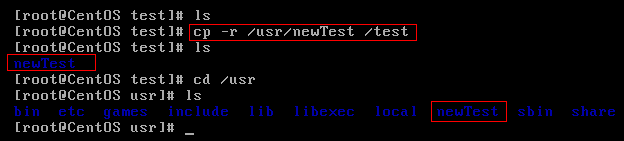


**注意：mv语法不仅可以对目录进行剪切操作，对文件和压缩包等都可执行剪切操作**

### 拷贝目录（改）

命令：cp -r 目录名称 目录拷贝的目标位置 -----r代表递归拷贝

示例：将/usr下的newTest拷贝到根目录下的test中，使用cp -r /usr/newTest /test

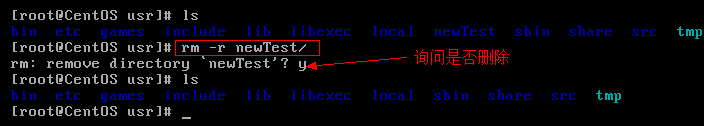


**注意：cp命令不仅可以拷贝目录还可以拷贝文件，压缩包等，拷贝文件和压缩包时不 用写-r递归**

### 删除目录（删）

命令：rm [-rf] 目录

示例：删除/usr下的newTest，进入/usr下使用rm -r newTest



示例：删除/test下的newTest而不需要询问强制删除，在/test下使用rm -rf newTest

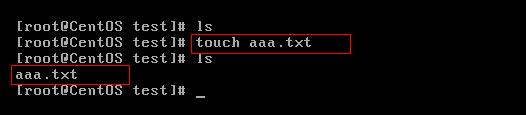


**注意：rm不仅可以删除目录，也可以删除其他文件或压缩包，为了增强大家的记忆， 无论删除任何目录或文件，都直接使用rm -rf 目录/文件/压缩包**

## 文件的操作命令（增删改查）

### 文件的创建（增）

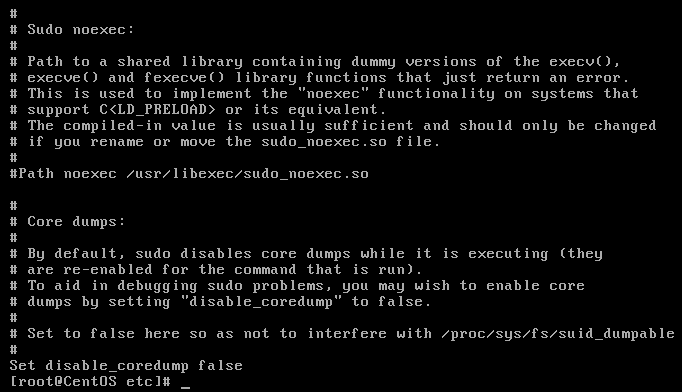
命令：touch 文件名称 ----- 空文件

示例：在test目录下创建一个空文件 touch aaa.txt

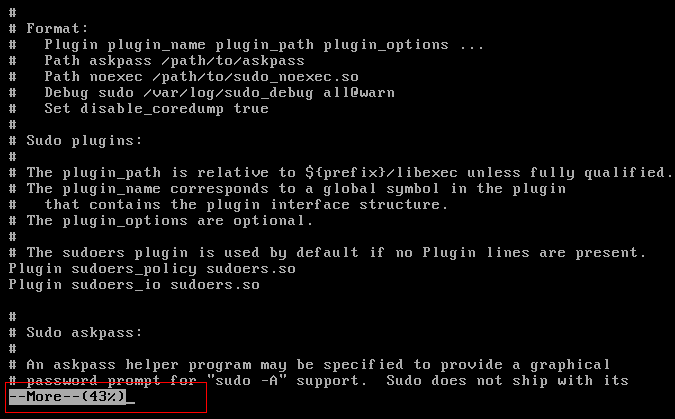
### 文件的查看（查）

命令：cat/more/less/tail 文件

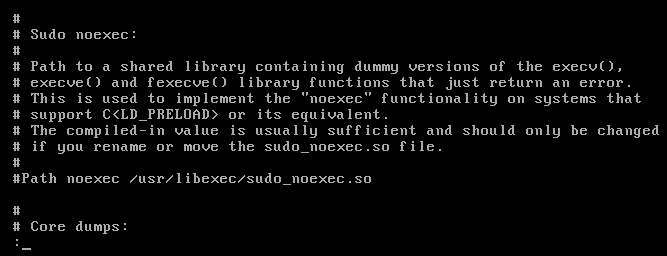
示例：使用cat查看/etc/sudo.conf文件，只能显示最后一屏内容



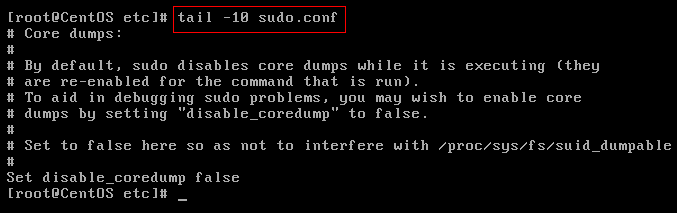
示例：使用more查看/etc/sudo.conf文件，可以显示百分比，回车可以向下一行， 空格可以向下一页，q可以退出查看



示例：使用less查看/etc/sudo.conf文件，可以使用键盘上的PgUp和PgDn向上 和向下翻页，q结束查看



示例：使用tail -10 查看/etc/sudo.conf文件的后10行，Ctrl+C结束

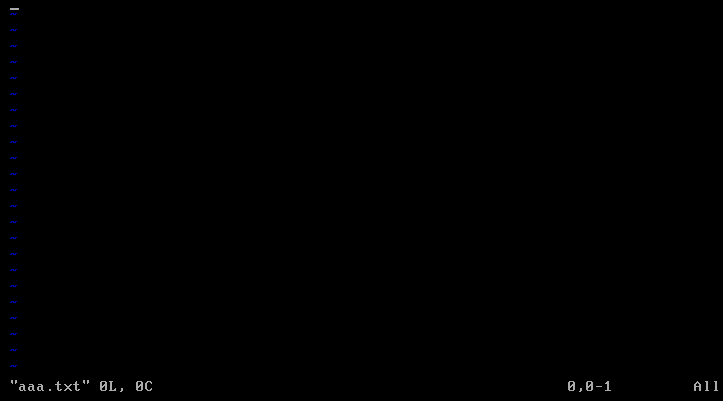


**注意：命令 tail -f 文件 可以对某个文件进行动态监控，例如tomcat的日志文件， 会随着程序的运行，日志会变化，可以使用tail -f catalina-2016-11-11.log 监控 文 件的变化**

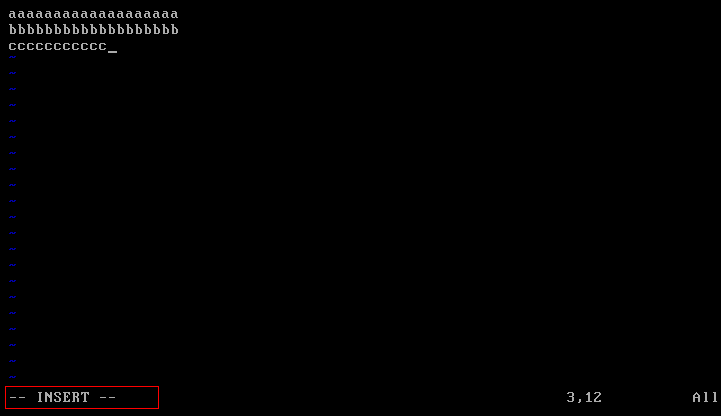
### 修改文件的内容（改）

命令：vim 文件

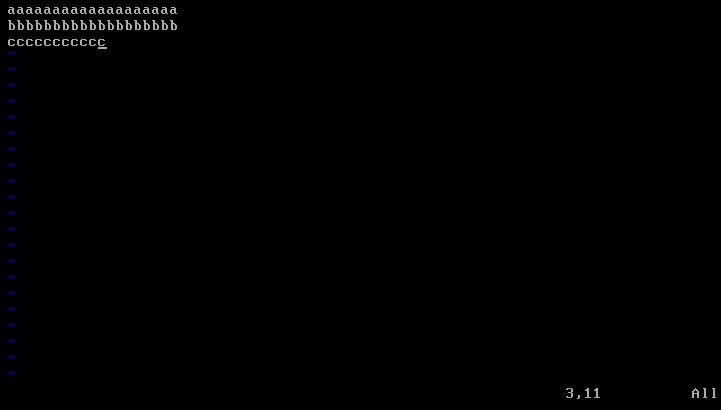
示例：编辑/test下的aaa.txt文件，使用vim aaa.txt



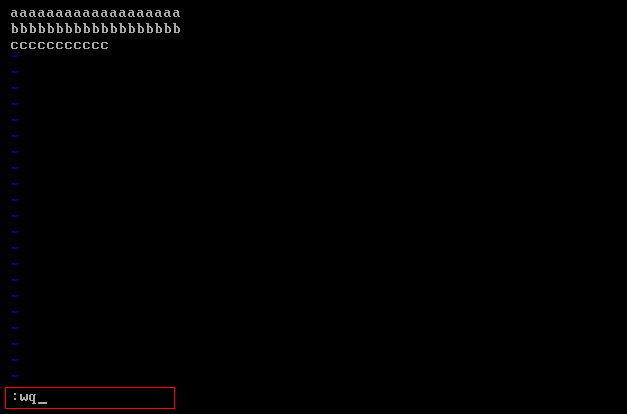
但此时并不能编辑，因为此时处于命令模式，点击键盘i/a/o进入编辑模式，可以 编辑文件



编辑完成后，按下Esc，退回命令模式



此时文件虽然已经编辑完成，但是没有保存，需输入冒号：进入底行模式，在底行模 式下输入wq代表写入内容并退出，即保存；输入q!代表强制退出不保存。



**总结：**

vim编辑器是Linux中的强大组件，是vi编辑器的加强版，vim编辑器的命令和快捷方式有很多，但此处不一一阐述，大家也无需研究的很透彻，使用vim编辑修改文件的方式基本会使用就可以了。附：[《vi使用方法介绍》](资料/04-vi使用方法详细介绍.docx)和[《vim命令合集》](资料/05-Vim命令合集.docx)

