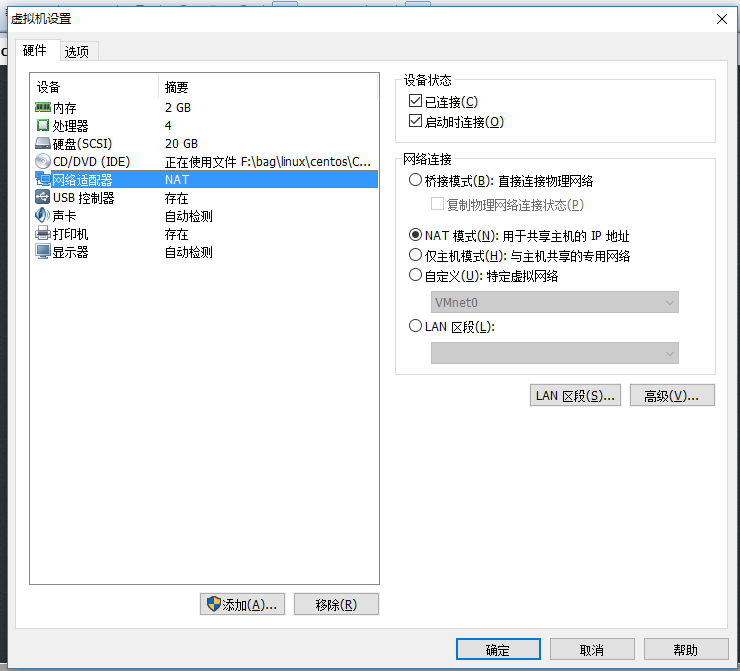
# Linux学习

## 基本介绍

## 网络配置



### 桥接模式

在桥接模式下，VMWare虚拟出来的操作系统就像是局域网中的一台独立的主机（主机和虚拟机处于对等地位），它可以访问网内任何一台机器。在桥接模式下，我们往往需要为虚拟主机配置ＩＰ地址、子网掩码等（注意虚拟主机的ｉｐ地址要和主机ｉｐ地址在同一网段）。

　　使用桥接模式的虚拟系统和主机的关系，就如同连接在一个集线器上的两台电脑；要让他们通讯就需要为虚拟系统配置ip地址和子网掩码。如果我们需要在局域网内建立一个虚拟服务器，并为局域网用户提供服务，那就要选择桥接模式。

　　但对于想进行种种网络实验的朋友这种模式是不太适合的，因为你无法对虚拟机的网络进行控制，它直接出去了。

### NAT模式

在NAT模式下，虚拟系统需要借助NAT(网络地址转换)功能，通过宿主机器所在的网络来访问公网。也就是说，使用NAT模式虚拟系统可把物理主机作为路由器访问互联网。NAT模式下的虚拟系统的TCP/IP配置信息是由VMnet8(NAT)虚拟网络的DHCP服务器提供的，无法进行手工修改，因此虚拟系统也就无法和本局域网中的其他真实主机进行通讯。采用NAT模式最大的优势是虚拟系统接入互联网非常简单，你不需要进行任何其他的配置，只需要宿主机器能访问互联网即可。

　　这种方式也可以实现Host OS与Guest OS的双向访问。但网络内其他机器不能访问Guest OS，Guest OS可通过Host OS用NAT协议访问网络内其他机器。NAT方式的IP地址配置方法是由VMware的虚拟DHCP服务器中分配一个IP ，在这个IP地址中已经设置好路由，就是指向192.168.138.1的。如果你想利用VMWare安装一个新的虚拟系统，在虚拟系统中不用进行任何手工配置就能直接访问互联网，建议你采用NAT模式。

　　这种方式下host内部出现了一个虚拟的网卡vmnet8（默认情况下），如果你有过 做nat服务器的经验，这里的vmnet8就相当于连接到内网的网卡，而虚拟机本身则相当于运 行在内网上的机器，虚拟机内的网卡（eth0）则独立于vmnet8。

　　你会发现在这种方式下，vmware自带的dhcp会默认地加载到vmnet8界面上，这样虚拟机就可以使用dhcp服务。更为重要的是，vmware自带了nat服务，提供了从vmnet8到外网的地址转 换，所以这种情况是一个实实在在的nat服务器在运行，只不过是供虚拟机用的。很显然，如果你只有一个外网地址，此种方式很合适。

### 仅主机模式

　主机模式下，真实环境和虚拟环境是隔离开的；在这种模式下，所有的虚拟系统是可以相互通信的，但虚拟系统和真实的网络是被隔离开。（虚拟系统和宿主机器系统是可以相互通信的，相当于这两台机器通过双绞线互连。）

　　这应该是最为灵活的方式，有兴趣的话可以进行各种网络实验。和nat唯一的不同的是，此种方式下，没有地址转换服务，因此，默认情况下，虚拟机只能到主机访问。

### 临时修改ip地址，重启后失效

Ifconfig eno16777736(网卡名称有的是eth0) 192.168.31.100

ifconfig eno16777736 192.168.31.200

### Ping不通虚拟机

#### 检查虚拟机网络适配器模式,仅主机模式使用VMNet8网卡,注意检查子网ip,子网掩码,dhcp ,NAT模式使用VMNet1网卡,注意检查子网ip,子网掩码,dhcp 等

#### 修改ip地址

进入/etc/sysconfig/network-scripts目录(下面是目录内的文件)

ifcfg-eno16777736 ifdown-post ifup-eth ifup-sit

ifcfg-lo ifdown-ppp ifup-ib ifup-Team

修改ifcfg-eno16777736为

BOOTPROTO="static" #dhcp改为static

ONBOOT="yes" #开机启用本配置

IPADDR=192.168.31.100 #静态IP

GATEWAY=192.168.31.1 #默认网关

NETMASK=255.255.255.0 #子网掩码

### 查看linux版本

lsb\_release -a

查看内核版本

cat /etc/redhat-release

### 修改主机名称

#### 修改主机名称

vim /etc/sysconfig/network

修改下面的hostname属性值即可

NETWORKING=yes

HOSTNAME=elasticsearch1

#### 将修改后的hostname与主机进行关联

vim /etc/hosts

添加一下内容

10.0.6.34 主机名称1

10.0.6.36 主机名称2

10.0.6.37 主机名称3

10.0.6.38 主机名称4

10.0.6.39 主机名称5

然后重启机器即可

重启后测试ping 主机名称1 发现可以ping通

## 安装jdk

whereis java查看jdk安装路径

which java （java执行路径）

编辑/etc/profile文件

export JAVA\_HOME= /usr/rkg/jdk/jdk1.8.0\_111  
export PATH=$JAVA\_HOME/bin:$PATH  
export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