关于vim使用过程：

在实际开发中，使用vim编辑器主要作用就是修改配置文件

**vim 文件------>进入文件----->命令模式------>按i进入编辑模式----->编辑文件 ------->按Esc进入底行模式----->输入:wq/q!**

### 删除文件（删）

同目录删除：熟记 rm -rf 文件 即可

### 复制、移动文件

cp -r xxxapi ../appBackup/

mv xxxapi ../appBackup/

#将zookeeper-3.4.6.tar.gz移动到同级目录下的zk文件夹里

mv zookeeper-3.4.6.tar.gz ./zk

unzip project.war -d project

## 压缩文件的操作命令

### 打包并压缩文件

### 解压缩命令:

tar -zcvf /root/xxx.tar /root/a/

tar -zxvf ./xxx.tar

Linux中的打包文件一般是以.tar结尾的，压缩的命令一般是以.gz结尾的。

而一般情况下打包和压缩是一起进行的，打包并压缩后的文件的后缀名一般.tar.gz。

命令：tar -zcvf 打包压缩后的文件名 要打包压缩的文件

其中：z：调用gzip压缩命令进行压缩

c：打包文件

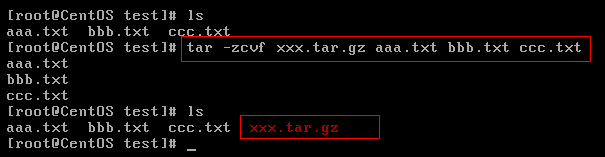
v：显示运行过程

f：指定文件名

示例：打包并压缩/test下的所有文件 压缩后的压缩包指定名称为xxx.tar.gz

tar -zcvf xxx.tar.gz aaa.txt bbb.txt ccc.txt

或：tar -zcvf xxx.tar.gz /test/\*

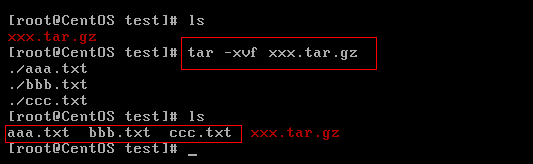


* + - 1. 解压压缩包**（重点）**

命令：tar [-xvf] 压缩文件

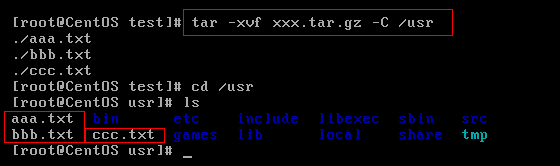
其中：x：代表解压

示例：将/test下的xxx.tar.gz解压到当前目录下

tar -xvf xxx.tar.gz

示例：将/test下的xxx.tar.gz解压到根目录/usr下

**tar -xvf xxx.tar.gz -C /usr------C代表指定解压的位置**



## 其他命令

### 显示当前所在位置

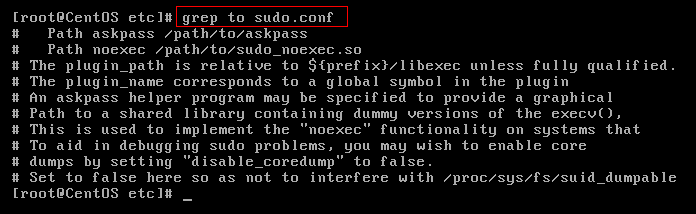
pwd



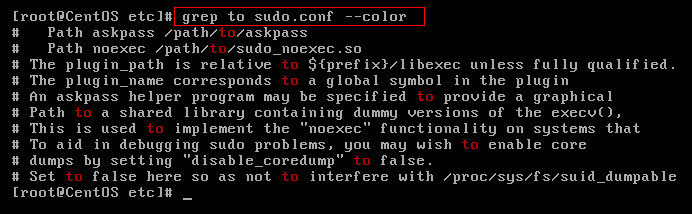
### 搜索命令

命令：grep 要搜索的字符串 要搜索的文件

示例：搜索/usr/sudu.conf文件中包含字符串to的行



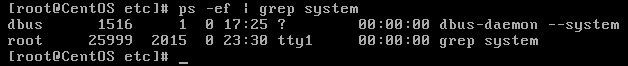
示例：搜索/usr/sudu.conf文件中包含字符串to的行 to要高亮显示



### 管道命令

命令：| 将前一个命令的输出作为本次目录的输入

示例：查看当前系统中所有的进程中包括system字符串的进程



### 查看进程

命令：ps -ef

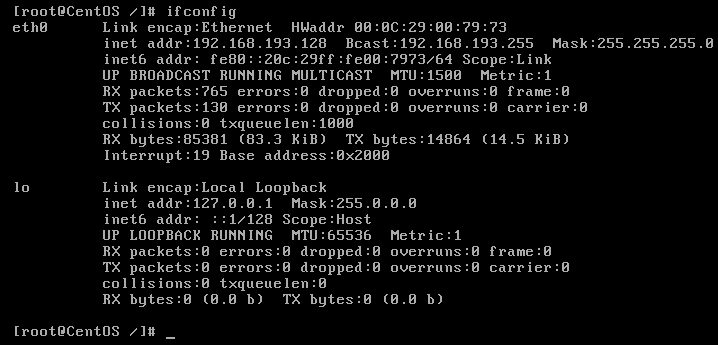
示例：查看当前系统中运行的进程

### 杀死进程

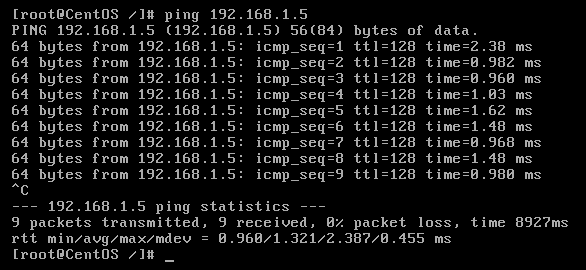
命令：kill -9 进程的pid

### 网络通信命令

查看当前系统的网卡信息：ifconfig

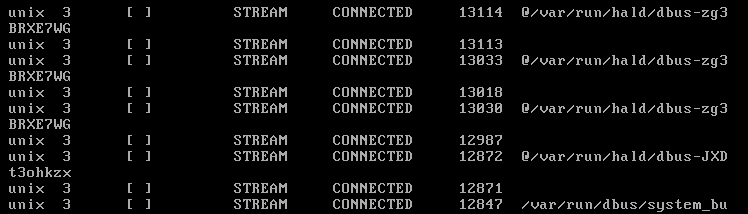


查看与某台机器的连接情况：ping



查看当前系统的端口使用：netstat -an

查看当前选中网卡



#### Centos 6关闭防火墙

service iptables stop #临时关闭防火墙

chkconfig iptables off #彻底关闭系统防火墙

service iptables status #查看防火墙的状态

#### Centos 7关闭防火墙

systemctl stop firewalld.service #停止firewall

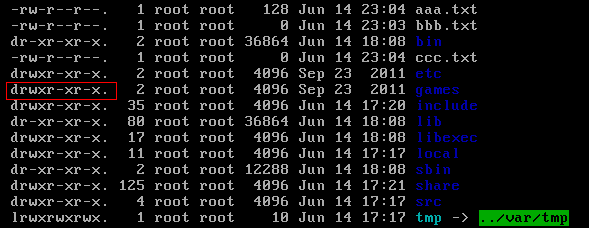
systemctl disable firewalld.service #禁止firewall开机启动

firewall-cmd --state #查看默认防火墙状态（关闭后显示notrunning，开启后显示running）

## Linux的权限命令

权限是Linux中的重要概念，每个文件/目录等都具有权限，通过ls -l命令我们可以 查看某个目录下的文件或目录的权限

示例：在随意某个目录下ls -l



第一列的内容的信息解释如下：



文件的类型：

d：代表目录

-：代表文件

l：代表链接（可以认为是window中的快捷方式）

后面的9位分为3组，每3位置一组，分别代表属主的权限，与当前用户同组的 用户的权限，其他用户的权限

r：代表权限是可读，r也可以用数字4表示

w：代表权限是可写，w也可以用数字2表示

x：代表权限是可执行，x也可以用数字1表示

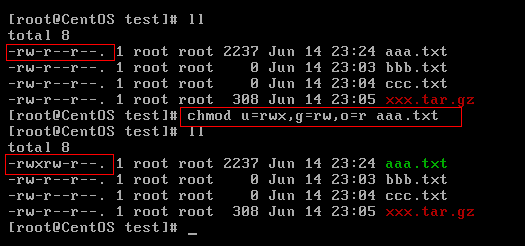
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **属主（user）** | | | **属组（group）** | | | **其他用户** | | |
| r | w | x | r | w | x | r | w | x |
| 4 | 2 | 1 | 4 | 2 | 1 | 4 | 2 | 1 |

修改文件/目录的权限的命令：chmod

示例：修改/test下的aaa.txt的权限为属主有全部权限，属主所在的组有读写权限，

其他用户只有读的权限

chmod u=rwx,g=rw,o=r aaa.txt



上述示例还可以使用数字表示：

chmod 764 aaa.txt

//给予超级权限

chomd u+x aaa.txt

## [Vim命令合集](http://www.cnblogs.com/softwaretesting/archive/2011/07/12/2104435.html)

第一种方式

:$ 跳转到最后一行

:1 跳转到第一行

第二种方式

shift+g 跳转到最后一行

gg 跳转到第一行

#### vi/vim下退出及保存命令

vi编辑器是Linux最基本的文本编辑器，工作在字符模式下，vim是vi的升级版。下面简单总结下vi/vim下退出及保存命令。

保存文件但不退出

:w 保存文件但不退出vi/vim编辑器:w file 将修改另外保存到file中，不退出vi/vim编辑器:w! 强制保存，不退出vi/vim编辑器

附：

:e! 放弃所有修改，从上次保存文件开始再编辑

保存文件并退出

:wq 保存文件，并退出vi/vim:wq! 强制保存文件，并退出vi/vim:wq! file 强制保存(覆盖)文件为file，并退出vi/vim:x 保存当前文件并退出vi/vim :x! 保存当前文件并退出vi/vimZZ 连续按两次ZZ，文件将保存并退出。

不保存文件并退出

:q 不保存文件，退出vi/vim:q! 不保存文件，强制退出vi/vim

#### 命令历史

以:和/开头的命令都有历史纪录，可以首先键入:或/然后按上下箭头来选择某个历史命令。

#### 启动vim

在命令行窗口中输入以下命令即可

vim 直接启动vim

vim filename 打开vim并创建名为filename的文件

#### 文件命令

打开单个文件

vim file

同时打开多个文件

vim file1 file2 file3 ...

在vim窗口中打开一个新文件

:open file

在新窗口中打开文件

:split file

切换到下一个文件

:bn

切换到上一个文件

:bp

查看当前打开的文件列表，当前正在编辑的文件会用[]括起来。

:args

打开远程文件，比如ftp或者share folder

:e ftp://192.168.10.76/abc.txt

:e \\qadrive\test\1.txt

#### vim的模式

正常模式（按Esc或Ctrl+[进入） 左下角显示文件名或为空

插入模式（按i键进入） 左下角显示--INSERT--

可视模式（不知道如何进入） 左下角显示--VISUAL--

#### 导航命令

% 括号匹配

#### 插入命令

i 在当前位置生前插入

I 在当前行首插入

a 在当前位置后插入

A 在当前行尾插入

o 在当前行之后插入一行

O 在当前行之前插入一行

#### 查找命令

/text 查找text，按n健查找下一个，按N健查找前一个。

?text 查找text，反向查找，按n健查找下一个，按N健查找前一个。

vim中有一些特殊字符在查找时需要转义 .\*[]^%/?~$

:set ignorecase 忽略大小写的查找

:set noignorecase 不忽略大小写的查找

查找很长的词，如果一个词很长，键入麻烦，可以将光标移动到该词上，按\*或#键即可以该单词进行搜索，相当于/搜索。而#命令相当于?搜索。

:set hlsearch 高亮搜索结果，所有结果都高亮显示，而不是只显示一个匹配。

:set nohlsearch 关闭高亮搜索显示

:nohlsearch 关闭当前的高亮显示，如果再次搜索或者按下n或N键，则会再次高亮。

:set incsearch 逐步搜索模式，对当前键入的字符进行搜索而不必等待键入完成。

:set wrapscan 重新搜索，在搜索到文件头或尾时，返回继续搜索，默认开启。

#### 替换命令

ra 将当前字符替换为a，当期字符即光标所在字符。

s/old/new/ 用old替换new，替换当前行的第一个匹配

s/old/new/g 用old替换new，替换当前行的所有匹配

%s/old/new/ 用old替换new，替换所有行的第一个匹配

%s/old/new/g 用old替换new，替换整个文件的所有匹配

:10,20 s/^/    /g 在第10行知第20行每行前面加四个空格，用于缩进。

ddp 交换光标所在行和其下紧邻的一行。

#### 移动命令

h 左移一个字符

l 右移一个字符，这个命令很少用，一般用w代替。

k 上移一个字符

j 下移一个字符

以上四个命令可以配合数字使用，比如20j就是向下移动20行，5h就是向左移动5个字符，在Vim中，很多命令都可以配合数字使用，比如删除10个字符10x，在当前位置后插入3个！，3a！<Esc>，这里的Esc是必须的，否则命令不生效。

w 向前移动一个单词（光标停在单词首部），如果已到行尾，则转至下一行行首。此命令快，可以代替l命令。

b 向后移动一个单词 2b 向后移动2个单词

e，同w，只不过是光标停在单词尾部

ge，同b，光标停在单词尾部。

^ 移动到本行第一个非空白字符上。

0（数字0）移动到本行第一个字符上，

<HOME> 移动到本行第一个字符。同0健。

$ 移动到行尾 3$ 移动到下面3行的行尾

gg 移动到文件头。 = [[

G（shift + g） 移动到文件尾。 = ]]

f（find）命令也可以用于移动，fx将找到光标后第一个为x的字符，3fd将找到第三个为d的字符。

F 同f，反向查找。

跳到指定行，冒号+行号，回车，比如跳到240行就是 :240回车。另一个方法是行号+G，比如230G跳到230行。

Ctrl + e 向下滚动一行

Ctrl + y 向上滚动一行

Ctrl + d 向下滚动半屏

Ctrl + u 向上滚动半屏

Ctrl + f 向下滚动一屏

Ctrl + b 向上滚动一屏

#### 撤销和重做

u 撤销（Undo）

U 撤销对整行的操作

Ctrl + r 重做（Redo），即撤销的撤销。

#### 删除命令

x 删除当前字符

3x 删除当前光标开始向后三个字符

X 删除当前字符的前一个字符。X=dh

dl 删除当前字符， dl=x

dh 删除前一个字符

dd 删除当前行

dj 删除上一行

dk 删除下一行

10d 删除当前行开始的10行。

D 删除当前字符至行尾。D=d$

d$ 删除当前字符之后的所有字符（本行）

kdgg 删除当前行之前所有行（不包括当前行）

jdG（jd shift + g）   删除当前行之后所有行（不包括当前行）

:1,10d 删除1-10行

:11,$d 删除11行及以后所有的行

:1,$d 删除所有行

J(shift + j) 删除两行之间的空行，实际上是合并两行。

#### 拷贝和粘贴

yy 拷贝当前行

nyy 拷贝当前后开始的n行，比如2yy拷贝当前行及其下一行。

p  在当前光标后粘贴,如果之前使用了yy命令来复制一行，那么就在当前行的下一行粘贴。

shift+p 在当前行前粘贴

:1,10 co 20 将1-10行插入到第20行之后。

:1,$ co $ 将整个文件复制一份并添加到文件尾部。

正常模式下按v（逐字）或V（逐行）进入可视模式，然后用jklh命令移动即可选择某些行或字符，再按y即可复制

ddp交换当前行和其下一行

xp交换当前字符和其后一个字符

#### 剪切命令

正常模式下按v（逐字）或V（逐行）进入可视模式，然后用jklh命令移动即可选择某些行或字符，再按d即可剪切

ndd 剪切当前行之后的n行。利用p命令可以对剪切的内容进行粘贴

:1,10d 将1-10行剪切。利用p命令可将剪切后的内容进行粘贴。

:1, 10 m 20 将第1-10行移动到第20行之后。

#### 退出命令

:wq 保存并退出

ZZ 保存并退出

:q! 强制退出并忽略所有更改

:e! 放弃所有修改，并打开原来文件。

#### 窗口命令

:split或new 打开一个新窗口，光标停在顶层的窗口上

:split file或:new file 用新窗口打开文件

split打开的窗口都是横向的，使用vsplit可以纵向打开窗口。

Ctrl+ww 移动到下一个窗口

Ctrl+wj 移动到下方的窗口

Ctrl+wk 移动到上方的窗口

关闭窗口

:close 最后一个窗口不能使用此命令，可以防止意外退出vim。

:q 如果是最后一个被关闭的窗口，那么将退出vim。

ZZ 保存并退出。

关闭所有窗口，只保留当前窗口

:only

录制宏

按q键加任意字母开始录制，再按q键结束录制（这意味着vim中的宏不可嵌套），使用的时候@加宏名，比如qa。。。q录制名为a的宏，@a使用这个宏。

#### 执行shell命令

:!command

:!ls 列出当前目录下文件

:!perl -c script.pl 检查perl脚本语法，可以不用退出vim，非常方便。

:!perl script.pl 执行perl脚本，可以不用退出vim，非常方便。

:suspend或Ctrl - Z 挂起vim，回到shell，按fg可以返回vim。

#### 注释命令

perl程序中#开始的行为注释，所以要注释某些行，只需在行首加入#

3,5 s/^/#/g 注释第3-5行

3,5 s/^#//g 解除3-5行的注释

1,$ s/^/#/g 注释整个文档。

:%s/^/#/g 注释整个文档，此法更快。

#### 帮助命令

:help or F1 显示整个帮助

:help xxx 显示xxx的帮助，比如 :help i, :help CTRL-[（即Ctrl+[的帮助）。

:help 'number' Vim选项的帮助用单引号括起

:help <Esc> 特殊键的帮助用<>扩起

:help -t Vim启动参数的帮助用-

：help i\_<Esc> 插入模式下Esc的帮助，某个模式下的帮助用模式\_主题的模式

帮助文件中位于||之间的内容是超链接，可以用Ctrl+]进入链接，Ctrl+o（Ctrl + t）返回

#### 其他非编辑命令

. 重复前一次命令

:set ruler? 查看是否设置了ruler，在.vimrc中，使用set命令设制的选项都可以通过这个命令查看

:scriptnames 查看vim脚本文件的位置，比如.vimrc文件，语法文件及plugin等。

:set list 显示非打印字符，如tab，空格，行尾等。如果tab无法显示，请确定用set lcs=tab:>-命令设置了.vimrc文件，并确保你的文件中的确有tab，如果开启了expendtab，那么tab将被扩展为空格。

Vim教程

在Unix系统上

$ vimtutor

在Windows系统上

:help tutor

:syntax 列出已经定义的语法项

:syntax clear 清除已定义的语法规则

:syntax case match 大小写敏感，int和Int将视为不同的语法元素

:syntax case ignore 大小写无关，int和Int将视为相同的语法元素，并使用同样的配色方案

### Vim命令

:

退出命令是，按ESC键 跳到命令模式，然后输入:q（不保存）或者:wq（保存） 退出。

更多退出命令：

: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命令历史

以:和/开头的命令都有历史纪录，可以首先键入:或/然后按上下箭头来选择某个历史命令。

启动vim

在命令行窗口中输入以下命令即可

vim 直接启动vim

vim filename 打开vim并创建名为filename的文件

文件命令

打开单个文件

vim file

同时打开多个文件

vim file1 file2 file3 …

在vim窗口中打开一个新文件

:open file

在新窗口中打开文件

:split file

切换到下一个文件

:bn

切换到上一个文件

:bp

查看当前打开的文件列表，当前正在编辑的文件会用[]括起来。

:args

打开远程文件，比如ftp或者share folder

:e ftp://192.168.10.76/abc.txt

:e \qadrive\test\1.txt

vim的模式

正常模式（按Esc或Ctrl+[进入） 左下角显示文件名或为空

插入模式（按i键进入） 左下角显示–INSERT–

可视模式（不知道如何进入） 左下角显示–VISUAL–

插入命令

i 在当前位置生前插入

I 在当前行首插入

a 在当前位置后插入

A 在当前行尾插入

o 在当前行之后插入一行

O 在当前行之前插入一行

查找命令

/text　　查找text，按n健查找下一个，按N健查找前一个。

?text　　查找text，反向查找，按n健查找下一个，按N健查找前一个。

vim中有一些特殊字符在查找时需要转义　　.\*[]^%/?~$

:set ignorecase　　忽略大小写的查找

:set noignorecase　　不忽略大小写的查找

查找很长的词，如果一个词很长，键入麻烦，可以将光标移动到该词上，按\*或#键即可以该单词进行搜索，相当于/搜索。而#命令相当于?搜索。

:set hlsearch　　高亮搜索结果，所有结果都高亮显示，而不是只显示一个匹配。

:set nohlsearch　　关闭高亮搜索显示

:nohlsearch　　关闭当前的高亮显示，如果再次搜索或者按下n或N键，则会再次高亮。

:set incsearch　　逐步搜索模式，对当前键入的字符进行搜索而不必等待键入完成。

:set wrapscan　　重新搜索，在搜索到文件头或尾时，返回继续搜索，默认开启。

替换命令

ra 将当前字符替换为a，当期字符即光标所在字符。

s/old/new/ 用old替换new，替换当前行的第一个匹配

s/old/new/g 用old替换new，替换当前行的所有匹配

%s/old/new/ 用old替换new，替换所有行的第一个匹配

%s/old/new/g 用old替换new，替换整个文件的所有匹配

:10,20 s/^/ /g 在第10行知第20行每行前面加四个空格，用于缩进。

ddp 交换光标所在行和其下紧邻的一行。

移动命令

h 左移一个字符

l 右移一个字符，这个命令很少用，一般用w代替。

k 上移一个字符

j 下移一个字符

以上四个命令可以配合数字使用，比如20j就是向下移动20行，5h就是向左移动5个字符，在Vim中，很多命令都可以配合数字使用，比如删除10个字符10x，在当前位置后插入3个！，3a！，这里的Esc是必须的，否则命令不生效。

w 向前移动一个单词（光标停在单词首部），如果已到行尾，则转至下一行行首。此命令快，可以代替l命令。

b 向后移动一个单词 2b 向后移动2个单词

e，同w，只不过是光标停在单词尾部

ge，同b，光标停在单词尾部。

^ 移动到本行第一个非空白字符上。

0（数字0）移动到本行第一个字符上，

移动到本行第一个字符。同0健。

移动到行尾3移动到行尾3 移动到下面3行的行尾

gg 移动到文件头。 = [[

G（shift + g） 移动到文件尾。 = ]]