**source/etc/profile**是让/etc/profile文件修改后立即生效，

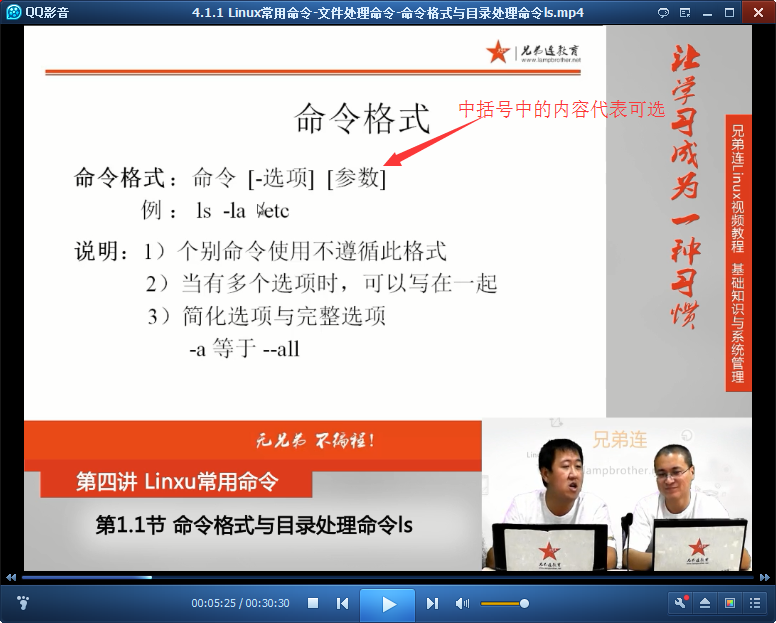
还有一种方法是**：. /etc/profile**

注意：.和/etc/profile有空格

使修改后的/etc/profile文件生效

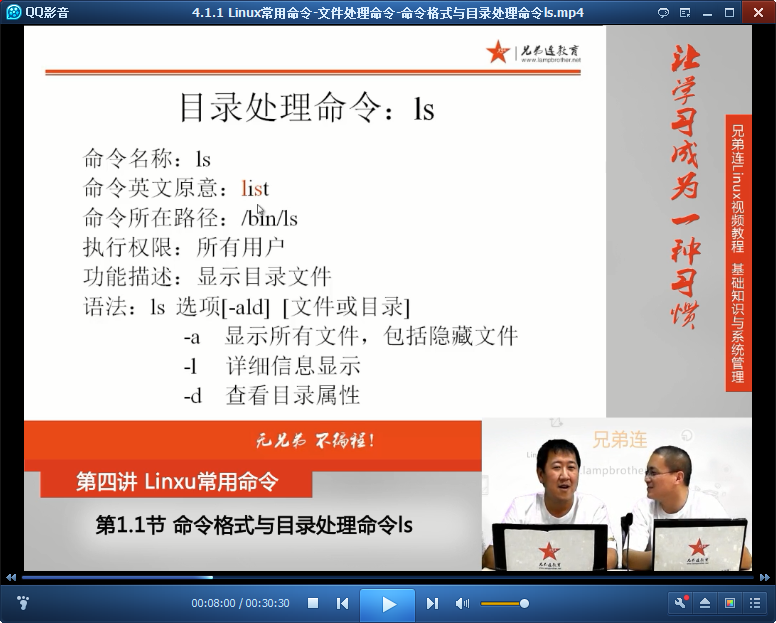
## Linux常用命令

### 文件处理命令



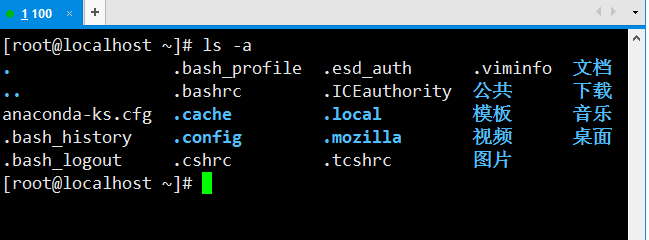
命令格式与目录处理命令ls

#### ls查看目录下的内容

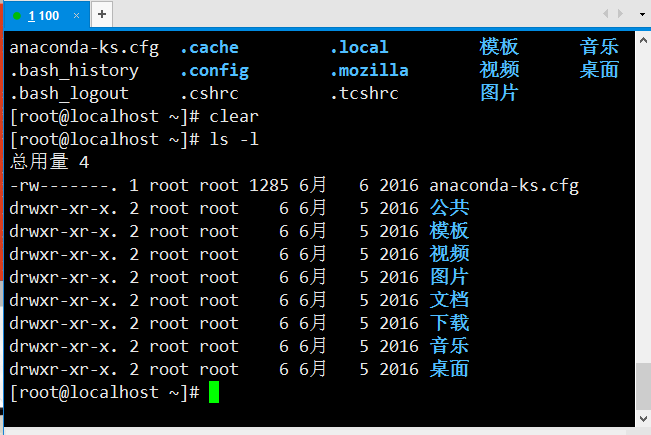


#### ls -a查看当前目录下所有的文件(包括隐藏文件) [root@localhost ~]# ls –a

注:以点(.)开头的文件是隐藏文件



#### ls –l以长格式的形式查看当前目录下所有可见文件的详细属性。[root@localhost ~]# ls -l



实例说明

总用量 4代表当前目录占了4k的空间(文件夹大小),默认单位kb,可以使用ls –lh来显示更加人性化的数据格式

drwxr-xr-x. 2 root root 6 6月 5 2016 桌面

d:文件类型(-:二进制文件,d:目录,l:软链接文件)

rwx:(文件所有者的操作权限),r:读,w:写,x:执行

r-x: (文件所属组的操作权限) ,r:读,w:写,x:执行

r-x: (其他用户的操作权限)