f（find）命令也可以用于移动，fx将找到光标后第一个为x的字符，3fd将找到第三个为d的字符。

F 同f，反向查找。

跳到指定行，冒号+行号，回车，比如跳到240行就是 :240回车。另一个方法是行号+G，比如230G跳到230行。

Ctrl + e 向下滚动一行

Ctrl + y 向上滚动一行

Ctrl + d 向下滚动半屏

Ctrl + u 向上滚动半屏

Ctrl + f 向下滚动一屏

Ctrl + b 向上滚动一屏

撤销和重做

u 撤销（Undo）

U 撤销对整行的操作

Ctrl + r 重做（Redo），即撤销的撤销。

删除命令

x 删除当前字符

3x 删除当前光标开始向后三个字符

X 删除当前字符的前一个字符。X=dh

dl 删除当前字符， dl=x

dh 删除前一个字符

dd 删除当前行

dj 删除上一行

dk 删除下一行

10d 删除当前行开始的10行。

D 删除当前字符至行尾。D=ddd 删除当前字符之后的所有字符（本行）

kdgg 删除当前行之前所有行（不包括当前行）

jdG（jd shift + g） 删除当前行之后所有行（不包括当前行）

:1,10d 删除1-10行

:11,d删除11行及以后所有的行:1,d删除11行及以后所有的行:1,d 删除所有行

J(shift + j)　　删除两行之间的空行，实际上是合并两行。

拷贝和粘贴

yy 拷贝当前行

nyy 拷贝当前后开始的n行，比如2yy拷贝当前行及其下一行。

p 在当前光标后粘贴,如果之前使用了yy命令来复制一行，那么就在当前行的下一行粘贴。

shift+p 在当前行前粘贴

:1,10 co 20 将1-10行插入到第20行之后。

:1,coco 将整个文件复制一份并添加到文件尾部。

正常模式下按v（逐字）或V（逐行）进入可视模式，然后用jklh命令移动即可选择某些行或字符，再按y即可复制

ddp交换当前行和其下一行

xp交换当前字符和其后一个字符

剪切命令

正常模式下按v（逐字）或V（逐行）进入可视模式，然后用jklh命令移动即可选择某些行或字符，再按d即可剪切

ndd 剪切当前行之后的n行。利用p命令可以对剪切的内容进行粘贴

:1,10d 将1-10行剪切。利用p命令可将剪切后的内容进行粘贴。

:1, 10 m 20 将第1-10行移动到第20行之后。

#### 退出命令

:wq 保存并退出

ZZ 保存并退出

:q! 强制退出并忽略所有更改

:e! 放弃所有修改，并打开原来文件。

#### 窗口命令

:split或new 打开一个新窗口，光标停在顶层的窗口上

:split file或:new file 用新窗口打开文件

split打开的窗口都是横向的，使用vsplit可以纵向打开窗口。

Ctrl+ww 移动到下一个窗口

Ctrl+wj 移动到下方的窗口

Ctrl+wk 移动到上方的窗口

关闭窗口

:close 最后一个窗口不能使用此命令，可以防止意外退出vim。

:q 如果是最后一个被关闭的窗口，那么将退出vim。

ZZ 保存并退出。

关闭所有窗口，只保留当前窗口

:only

录制宏

按q键加任意字母开始录制，再按q键结束录制（这意味着vim中的宏不可嵌套），使用的时候@加宏名，比如qa。。。q录制名为a的宏，@a使用这个宏。

#### 执行shell命令

:!command

:!ls 列出当前目录下文件

:!perl -c script.pl 检查perl脚本语法，可以不用退出vim，非常方便。

:!perl script.pl 执行perl脚本，可以不用退出vim，非常方便。

:suspend或Ctrl - Z 挂起vim，回到shell，按fg可以返回vim。

注释命令

perl程序中#开始的行为注释，所以要注释某些行，只需在行首加入#

3,5 s/^/#/g 注释第3-5行

3,5 s/^#//g 解除3-5行的注释

1,$ s/^/#/g 注释整个文档。

:%s/^/#/g 注释整个文档，此法更快。

#### 帮助命令

:help or F1 显示整个帮助

:help xxx 显示xxx的帮助，比如 :help i, :help CTRL-[（即Ctrl+[的帮助）。

:help ‘number’ Vim选项的帮助用单引号括起

:help 特殊键的帮助用<>扩起

:help -t Vim启动参数的帮助用-

：help i\_ 插入模式下Esc的帮助，某个模式下的帮助用模式\_主题的模式

帮助文件中位于||之间的内容是超链接，可以用Ctrl+]进入链接，Ctrl+o（Ctrl + t）返回

#### 其他非编辑命令

. 重复前一次命令

:set ruler?　　查看是否设置了ruler，在.vimrc中，使用set命令设制的选项都可以通过这个命令查看

:scriptnames　　查看vim脚本文件的位置，比如.vimrc文件，语法文件及plugin等。

:set list 显示非打印字符，如tab，空格，行尾等。如果tab无法显示，请确定用set lcs=tab:>-命令设置了.vimrc文件，并确保你的文件中的确有tab，如果开启了expendtab，那么tab将被扩展为空格。

Vim教程

在Unix系统上

$ vimtutor

在Windows系统上

:help tutor

syntax

:syntax 列出已经定义的语法项

:syntax clear 清除已定义的语法规则

:syntax case match 大小写敏感，int和Int将视为不同的语法元素

:syntax case ignore 大小写无关，int和Int将视为相同的语法元素，并使用同样的配色方案

底线命令模式

:0或:1跳到文件第一行

:$跳到文件最后一行

命令模式

gg跳到第一行

shift+g跳到文件最后一行

## Linux剩余核心命令

关机/重启: reboot halt、

kill 5317 正常 关闭

kill/ 5317

### 更改目录/文件权限

权限:

用户/组/权限:

root管理员用户:

安装好Linux操作系统,同时在系统上安装QQ,火狐浏览器,快播,QQ游戏,eclipse

zhangsan:经常使用QQ,火狐浏览器,快播

root管理员在linux系统上为zhangsan创建一个用户名和密码 zhangsan<===>zhangsan

为张三分配QQ,火狐浏览器,快播三个软件的使用权限

zhangsan:经常使用QQ,火狐浏览器,快播

root管理员在linux系统上为lisi创建一个用户名和密码 lisi<===>lisi

为李四分配QQ,火狐浏览器,快播三个软件的使用权限

wangwu:经常使用QQ游戏,eclipse,快播

root管理员在linux系统上为wangwu创建一个用户名和密码 wangwu<===>wangwu

为王五分配QQ,火狐浏览器,快播三个软件的使用权限

root用户建立组: A组 ,为A组分配了QQ,火狐浏览器,快播三个软件的使用权限

tom申请使用linux系统,经常使用QQ,火狐浏览器,快播三个软件的使用权限

root为tom创建用户名和密码,创建用户的同时说明tom是属于A组

更改目录/文件用户权限

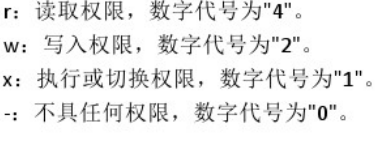
chmod u-rwx ./1.txt 取消当前用户对1.txt的”读写执行”权限

chmod g-rwx ./1.txt 取消当前组用户对1.txt的”读写执行”权限

chmod o-rwx ./1.txt 取消其他用户对1.txt的”读写执行”权限

chmod 777 ./1.txt 设置当前用户,当前组用户以及其他所有用户对文件1.txt的权限

以数字形式更改目录/文件权限的规则如下:

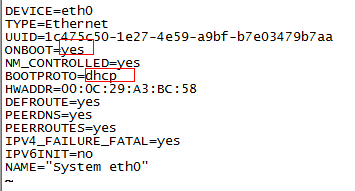


0:不具有任何权限 1: 可以执行 2:可以写　　　３(1+2):可以执行,可以写

4:可以读 5:(1+4):可以执行,可以读 6:(2+4):可以写可以读 7:(1+2+4):可以读,写,执行

Linux下网络配置

如果虚拟机安装完毕之后,看不到对应的2个虚拟网卡:通过vim修改网卡的相关配置项



如何查看各个命令的参数

Linux命令

内部命令:Linux系统核心命令 , 等价于windows<===>dir

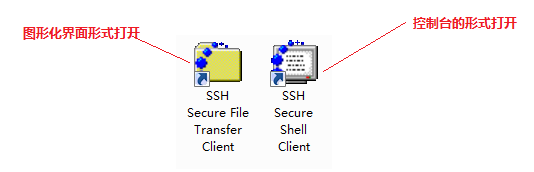
外部命令:后来加入的命令 , 等价于windows<===>java

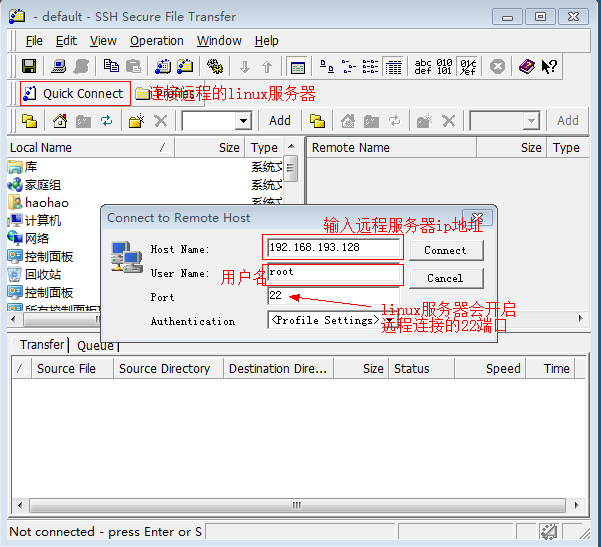
如果查看内部命令帮助文档方式: help + 命令

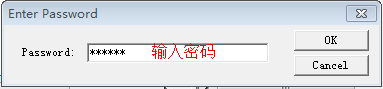
如果查看外部命令帮助文档方式: man + 命令

# 远程连接工具的使用

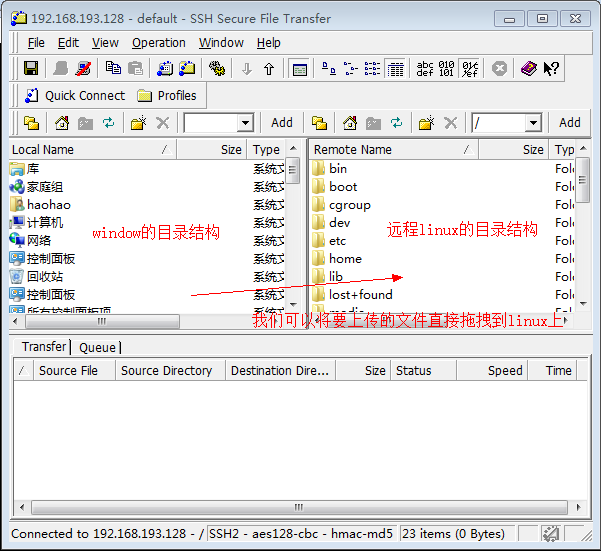
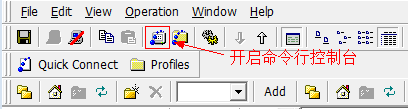
实际开发中，Linux服务器都在其他的地方，我们要通过远程的方式去连接Linux并操作它，Linux远程的操作工具有很多，企业中常用的有Puttty、secureCRT、SSH Secure等。课程中我们使用SSH Secure工具进行远程连接，该工具是免费的图形化界面及命令行窗口集一身的远程工具。安装包如下：

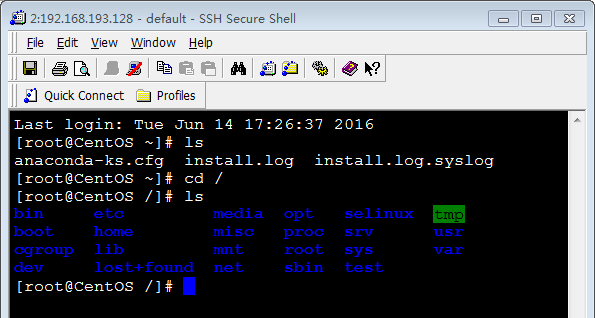
[](资料/SSHSecureShellClient-3.2.9.exe)安装后，会出现两个图标：

点击图形化界面图标：



进入界面：

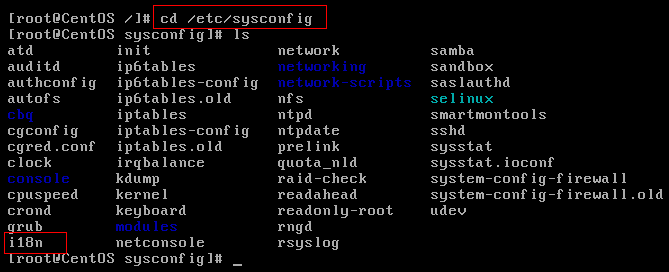




关于远程工具乱码的解决：

使用远程工具进行连接时，如果linux有中文文件或目录，显示时会出现乱码，原因是linux编码是UTF-8，而远程工具默认是当前系统本地编码即GBK。所以解决方案是统一两者编码就OK了，但是该SSH Secure工具不能设置编码，所以通过修改linux的系统编码的方式进行统一编码。

在linux的/etc/sysconfig目录下有一个i18n的文件代表linux的系统编码



将其从UTF-8修改成GBK重现连接linux即可：



# 服务器环境配置

## 虚拟机的安装

因为Linux也是一个系统，本质上跟我们电脑的Window没有区别，所以我们要学习Linux就首先将我们电脑的Window系统换成Linux系统，或者在我们电脑上安装双系统，听上去是不是很可怕。其实我们可以在我们电脑上安装一个软件，这个软甲可以模拟一台或多台虚拟的电脑机器，这就是虚拟机

虚拟器常用的有两种：

VMware（威睿）公司的虚拟机软件，功能强大，收费产品

 VitrualBox 虚拟机软件 Oracle公司的虚拟机软件，免费商品

由于市场上VMware使用率很高，所以我们安装VMware [《VMware安装手册》](资料/01-虚拟软件vmware安装.doc)

## CentOS的安装

CentOS是一个Linux的发行版本，是目前企业中用来做应用服务器系统的主要版本，CentOS的安装，其实是将该系统安装到VMware虚拟机软件中，让VMware虚拟机软件模拟出一台Linux系统的电脑。CentOS6.5安装请见[《CentOS安装手册》](资料/02-centOS6.7安装.doc)

## jdk

### JDK安

#### 上传jdk到linux

1. 安装jdk运行需要的插件yum install glibc.i686（选做）

解压jdk到/usr/local下 tar –xvf jdk-7u71-linux-i586.tar.gz –C /usr/local

配置jdk环境变量，打开/etc/profile配置文件，将下面配置拷贝进去

#set java environment

JAVA\_HOME=/usr/local/jdk1.7.0\_71

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

PATH=$JAVA\_HOME/bin:$PATH

export JAVA\_HOME CLASSPATH PATH

#### 重新加载/etc/profile配置文件 source /etc/profile

### Jdk的安装

查看linux上是否存在已经安装好的JDK javac

查看本机上所有已经安装成功的软件,只查看和java相关的

删除linux自带jdk

rpm -e --nodeps java-1.6.0-openjdk-1.6.0.35-1.13.7.1.el6\_6.i686

rpm -e --nodeps java-1.7.0-openjdk-1.7.0.79-2.5.5.4.el6.i686

#### 步骤一：查看现有安装的JDK版本

注意：rpm与软件相关命令 相当于window下的软件助手 管理软件

java –version

rpm -qa | grep java

rpm –qa | grep –i java



#### 步骤二：卸载已有的软件

先卸载open-jdk

rpm –e –nadeps javaxxx.xxx

输入rpm -e --nodeps javaxxx.xxx 要卸载的软件



#### 安装依赖包:

yum install glibc.i686

(需要联网)

#### 步骤三：解压jdk

开始安装：

mkdir /usr/local/src/java

mkdir /usr/local/src/mysql

mkdir /usr/local/src/tomcat

FileZilla\_3.7.3\_win32 上传jdk tar包

FileZilla\_3.7.3\_win32 上传mysql包

FileZilla\_3.7.3\_win32 上传tomcat包

cd /usr/local/src/java

将jdk压缩包进行解压

tar -zxvf jdk-7u71-linux-i586.tar.gz

z:制定压缩格式。

X:解压缩 c:创建

V:显示进度

F：制定文件

#### 步骤四：移动jdk到“/usr/local/src/java”

1\_利用Filezillar.exe在linux /usr/local/src/ 建立3个目录java,mysql,tomcat

2\_利用Filezillar.exe将windows上的3个软件jdk,mysql,tomcat上传到linux

#### 步骤五：配置环境变量

严格参照文档

① vi /etc/profile

② 在末尾行添加

③ 设置环境变量

#setjava environment

JAVA\_HOME=/www/server/jdk1.8.0\_60

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

PATH=$JAVA\_HOME/bin:$PATH

exportJAVA\_HOME CLASSPATH PATH

#setjava environment

JAVA\_HOME=/usr/java/jdk1.8.0\_60/

PATH=$JAVA\_HOME/bin:$PATH

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

export JAVA\_HOMEexport PATHexport CLASSPATH

保存退出

④ source /etc/profile 使更改的配置立即生效

⑤ java -version 查看JDK版本信息，如果显示出1.7.0证明成功

#### 1.8变1.7

不管在什么地方,什么时候,学习是快速提升自己的能力的一种体现!!!!!!!!!!!

关于JDK1.8 与之前的版本相比有哪些变化和新特性我也不在这详细的说明了,毕竟一度娘啥都有了,既然不多说那就直接开始吧!!

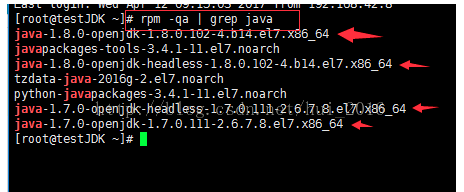
准备资料:

1、已经在VMware安装好了的虚拟机,本文使用的是CentOS7,我就以这个为例,如果又不知道怎么安装的朋友可以参考我第一篇文章,上面详细的介绍了如何安装CentOS7 下载安装centos7

2、jdk:jdk-8u11-linux-x64.tar.gz 官网下载jdk8

3、xshell 远程操作虚拟机工具 xshell的下载与安装请参考我上一篇文章(为什么要用到它来操作,因为我觉得它操作简单又是免费的.) 下载安装 xshell

OK，资料已准备好了，现在可以开始了



通过xshell工具成功连接安装好的虚拟机之后可通过 rpm -qa | grep java 或 rpm -qa | grep jdk 命令来查询出系统自带的jdk（带箭头的四个就是系统自带的）注：不带箭头的那三个不要删掉

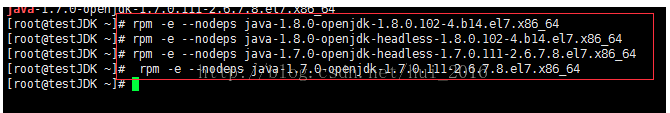
然后通过 rpm -e --nodeps 后面跟系统自带的jdk名 这个命令来删除系统自带的jdk，

例如：rpm -e --nodeps java-1.8.0-openjdk-1.8.0.102-4.b14.el7.x86\_64

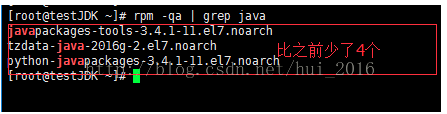
rpm -e --nodeps java-1.8.0-openjdk-headless-1.8.0.102-4.b14.el7.x86\_64

rpm -e --nodeps java-1.7.0-openjdk-headless-1.7.0.111-2.6.7.8.el7.x86\_64

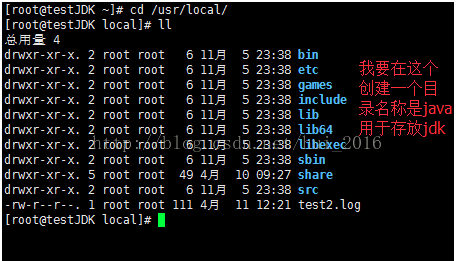
rpm -e --nodeps java-1.7.0-openjdk-1.7.0.111-2.6.7.8.el7.x86\_64



删完之后可以再通过 rpm -qa | grep java 或 rpm -qa | grep jdk 命令来查询出是否删除掉



接下来就是要安装自己的jdk了，通过命令 cd /usr/local/ 进入local目录，并通过 ll（两个小写的L）命令或者 ls 命令（ ll 本身不是命令，只是 ls -l 命令的一个别名，不过无所谓了，都是一样查看的）列出当前目录下得所有非隐含的文件，如果想要看到隐含（以 . 开头的，如 .test.txt）文件信息可通过 ll -a 来查看



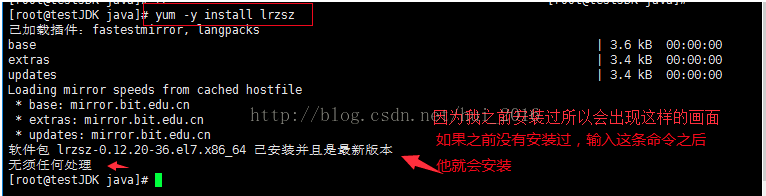
进入local目录之后 通过 mkdir java 命令来创建java目录存放自己的jdk

（扩展：如果你想一次性在同一级目录下创建多个平级的目录可以通过 mkdir brother1 brother2 (如要创建更多就在后面加上去就可以了，中间用空格隔开)这样的命令来创建，如果要一次创建父子目录（parent/child）可以通过

mkdir -p parent/child/grandson 来创建）

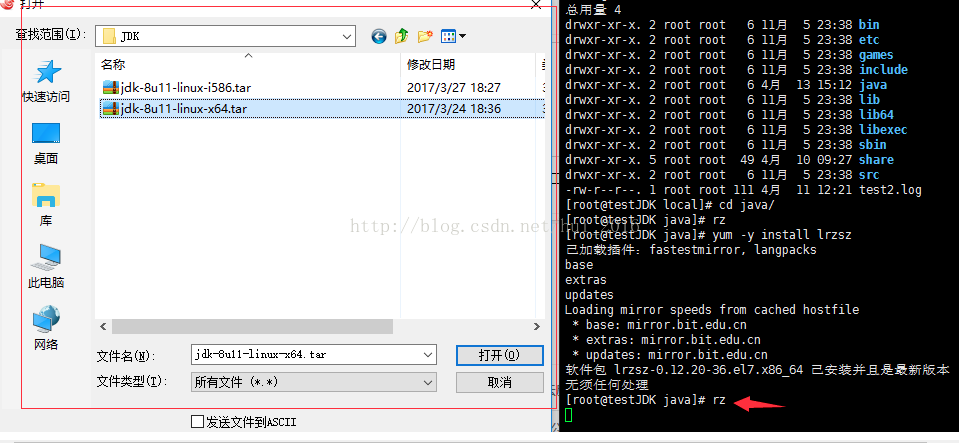
创建好之后，进入java目录 cd java

不知道大家有没有用过 在线导入安装包的插件，如果之前没有用过可以通过 yum -y install lrzsz 命令安装这个插件



安装插件完成之后输入 rz 命令然后按回车，就会弹出一个窗口，然后你就在这个窗口找到你下载好的jdk，

注：使用 rz 命令的好处就是你在哪里输入rz导入的安装包他就在哪里，不会跑到根目录下

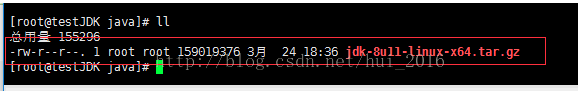


双击jdk或者按打开之后就会进入传输的界面

有时会出现下面这样的问题，如果出现请关闭该窗口重新输入 rz 命令按回车

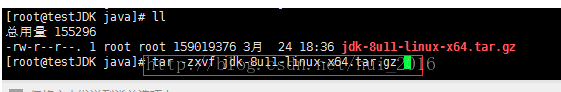
这个就是正常的了

传输完成之后 ll 命令 查看



通过 tar -zxvf {安装包名} 命令解压jdk 如：tar -zxvf jdk-8u11-linux-x64.tar.gz 我个人比较喜欢把 zxvf 转成中文念，叫

“至小威风” 这样我觉得方便记偷笑，不喜勿喷哦



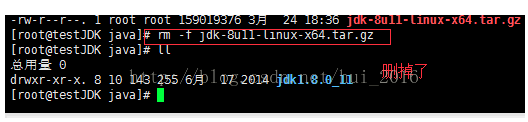
解压过后出现如下画面



这时安装包已经没用了，我一般都会删掉安装包 通过 rm -f jdk-8u11-linux-x64.tar.gz 删除安装包 -f 的意思就是 不询问删除 如果你不加 -f 在删除时它会询问你是否要删除该安装包，我是确定要删了就加上 -f 了。说到这里那我顺便说说

递归删除不询问 的命令吧，在什么情况下使用该命令呢，如果你要删除一个目录，而这个目录下还有目录或者有文件，比如在parent/child/grandson 这样的目录下你要删除 parent 下得所有目录和文件（包括parent）就可以用到

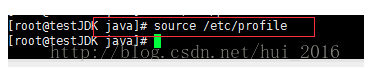
rm -rf parent 命令就可以删除掉了



删掉安装包之后就开始配置环境变量了 通过 vim /etc/profile 命令打开 profile 文件盘配置环境变量



打开之后按 i 进入 insert （插入）模式，在文件末尾添加上环境变量



export JAVA\_HOME=/usr/local/java/jdk1.8.0\_11

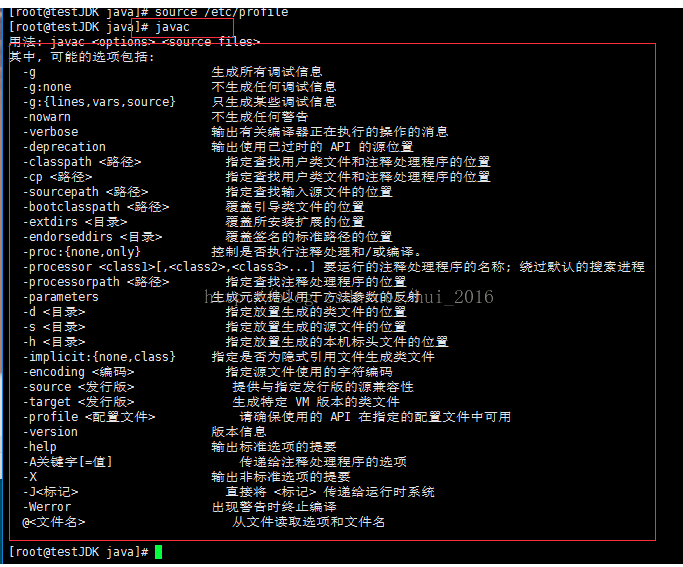
export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

export PATH=$PATH:$JAVA\_HOME/bin

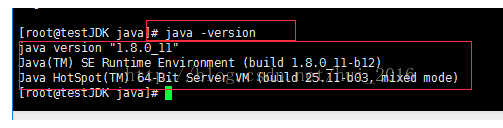
添加完之后保存并退出，保存并退出的命令有两种 第一种是： 按住shift 键然后连按两次z（这是我常用的，因为它方便快速），第二种是：wq 命令，有一种是不保存退出的命令：q ，注意：以上三种命令都是在非插入模式（插入模式下按键盘左上角的ESC键退出插入模式就是非插入模式了）下操作。