2:当前文件被引用的次数

第一个root:当前文件的所有者(一般指文件创建者),所有者只能有一个

第二个root:当前文件所属组,所属组只能有一个

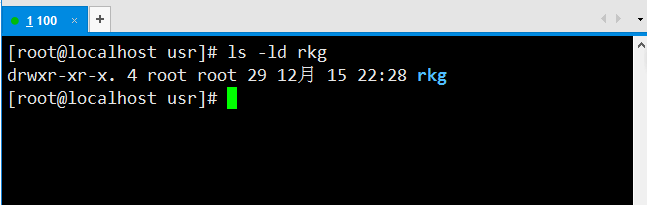
6:文件大小(默认单位byte),可以使用ls –lh进行人性化显示大小(根据大小显示K,M,G等)

6月 5 2016:文件最后修改时间

桌面:文件名称

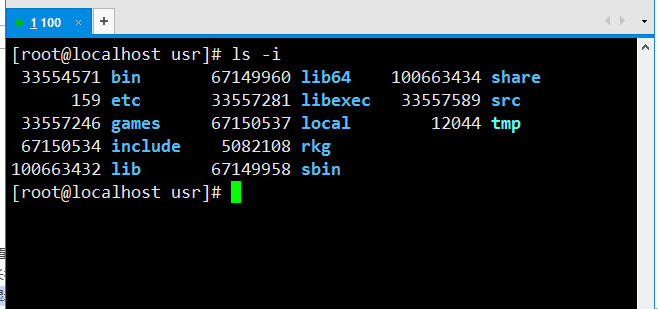
注:ls –lh /usr/rkg 显示/usr/rkg目录下的内容

#### ls –ld 显示目录的信息



ls –ld rkg 显示当前usr目录下rkg目录的信息

#### ls -i输出文件i (inode)节点的索引信息。(当前文件在系统中的唯一标识号)



#### mkdir创建文件夹



### 权限管理命令

#### 新增用户

useradd rkg

#### 设置密码

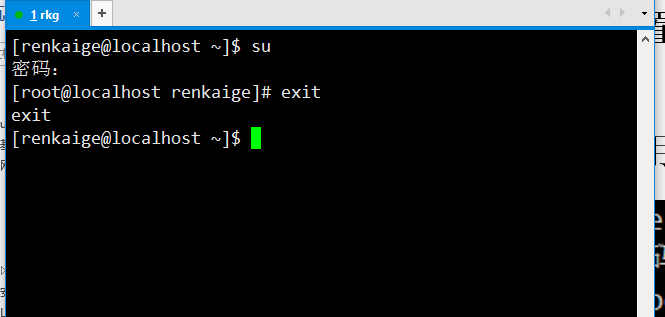
passwd rkg

#### 删除用户

**userdel -r rkg**

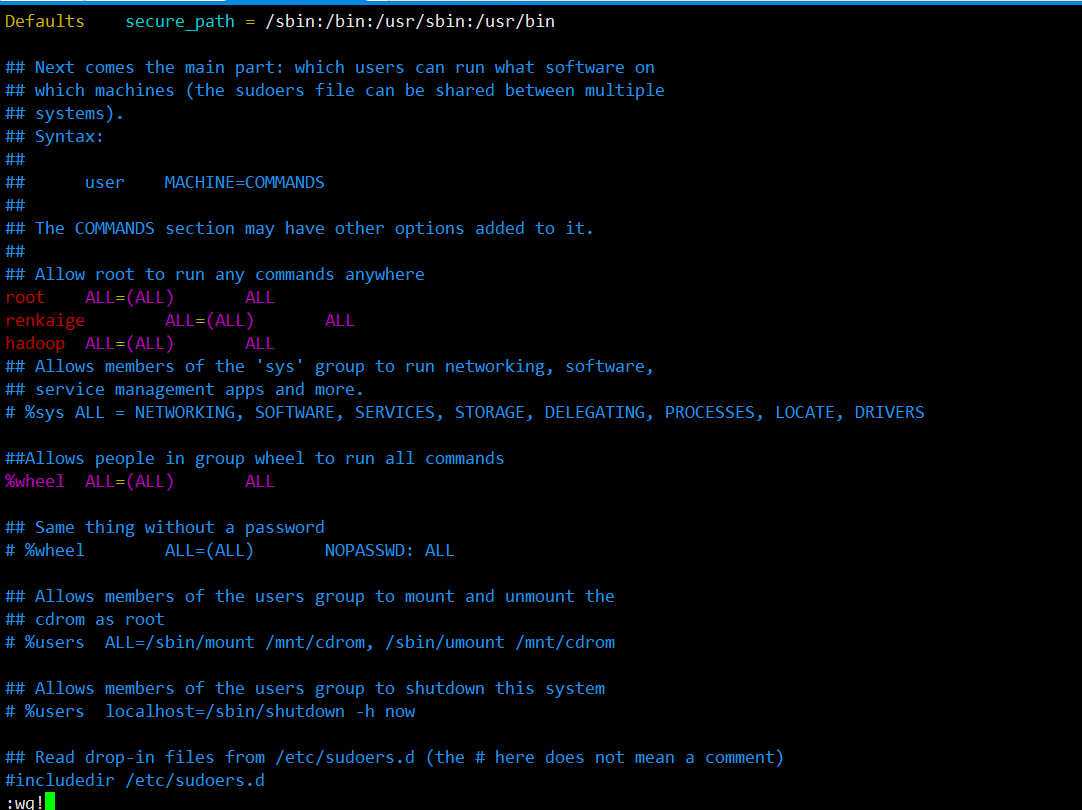
#### 设置sudo权限

1:使用su命令切换为root用户,不建议使用该命令不安全



2:使用sudo前需要先为用root用户为该目录添加sudo权限

需要在/etc/sudoers中添加如下(hadoop那一行)