保存完之后输入：source /etc/profile 命令使刚才配置的环境变量生效

现在就可以测试jdk了 输入 javac 命令如果出现以下的文字就说明编译成功了（如果你之前安装centos7时使用的语言是英文，那出现的就是类似这样排版的英文）



接下来我们通过 java -version命令来查看你安装的jdk信息



## Mysql安装

步骤：

查看CentOS自带的mysql

输入 rpm -qa | grep mysql



将自带的mysql卸载



上传Mysql到linux

1. 安装mysql的依赖（选做）

yum -y install libaio.so.1 libgcc\_s.so.1 libstdc++.so.6

yum update libstdc++-4.4.7-4.el6.x86\_64

解压Mysql到/usr/local/下的mysql目录(mysql目录需要手动创建)内

cd /usr/local

mkdir mysql

tar -xvf MySQL-5.6.22-1.el6.i686.rpm-bundle.tar -C /usr/local/mysql

在/usr/local/mysql下安装mysql

安装服务器端：rpm -ivh MySQL-server-5.6.22-1.el6.i686.rpm

安装客户端：rpm -ivh MySQL-client-5.6.22-1.el6.i686.rpm

启动mysql

service mysql start

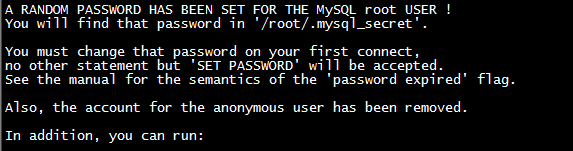
将mysql加到系统服务中并设置开机启动

加入到系统服务：chkconfig --add mysql

自动启动：chkconfig mysql on

登录mysql

mysql安装好后会生成一个临时随机密码，存储位置在/root/.mysql\_secret



msyql –u root -p

修改mysql的密码

set password = password('root');

开启mysql的远程登录

默认情况下mysql为安全起见，不支持远程登录mysql，所以需要设置开启 远程登录mysql的权限

登录mysql后输入如下命令：

grant all privileges on \*.\* to 'root' @'%' identified by 'root';

flush privileges;

开放Linux的对外访问的端口3306

/sbin/iptables -I INPUT -p tcp --dport 3306 -j ACCEPT

/etc/rc.d/init.d/iptables save ---将修改永久保存到防火墙中



## tomcat

### Tomcat安装

步骤：

上传Tomcat到linux上

解压Tomcat到/usr/local下

开放Linux的对外访问的端口8080

/sbin/iptables -I INPUT -p tcp --dport 8080 -j ACCEPT

/etc/rc.d/init.d/iptables save

启动关闭Tomcat

进入tomcat的bin下启动：./startup.sh

进入tomcat的bin下关闭：./shutdown.sh

### Tomcat

安装Tomcat

tomcat只要解压就可以使用。

创建web目录  
mkdir /usr/local/src/tomcat

上传apache-tomcat-7.0.57.tar.gz

解压：tar -zxvf apache-tomcat-7.0.57.tar.gz

重命名：mv apache-tomcat-7.0.57 tomcat

启动tomcat：  
cd itcast-usermanage/bin/  
./startup.sh 或者 sh startup.sh

查看日志：  
tail -f ../logs/catalina.out

图片.png

1. 4

找到“显示文档内容”模块，选中

图片.png

一、linux上操作tomacat相关命令

1.重启tomcat

./startup.sh

2.关闭tomacat

./shutdown.sh

3.解压文件

unzip project.war -d project

4、删除文件夹以及文件夹中的所有文件命令：

rm -rf 目录名字

其中：

-r：向下递归删除

-f：直接强行删除，且没有任何提示

5、删除文件命令

rm -f 文件名

6.查看文件

tail -f catalina.out

二、重启tomacat

1.首先，进入Tomcat下的bin目录

cd /usr/local/tomcat/bin

2.使用Tomcat关闭命令

./shutdown.sh

3.查看Tomcat是否以关闭

ps -ef|grep java

如果显示以下相似信息，说明Tomcat还没有关闭

root 7010 1 0 Apr19 ?

00:30:13 /usr/local/java/bin/java -Djava.util.logging.config.file=/usr/local/tomcat/conf/logging.properties

-Djava.awt.headless=true -Dfile.encoding=UTF-8 -server -Xms1024m

-Xmx1024m -XX:NewSize=256m

-XX:MaxNewSize=256m -XX:PermSize=256m -XX:MaxPermSize=256m -XX:+DisableExplicitGC

-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager

-Djava.endorsed.dirs=/usr/local/tomcat/endorsed

-classpath /usr/local/tomcat/bin/bootstrap.jar

-Dcatalina.base=/usr/local/tomcat -Dcatalina.home=/usr/local/tomcat

-Djava.io.tmpdir=/usr/local/tomcat/temp org.apache.catalina.startup.Bootstrap start

5\*如果你想直接干掉Tomcat，你可以使用kill命令，直接杀死Tomcat进程

kill -9 7010

6.然后继续查看Tomcat是否关闭

ps -ef|grep java

7.如果出现以下信息，则表示Tomcat已经关闭

root 7010 1 0 Apr19 ? 00:30:30 [java] <defunct>

8.最后，启动Tomcat

./startup.sh

三、删除文件命令

//删除对应的文件

rm -rf 文件全名

//删除一下的目录文件

rm -f 文件路径

四、复制、移动文件

cp -r xxxapi ../appBackup/

mv xxxapi ../appBackup/

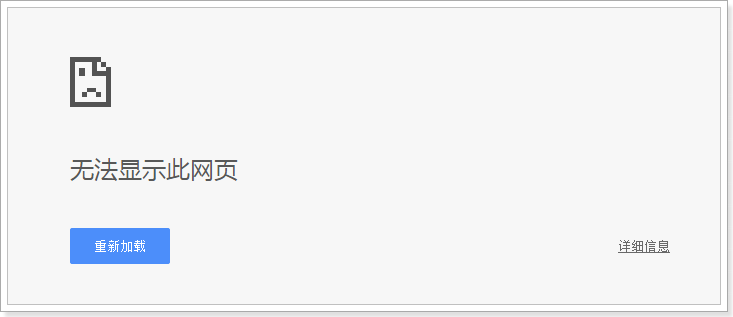
unzip project.war -d project

#### 1.指定运行jdk

1)setclasspath.sh和catalina.sh中写入:

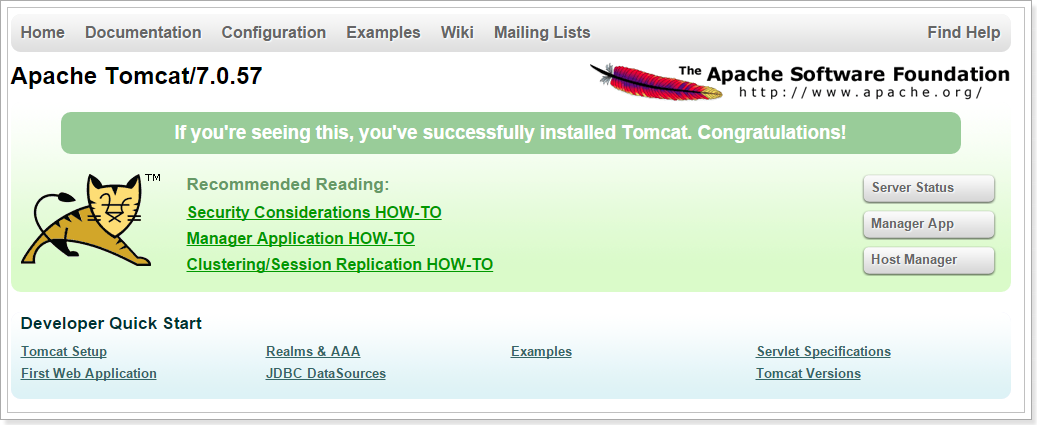
export JAVA\_HOME=/home/manager/java/jdk1.8.0\_121

export JRE\_HOME=/home/manager/java/jdk1.8.0\_121/jre/

查看效果 http://192.168.0.160:8080/  
发现无法访问：  


1. 防火墙打开 8080 端口  
   /sbin/iptables -I INPUT -p tcp --dport 8080 -j ACCEPT

/etc/rc.d/init.d/iptables save

1. 安装成功  
   

#### 账户

方法一：

修改端口号

在安装目录的conf文件夹下找到server.xml文件，找到此下代码

<Connector port="8080" protocol="HTTP/1.1"

connection Timeout="20000"

redirectPort="8443"/>

将port=8080改为任意值，例：port=8081.

（如果多处有<Connector port="8080 …………">,多处的8080值均要修改）。

方法二：

tomcat 7.0没有默认值，需要在conf的tomcat-users.xml中设置密码。

文件内容添加下方红色标注的内容，改后部分内容为（不是全部）：

<?xml version='1.0' encoding='utf-8'?>

<tomcat-users>

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<role rolename="manager-jmx"/>

<role rolename="manager-status"/>

<role rolename="admin-gui"/>

<role rolename="admin-script"/>

<user username="yaosiyuan" password="yaoSIyuan" roles="manager-gui, manager-script, manager-jmx, manager-status, admin-gui, admin-script"/>

</tomcat-users>

附言：

1、注意修改时不要在

<!--

<role rolename="tomcat"/>

<role rolename="role1"/>

<user username="tomcat" password="tomcat" roles="tomcat"/>

<user username="both" password="tomcat" roles="tomcat,role1"/>

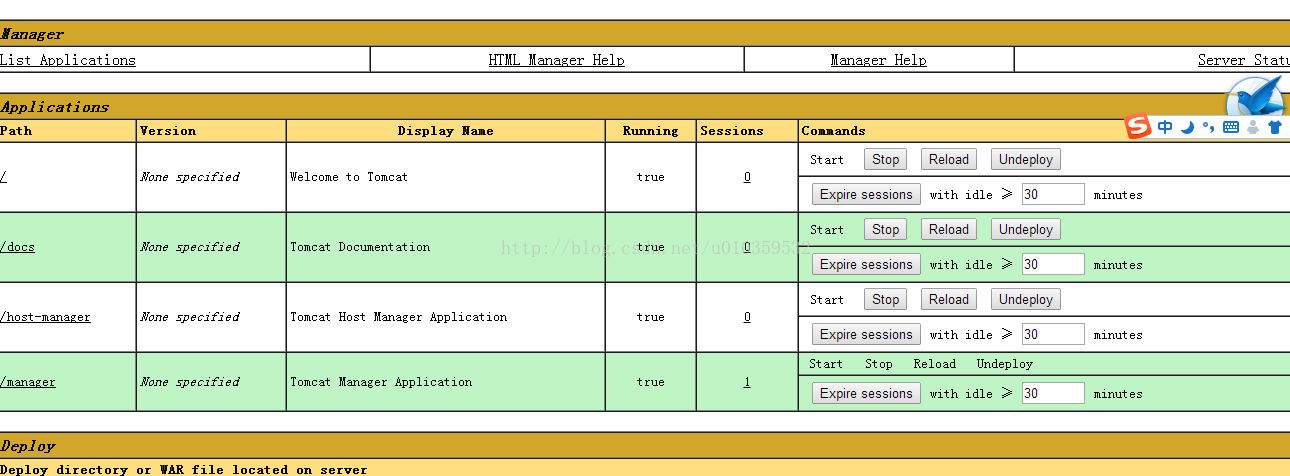
<user username="role1" password="tomcat" roles="role1"/>

-->

标注的内容里面修改，因为<!-- -->是注释，里面内容会被忽略掉。

2、tomcat 7.0需要将manager-gui, manager-script, manager-jmx, manager-status, admin-gui, admin-script 这些全写上。只写manager-gui还是会登陆不上的。

3、登陆上会出现如下图所示。



#### Tomcat下设置项目为默认项目

项目的实际使用中常常须要将当前项目设为tomcat的默认项目，而不是进入到tomcat的页面，有几种方法能够实现，注意另外一种、第三种情况须要先删除webapps下的ROOT文件夹，否则会失败。

一、将自己的项目改名为ROOT注意项目名称要大写

假设改完后没生效。建议清空tomcat中的work文件夹（缓存）

二、改动配置文件server.xml

添加<Context path="" docBase="Cluster" debug="0" reloadable="true"/>节点，当中docBase为webapps中的项目。注意，<Context>一定要加在<Host>节点中。样例例如以下：

<Host name="localhost" appBase="webapps"

unpackWARs="true" autoDeploy="true"

xmlValidation="false" xmlNamespaceAware="false">

<Context path="" docBase="navigation" debug="0" reloadable="true"/>

三、改动配置文件server.xml

添加<Context path="" docBase="E:\apps\Cluster" debug="0" reloadable="true"/>节点，当中docBase为须要设为默认项目的绝对路径。注意，<Context>一定要加在<Host>节点中。样例例如以下：

<Host name="localhost" appBase="webapps"

unpackWARs="true" autoDeploy="true"

xmlValidation="false" xmlNamespaceAware="false">

<Context path="" docBase="E:\apps\Cluster" debug="0" reloadable="true"/>

## 依赖包

1\_linux系统安装完毕之后做快照

2\_在有网络的环境下,可以安装一下依赖包

JDK依赖包:

yum install glibc.i686

MYSQL依赖包:

yum -y install libaio.so.1 libgcc\_s.so.1 libstdc++.so.6

yum update libstdc++-4.4.7-4.el6.x86\_64

yum -y install libncurses.so.5 libtinfo.so.5

redis依赖程序

yum install gcc-c++

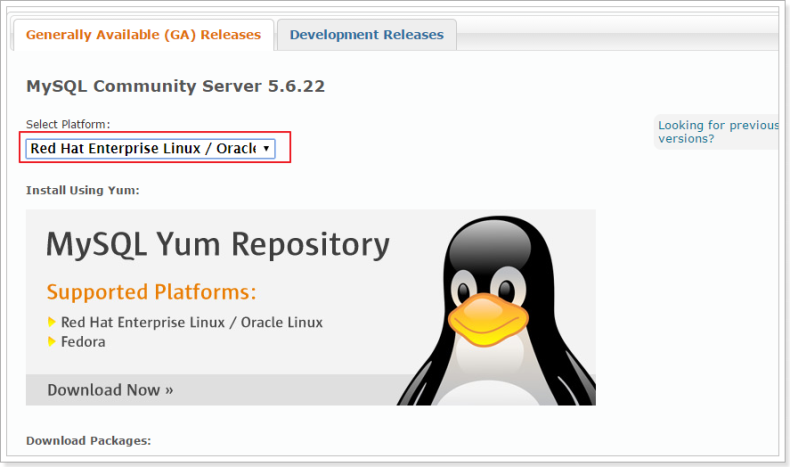
安装jdk,mysql,gcc相关的依赖包

在linux安装jdk

## Mysql的安装

### 下载

<http://dev.mysql.com/downloads/mysql/>





或者使用wget下载：

wget http://dev.mysql.com/get/Downloads/MySQL-5.6/MySQL-5.6.22-1.el6.i686.rpm-bundle.tar

### 安装

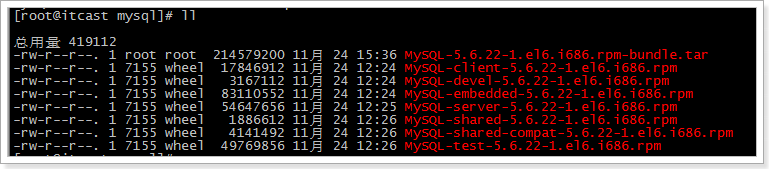
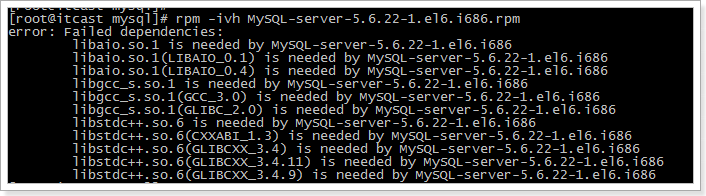
### 检测是否已经安装了mysql

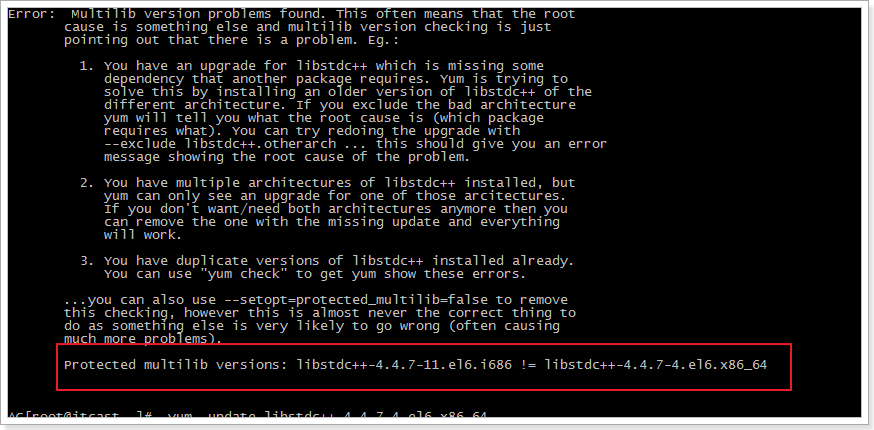
rpm -qa | grep mysql

如果已经安装了，将其卸载，如：

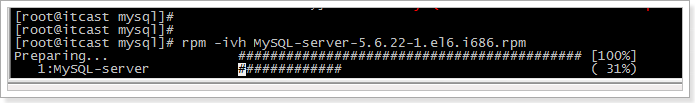
rpm -e --nodeps mysql-libs-5.1.73-5.el6\_6.i686

### 安装mysql

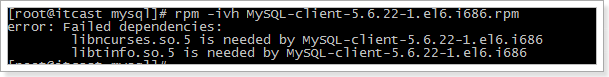
1. mkdir /usr/local/src/mysql
2. cd /usr/local/src/mysql
3. tar -xvf MySQL-5.6.22-1.el6.i686.rpm-bundle.tar  
   
4. 安装server  
   rpm -ivh MySQL-server-5.6.22-1.el6.i686.rpm  
   出错：  
     
   安装依赖：

yum -y install libaio.so.1 libgcc\_s.so.1 libstdc++.so.6   
  
   
  
需要升级libstdc++-4.4.7-4.el6.x86\_64  
yum update libstdc++-4.4.7-4.el6.x86\_64

1. 安装中：

rpm -ivh MySQL-server-5.6.22-1.el6.i686.rpm  
 

1. 安装client

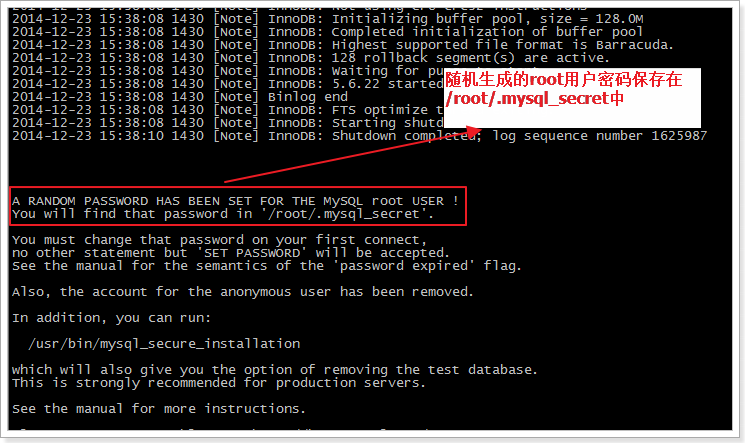
rpm -ivh MySQL-client-5.6.22-1.el6.i686.rpm   
   
安装依赖：

yum -y install libncurses.so.5 libtinfo.so.5

1. 查询mysq服务运行状态

service mysql status  
   
服务未启动1。

1. 启动mysql服务  
   service mysql start
2. 使用root账号登录mysql  
   提示：

  
在安装mysql server时有句提示：  
  
注意：这个密码是不安全的，所有需要修改初始密码。

1. 使用密码登录mysql账号：mysql -uroot -p
2. 修改root密码：SET PASSWORD = PASSWORD('123456');

### 系统启动时自动启动mysql服务

加入到系统服务：

chkconfig --add mysql

自动启动：

chkconfig mysql on

查询列表：

chkconfig

说明：都没关闭（off）时是没有自动启动。

### 开启远程访问

登录：

mysql -uroot –p123456

设置远程访问（使用root密码）：

grant:赋予

privileges:权限,规则

grant all privileges on \*.\* to 'root' @'%' identified by 'root';

flush privileges;

防火墙打开3306端口

/sbin/iptables -I INPUT -p tcp --dport 3306 -j ACCEPT

将打开的端口的访问规则保存在文件中

/etc/rc.d/init.d/iptables save

查看机器开放的端口号

/etc/init.d/iptables status

tar -xvf MySQL-5.6.22-1.el6.i686.rpm-bundle.tar^C

MySQL-client-5.6.22-1.el6.i686.rpm

MySQL-server-5.6.22-1.el6.i686.rpm

rpm -ivh MySQL-client-5.6.22-1.el6.i686.rpm

### 检查mysql服务器的编码格式

1）登录mysql客户端：mysql -uroot -p

2）查询编码格式：show variables like 'character%';

### 通过mysql命令查看mysql的安装路径

select @@basedir as basePath from dual

## CentOS安装Maven

现有的一个项目使用了Maven来管理，源代码放到了Subversion中。虽然Maven管理项目很方便，但是部署起来还是很麻烦的。先要在本地生成项目jar包，上传到服务器，然后再重启服务。如果在服务器上面安装Maven，便可以直接在服务器上面生成项目jar包，部署起来更加方便了。

Maven的下载地址是：http://maven.apache.org/download.cgi

安装Maven非常简单，只需要将下载的压缩文件解压就可以了。

cd /usr/local/src/

wget http://mirrors.hust.edu.cn/apache/maven/maven-3/3.1.1/binaries/apache-maven-3.1.1-bin.tar.gz

tar zxf apache-maven-3.1.1-bin.tar.gz

mv apache-maven-3.1.1 /usr/local/maven3

vi /etc/profile然后还需要 配置环境变量。

#在适当的位置添加

export M2\_HOME=/usr/local/maven3

export PATH=$PATH:$JAVA\_HOME/bin:$M2\_HOME/bin

保存退出后运行下面的命令使配置生效，或者重启服务器生效。

source /etc/profile

验证版本

mvn -v

如果安装正确就可以看到Maven的版本了。如果提示以下错误，请检查/etc/profile中PATH环境变量值中是否包含Maven可执行文件的路径，以及是否使该配置生效。

-bash: mvn: command not found

## zookeeper

### Zookeeper介绍

官方推荐使用zookeeper注册中心。

注册中心负责服务地址的注册与查找，相当于目录服务，服务提供者和消费者只在启动时与注册中心交互，注册中心不转发请求，压力较小。使用dubbo-2.3.3以上版本，建议使用zookeeper注册中心。

Zookeeper是Apacahe Hadoop的子项目，是一个树型的目录服务，支持变更推送，适合作为Dubbo服务的注册中心，工业强度较高，可用于生产环境，并推荐使用