### 文件搜索命令

### 网络命令

### 关机重启命令

### scp命令

scp zookeeper-3.4.6.tar.gz root@10.0.6.37:/home/rkg/zookeeper/

### 查看端口占用

lsof -i tcp:2181

netstat -ano|grep 80

[root@slave4 bin]# lsof -i tcp:2181

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

java 1708 root 23u IPv6 13505 0t0 TCP \*:eforward (LISTEN)

[root@slave4 bin]

netstat –ntlp列出所有端口

### 查看程序是否启动

ps aux|grep nginx

ps –ef|grep nginx

## 文本编辑器Vim

### 常用操作

#### yyp复制当前所在行

或者直接按一下p

#### dd快速删除当前行

#### 比较两个文件

diff 文件A 文件B

vimdiff 文件A 文件B

### 使用技巧

## shell编程

shell是用户与内核进行交互操作的一种接口,目前最流行的shell称为bash shell,shell也是一门编程语言<解释型的编程语言>,即shell脚本<就是在用linux的shell命令编程>,一个系统可以存在多个shell,可以通过cat /etc/shells命令查看系统中安装的shell,不同的shell可能支持的命令语法是不相同的

### 基本语法

代码写在普通文本文件中,通常以.sh为后缀名

vi helloworld.sh

#!/bin/bash ##表示用哪一种shell解析器来解析执行我们的这个脚本程序

echo “hello world”

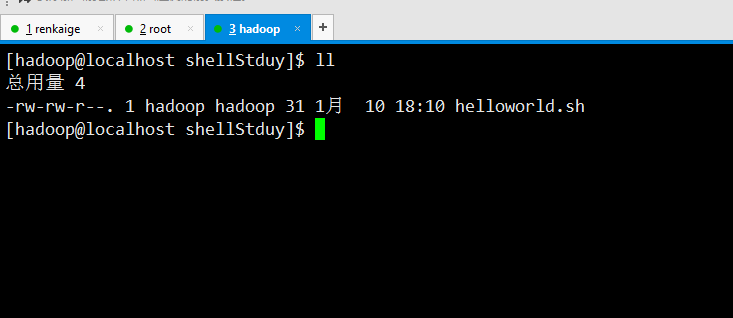
#### 执行脚本

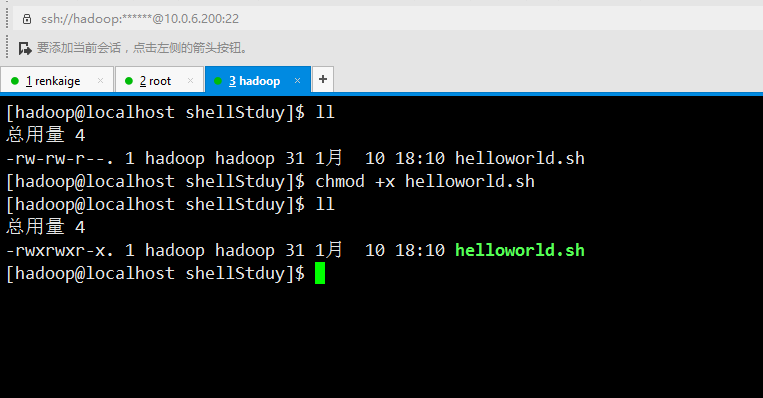
./helloworld.sh不能直接helloworld.sh这样会去系统环境变量中查找helloworld.sh文件,找不到会报错

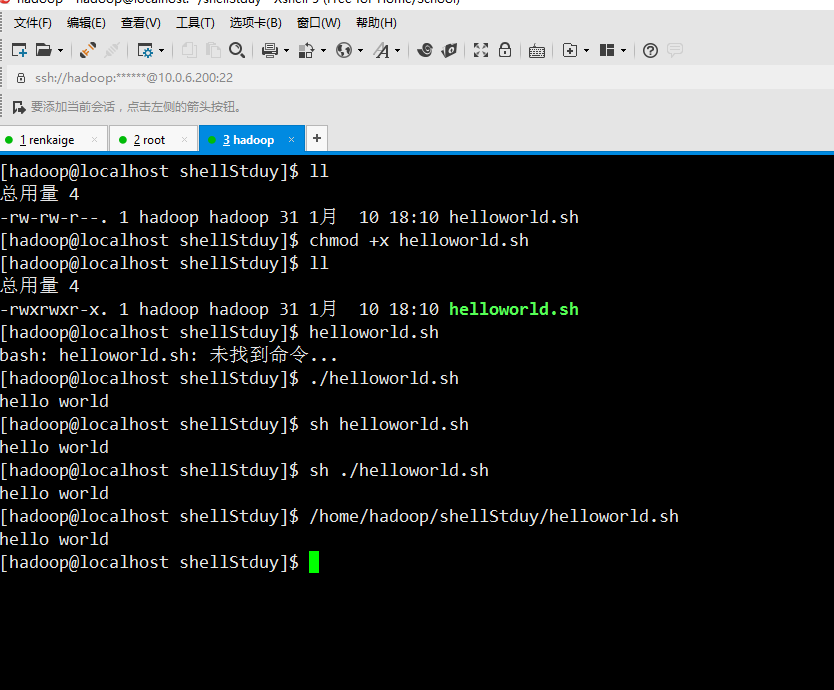
sh helloworld.sh

#### 注:为文件添加执行权限(当前文件只有读写权限)

chmod +x helloworld.sh







#### 系统变量

linux shell中的变量分为系统变量和用户自定义变量,可以通过set命令查看系统变量

查看变量 echo $变量名

[hadoop@localhost /]$ echo $JAVA\_HOME

/usr/renkaige/jdk/jdk1.8.0\_73

[hadoop@localhost /]$

#### 自定义变量

变量=值(例如STR=abc)

等号两侧不能有空格(有空格会被解析成命令执行,找不到命令会报错)

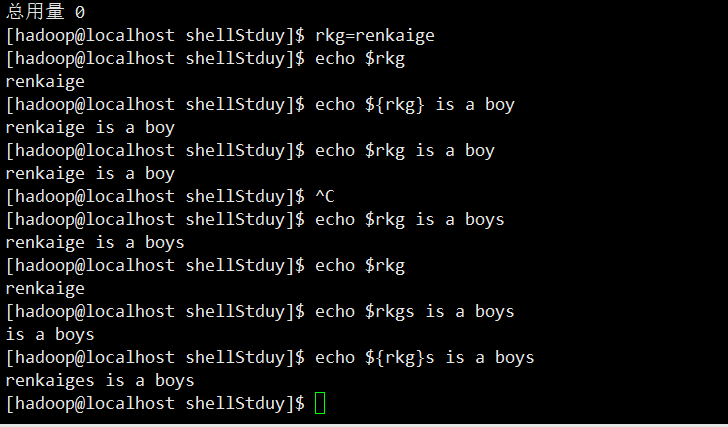
变量名称一般习惯大写

使用变量$STR

双引号和单引号有区别:双引号仅将空格脱意,单引号会将变量引用比如$STR脱意

#### 示例

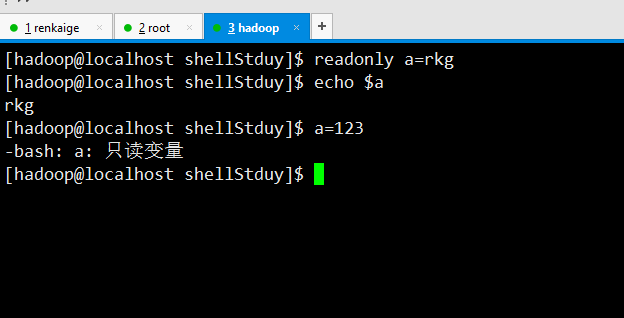
变量和其他字符一块使用时需要使用大括号{}



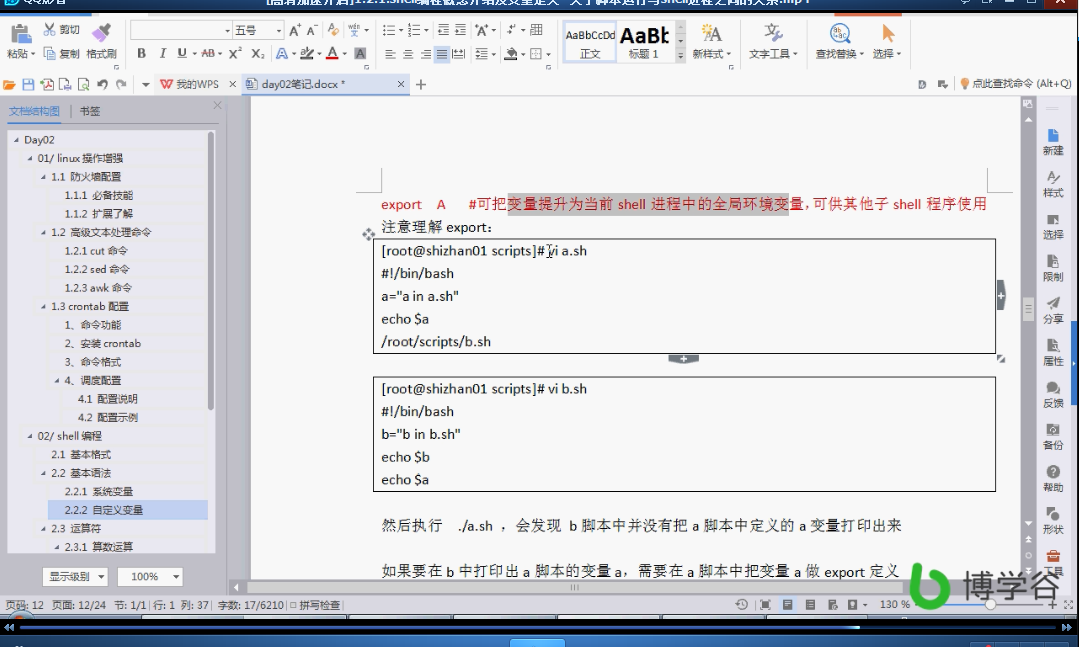
#### 删除变量

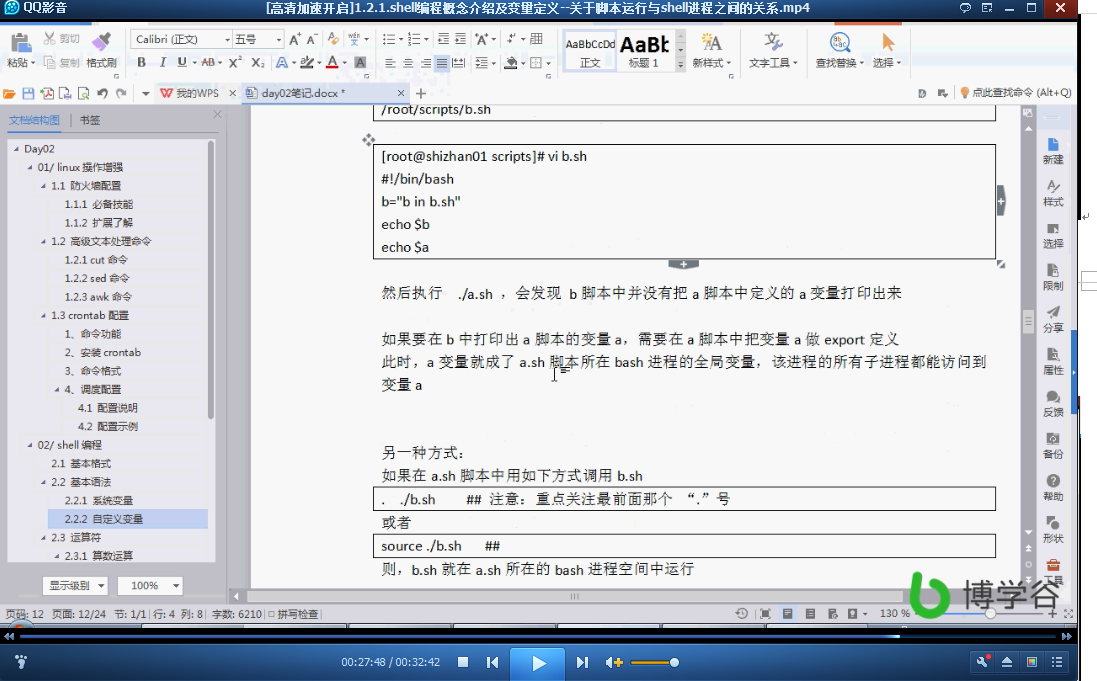
unset 变量名称

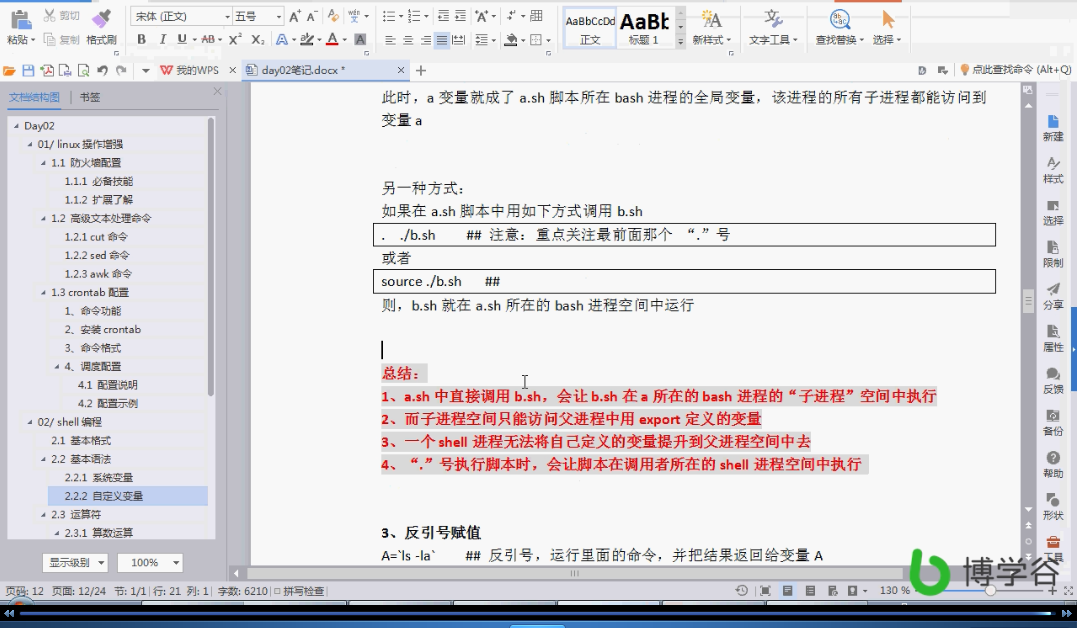
#### 定义常量(不可修改值)



#### export可将变量提升为当前shell进程中的全局环境变量,可供其他子shell程序使用







### 算数运算

#### 使用expr进行运算(每两个元素之间必须要有空格)

[hadoop@localhost shellStduy]$ expr 1 +2

expr: 语法错误

[hadoop@localhost shellStduy]$ expr 1 + 2

3

[hadoop@localhost shellStduy]$

将运算后的值,赋给一个变量(注意expr前面不是单引号是反引号,键盘tab键上面数字1左边那个键)

[hadoop@localhost shellStduy]$ a=`expr 5 + 6`

[hadoop@localhost shellStduy]$ echo $a

11

#### 使用(())进行运算

[hadoop@localhost shellStduy]$ a=$((1+2))

[hadoop@localhost shellStduy]$ echo $a

3

[hadoop@localhost shellStduy]$ a=$(((1+2)\*6))

[hadoop@localhost shellStduy]$ echo $a

18

#### 使用$[]进行运算

[hadoop@localhost shellStduy]$ a=$[1+6]

[hadoop@localhost shellStduy]$ echo $a

7

[hadoop@localhost shellStduy]$ a=$[(1+6)\*2]

[hadoop@localhost shellStduy]$ echo $a

14

### 流程控制

#### if控制

#!/bin/bash

read -p "请输入您的名字" NAME ##read命令用户从控制台获取用户输入的数据,NAME代表用户输入的数据赋值给它,-p “请输入您的名字”控制台提示的内容

##printf '%s\n'$NAME

if [ $NAME = root ]

then

echo "欢迎你${NAME}!"

elif [ $NAME = rkg ]

then

echo "欢迎你任凯歌"