Zookeeper：

1. 可以作为集群的管理工具使用。
2. 可以集中管理配置文件。

### 安装1

安装环境：

Linux：centos6.4

Jdk:1.7以上版本

Zookeeper是java开发的可以运行在windows、linux环境。需要先安装jdk。

安装步骤：

第一步：安装jdk

第二步：把zookeeper的压缩包上传到linux系统。

第三步：解压缩压缩包

tar -zxvf zookeeper-3.4.6.tar.gz

第四步：进入zookeeper-3.4.6目录，创建data文件夹。

第五步：把zoo\_sample.cfg改名为zoo.cfg

[root@localhost conf]# mv zoo\_sample.cfg zoo.cfg

第六步：修改data属性：dataDir=/root/zookeeper-3.4.6/data

第七步：启动zookeeper

[root@localhost bin]# ./zkServer.sh start

关闭：[root@localhost bin]# ./zkServer.sh stop

查看状态：[root@localhost bin]# ./zkServer.sh status

**注意：需要关闭防火墙。**

### 安装2

user/local/zk

上传.tar

tar –zxvf ..

cd conf

cp zoo\_sample.cfg zoo.cfg

vim zoo.cfg

dataDir=/www/server/zookeeper-3.4.6/data

#修改环境变量



vim /etc/profile

export ZK\_HOME=/www/server/zookeeper-3.4.6

export PATH=.:$ZK\_HOME/BIN:$PATH

:wq 保存

## 发布

# 常见问题

## maven

### 关于Could not transfer artifact (https://repo.maven.apache.org/maven2): Received fatal alert: protocol

Failed to read artifact descriptor for org.slf4j:slf4j-api:jar:1.7.2

org.eclipse.aether.resolution.ArtifactDescriptorException: Failed to read artifact descriptor for org.slf4j:slf4j-api:jar:1.7.2

at org.apache.maven.repository.internal.DefaultArtifactDescriptorReader.loadPom(DefaultArtifactDescriptorReader.java:302)

at org.apache.maven.repository.internal.DefaultArtifactDescriptorReader.readArtifactDescriptor(DefaultArtifactDescriptorReader.java:218)

at org.eclipse.aether.internal.impl.DefaultDependencyCollector.resolveCachedArtifactDescriptor(DefaultDependencyCollector.java:535)

at org.eclipse.aether.internal.impl.DefaultDependencyCollector.getArtifactDescriptorResult(DefaultDependencyCollector.java:519)

at org.eclipse.aether.internal.impl.DefaultDependencyCollector.processDependency(DefaultDependencyCollector.java:409)

at org.eclipse.aether.internal.impl.DefaultDependencyCollector.processDependency(DefaultDependencyCollector.java:363)

at org.eclipse.aether.internal.impl.DefaultDependencyCollector.process(DefaultDependencyCollector.java:351)

at org.eclipse.aether.internal.impl.DefaultDependencyCollector.doRecurse(DefaultDependencyCollector.java:504)

at org.eclipse.aether.internal.impl.DefaultDependencyCollector.processDependency(DefaultDependencyCollector.java:458)

at org.eclipse.aether.internal.impl.DefaultDependencyCollector.processDependency(DefaultDependencyCollector.java:363)

at org.eclipse.aether.internal.impl.DefaultDependencyCollector.process(DefaultDependencyCollector.java:351)

at org.eclipse.aether.internal.impl.DefaultDependencyCollector.doRecurse(DefaultDependencyCollector.java:504)

at org.eclipse.aether.internal.impl.DefaultDependencyCollector.processDependency(DefaultDependencyCollector.java:458)

at org.eclipse.aether.internal.impl.DefaultDependencyCollector.processDependency(DefaultDependencyCollector.java:363)

at org.eclipse.aether.internal.impl.DefaultDependencyCollector.process(DefaultDependencyCollector.java:351)

at org.eclipse.aether.internal.impl.DefaultDependencyCollector.collectDependencies(DefaultDependencyCollector.java:254)

at org.eclipse.aether.internal.impl.DefaultRepositorySystem.collectDependencies(DefaultRepositorySystem.java:316)

at org.apache.maven.project.DefaultProjectDependenciesResolver.resolve(DefaultProjectDependenciesResolver.java:172)

at org.apache.maven.project.DefaultProjectBuilder.resolveDependencies(DefaultProjectBuilder.java:215)

at org.apache.maven.project.DefaultProjectBuilder.build(DefaultProjectBuilder.java:188)

at org.apache.maven.project.DefaultProjectBuilder.build(DefaultProjectBuilder.java:119)

at org.eclipse.m2e.core.internal.embedder.MavenImpl.readMavenProject(MavenImpl.java:636)

at org.eclipse.m2e.core.internal.project.registry.DefaultMavenDependencyResolver.resolveProjectDependencies(DefaultMavenDependencyResolver.java:63)

at org.eclipse.m2e.core.internal.project.registry.ProjectRegistryManager.refreshPhase2(ProjectRegistryManager.java:529)

at org.eclipse.m2e.core.internal.project.registry.ProjectRegistryManager$3.call(ProjectRegistryManager.java:491)

at org.eclipse.m2e.core.internal.project.registry.ProjectRegistryManager$3.call(ProjectRegistryManager.java:1)

at org.eclipse.m2e.core.internal.embedder.MavenExecutionContext.executeBare(MavenExecutionContext.java:176)

at org.eclipse.m2e.core.internal.embedder.MavenExecutionContext.execute(MavenExecutionContext.java:151)

at org.eclipse.m2e.core.internal.project.registry.ProjectRegistryManager.refresh(ProjectRegistryManager.java:495)

at org.eclipse.m2e.core.internal.project.registry.ProjectRegistryManager.refresh(ProjectRegistryManager.java:350)

at org.eclipse.m2e.core.internal.project.registry.ProjectRegistryManager.refresh(ProjectRegistryManager.java:297)

at org.eclipse.m2e.core.internal.project.ProjectConfigurationManager.updateProjectConfiguration0(ProjectConfigurationManager.java:398)

at org.eclipse.m2e.core.internal.project.ProjectConfigurationManager$2.call(ProjectConfigurationManager.java:345)

at org.eclipse.m2e.core.internal.project.ProjectConfigurationManager$2.call(ProjectConfigurationManager.java:1)

at org.eclipse.m2e.core.internal.embedder.MavenExecutionContext.executeBare(MavenExecutionContext.java:176)

at org.eclipse.m2e.core.internal.embedder.MavenExecutionContext.execute(MavenExecutionContext.java:151)

at org.eclipse.m2e.core.internal.embedder.MavenExecutionContext.execute(MavenExecutionContext.java:99)

at org.eclipse.m2e.core.internal.embedder.MavenImpl.execute(MavenImpl.java:1351)

at org.eclipse.m2e.core.internal.project.ProjectConfigurationManager.updateProjectConfiguration(ProjectConfigurationManager.java:342)

at org.eclipse.m2e.core.ui.internal.UpdateMavenProjectJob.runInWorkspace(UpdateMavenProjectJob.java:77)

at org.eclipse.core.internal.resources.InternalWorkspaceJob.run(InternalWorkspaceJob.java:39)

at org.eclipse.core.internal.jobs.Worker.run(Worker.java:55)

Caused by: org.eclipse.aether.resolution.ArtifactResolutionException: Could not transfer artifact org.slf4j:slf4j-api:pom:1.7.2 from/to central (https://repo.maven.apache.org/maven2): Received fatal alert: protocol\_version

at org.eclipse.aether.internal.impl.DefaultArtifactResolver.resolve(DefaultArtifactResolver.java:444)

at org.eclipse.aether.internal.impl.DefaultArtifactResolver.resolveArtifacts(DefaultArtifactResolver.java:246)

at org.eclipse.aether.internal.impl.DefaultArtifactResolver.resolveArtifact(DefaultArtifactResolver.java:223)

at org.apache.maven.repository.internal.DefaultArtifactDescriptorReader.loadPom(DefaultArtifactDescriptorReader.java:287)

... 41 more

Caused by: org.eclipse.aether.transfer.ArtifactTransferException: Could not transfer artifact org.slf4j:slf4j-api:pom:1.7.2 from/to central (https://repo.maven.apache.org/maven2): Received fatal alert: protocol\_version

at io.takari.aether.connector.AetherRepositoryConnector$2.wrap(AetherRepositoryConnector.java:895)

at io.takari.aether.connector.AetherRepositoryConnector$2.wrap(AetherRepositoryConnector.java:1)

at io.takari.aether.connector.AetherRepositoryConnector$GetTask.flush(AetherRepositoryConnector.java:673)

at io.takari.aether.connector.AetherRepositoryConnector.get(AetherRepositoryConnector.java:310)

at org.eclipse.aether.internal.impl.DefaultArtifactResolver.performDownloads(DefaultArtifactResolver.java:520)

at org.eclipse.aether.internal.impl.DefaultArtifactResolver.resolve(DefaultArtifactResolver.java:421)

... 44 more

Caused by: javax.net.ssl.SSLException: Received fatal alert: protocol\_version

at sun.security.ssl.Alerts.getSSLException(Unknown Source)

at sun.security.ssl.Alerts.getSSLException(Unknown Source)

at sun.security.ssl.SSLSocketImpl.recvAlert(Unknown Source)

at sun.security.ssl.SSLSocketImpl.readRecord(Unknown Source)

at sun.security.ssl.SSLSocketImpl.performInitialHandshake(Unknown Source)

at sun.security.ssl.SSLSocketImpl.startHandshake(Unknown Source)

at sun.security.ssl.SSLSocketImpl.startHandshake(Unknown Source)

at com.squareup.okhttp.Connection.upgradeToTls(Connection.java:242)

at com.squareup.okhttp.Connection.connect(Connection.java:159)

at com.squareup.okhttp.Connection.connectAndSetOwner(Connection.java:175)

at com.squareup.okhttp.OkHttpClient$1.connectAndSetOwner(OkHttpClient.java:120)

at com.squareup.okhttp.internal.http.HttpEngine.nextConnection(HttpEngine.java:330)

at com.squareup.okhttp.internal.http.HttpEngine.connect(HttpEngine.java:319)

at com.squareup.okhttp.internal.http.HttpEngine.sendRequest(HttpEngine.java:241)

at com.squareup.okhttp.Call.getResponse(Call.java:271)

at com.squareup.okhttp.Call$ApplicationInterceptorChain.proceed(Call.java:228)

at com.squareup.okhttp.Call.getResponseWithInterceptorChain(Call.java:199)

at com.squareup.okhttp.Call.execute(Call.java:79)

at io.takari.aether.okhttp.OkHttpAetherClient.execute(OkHttpAetherClient.java:154)

at io.takari.aether.okhttp.OkHttpAetherClient.get(OkHttpAetherClient.java:100)

at io.takari.aether.connector.AetherRepositoryConnector$GetTask.resumableGet(AetherRepositoryConnector.java:600)

at io.takari.aether.connector.AetherRepositoryConnector$GetTask.run(AetherRepositoryConnector.java:453)

at io.takari.aether.connector.AetherRepositoryConnector.get(AetherRepositoryConnector.java:304)

... 46 more

#### 解决

最近突然发现，maven的中央仓库下载pom文件失败，报错是Could not transfer artifact (https://repo.maven.apache.org/maven2): Received fatal alert: protocol\_version -> [Help 1]，浏览器是能够访问的。

其实就是中央仓库必须要TLS1.2版本才能访问，貌似是今年六月份刚改的，所以有两种解决方法

第一种方法：将jdk版本升级到jdk1.8，因为在1.8及以上版本，才会默认使用TLS1.2

第二种方法(没有测试过):

maven的setting.xml设置为阿里云私服

<mirrors>

<!-- mirror

| Specifies a repository mirror site to use instead of a given repository. The repository that

| this mirror serves has an ID that matches the mirrorOf element of this mirror. IDs are used

| for inheritance and direct lookup purposes, and must be unique across the set of mirrors.

|

<mirror>

<id>mirrorId</id>

<mirrorOf>repositoryId</mirrorOf>

<name>Human Readable Name for this Mirror.</name>

<url>http://my.repository.com/repo/path</url>

</mirror>

-->

<mirror>

<id>nexus-aliyun</id>

<mirrorOf>\*</mirrorOf>

<name>Nexus aliyun</name>

<url>http://maven.aliyun.com/nexus/content/groups/public</url>

</mirror>

</mirrors>