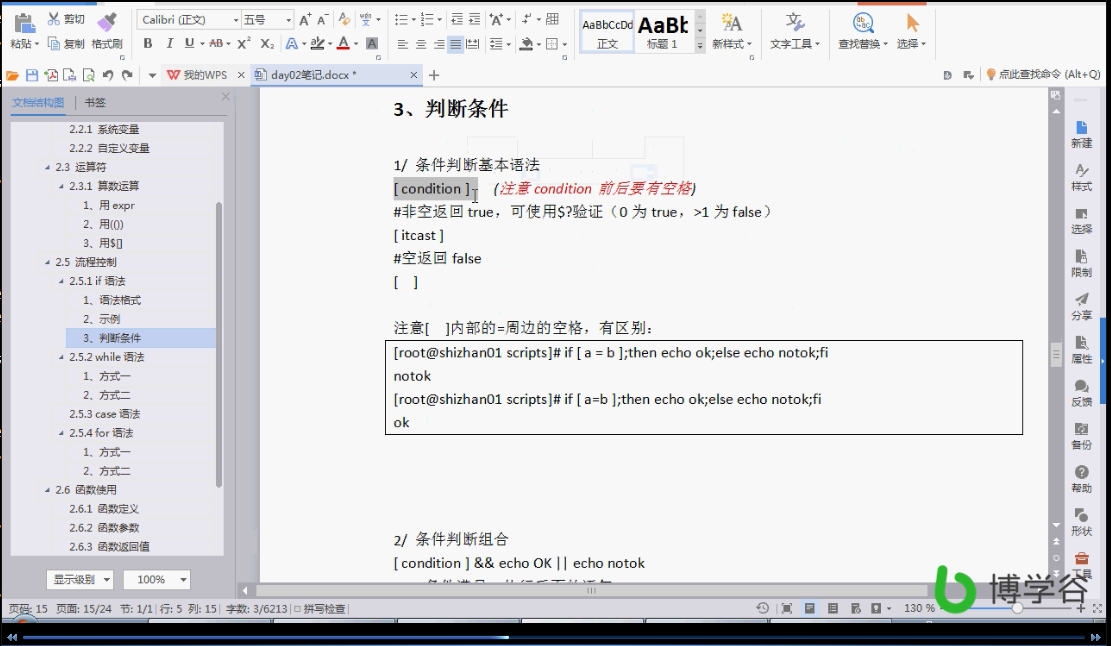
else

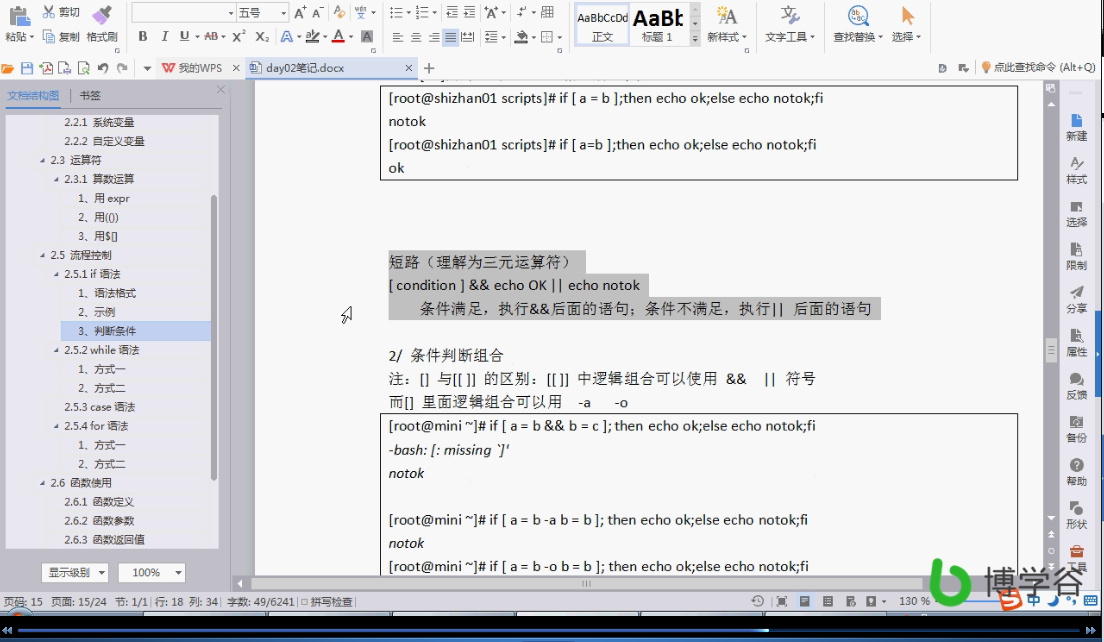
echo "对不起不认识你"

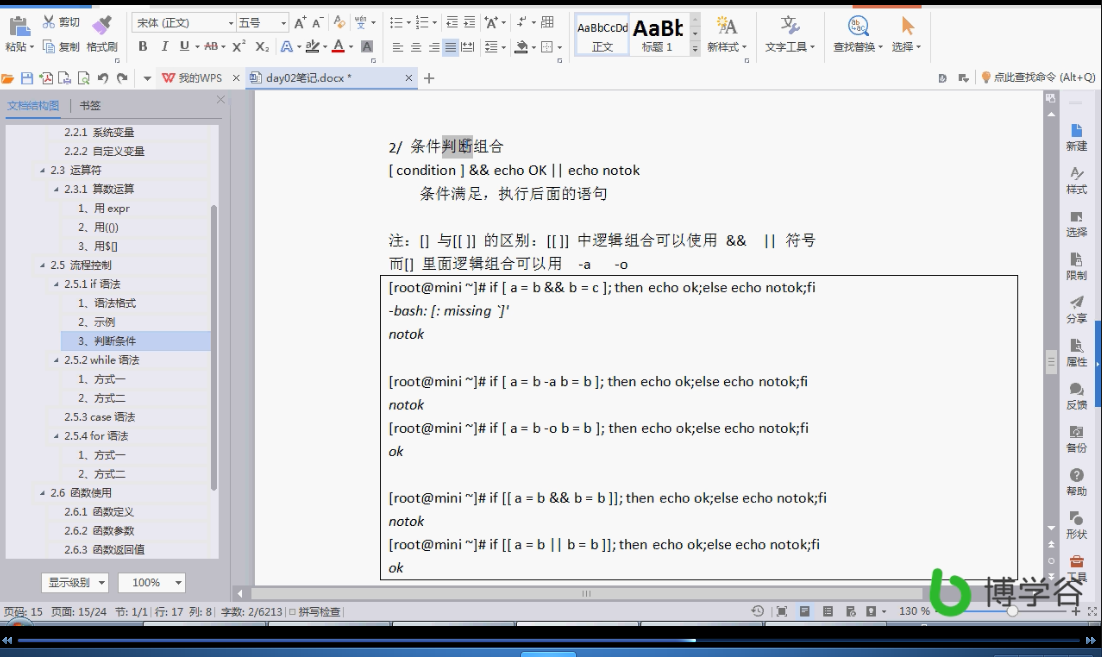
fi

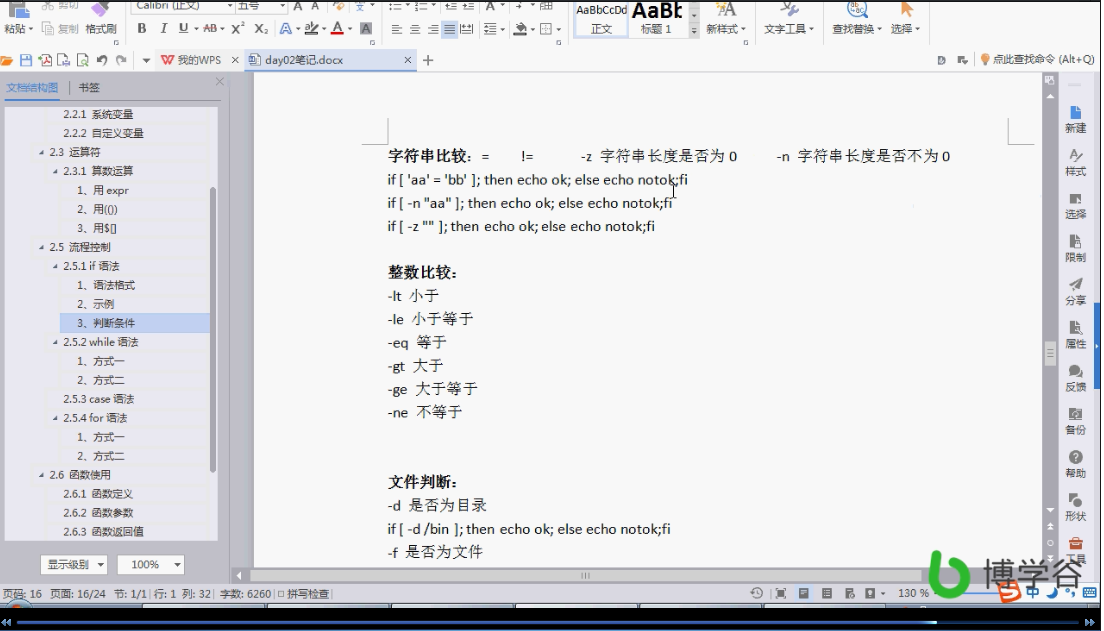
注意:if,elif后面,[]中间等有空格

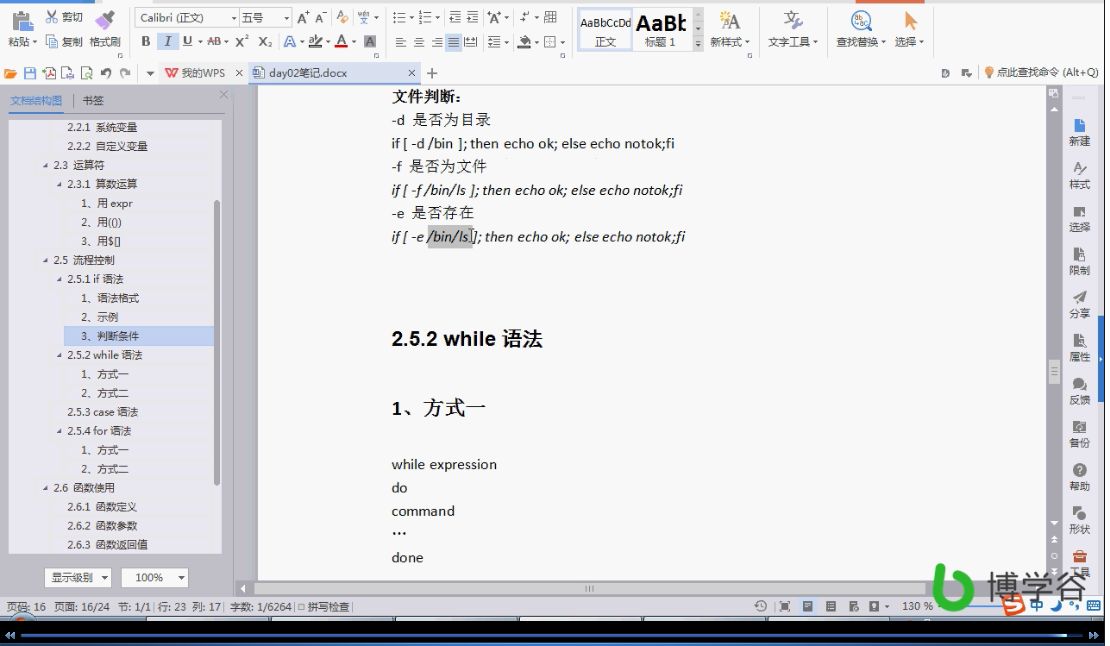
##### 语法











#### 函数

## 配置ssh无密登录

### 安装ssh

centos7窗口化默认已经安装了ssh(客户端)和sshd(服务端)

### 生成秘钥对

ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa ##-t rsa代表加密类型使用rsa类型,-P ''代表无密登录,-f ~/.ssh/id\_rsa 代表密钥文件存放位置

### 将生成的公钥添加到对方的认证密钥库中

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

### centos失败情况总结

在centos中当配置完以上步骤后,还是不能免密登录时,需要修改一下内容

1:vim /etc/ssh/sshd\_config

将下面内容的注释打开

RSAAuthentication yes

PubkeyAuthentication yes

AuthorizedKeysFile .ssh/authorized\_keys

2: vim /etc/selinux/config

将SELINUX的值改为disabled

SELINUX=disabled

3:重启计算机