# 用户

## 添加用户

useradd [options] 用户名

options:

-c comment 指定一段注释性描述。

-d 目录 指定用户主目录，如果此目录不存在，则同时使用-m选项，可以创建主目录。

-g 用户组 指定用户所属的用户组。

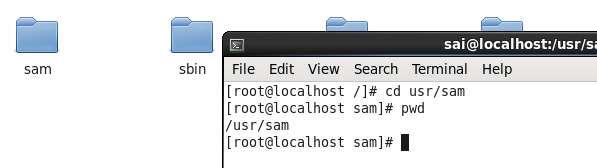
-G 用户组，用户组 指定用户所属的附加组。

-s Shell文件 指定用户的登录Shell。

-u 用户号 指定用户的用户号，如果同时有-o选项，则可以重复使用其他用户的标识号。

案例1.





2.创建用户，当创建的用户无法登陆

useradd username –s /sbin/nologin

## 删除用户

userdel [options] 用户名

options:

-r 将用户的主目录删除 userdel –r 用户名

-f 强制删除用户(甚至用户已经登录Linux系统，此命令任然有效)。

## 用户口令

用户管理的一项重要内容是用户口令的管理。用户账号刚创建时没有口令，但是被系统锁定，无法使用，必须为其指定口令后才可以使用，即使是指定空口令。

指定和修改用户口令的Shell命令是passwd。超级用户可以为自己和其他用户指定口令，普通用户只能用它修改自己的口令

命令格式：passwd [options] 用户名

options:

-l 锁定口令，即禁用账号。

-u 口令解锁。

-d 使账号无口令。

-f 强迫用户下次登录时修改口令。

--stdin

# 用户组

## 添加用户组

## 修改用户组

usermod –g 用户组 用户名

只有超级用户使用的命令：chown chgrp

chown（change owner）修改文件/目录所有者和组

chgrp 对文件的组进行修改。

/etc/profile:初始化系统全局Shell变量

/etc/profile.d

/etc/bashrc:定义用于Shell函数和别名的系统全局变量

~/.bash\_profile:用户个人的环境设置文件

~/.bash：用户个人的别名及变量设置文件

~/.bash\_logout:系统退出设置文件

Source:立即使用shell环境设置文件

**命令行解析：**

子进程执行命令时，根据下列顺序判断命令类型：

1. 别名
2. 关键字
3. 函数
4. 内置命令
5. 可执行文件

# 网络篇

## 网卡重命名

* 1.修改grub2启动参数，在GRUB\_CMDLINE\_LINUX的中加上"net.ifnames=0 biosdevname=0"的参数  
  vi /etc/sysconfig/grub  
  GRUB\_CMDLINE\_LINUX=”rd.lvm.lv=vg0/swap vconsole.keymap=us crashkernel=auto vconsole.font=latarcyrheb-sun16 **net.ifnames=0 biosdevname=0** rd.lvm.lv=vg0/usr rhgb quiet”
* 2.重新加载到启动中  
  grub2-mkconfig -o /boot/grub2/grub.cfg
* 3.重新对网卡配置文件进行命名（网卡文件全部重命名，顺便修改配置文件NAME、DEVICE的名称）  
  mv /etc/sysconfig/network-scripts/ifcfg-enp0s3 /etc/sysconfig/network-scripts/ifcfg-eth0
* 4.reboot重启生

## 一个网卡接口配置多个IP

**方法一：**命令（临时配置，重启后就失效）

通过网路别名来实现：ethx:y 如：eth0:1,eth0:2,eth0:3等。

如：eth0:1 192.169.253.130/24

**方法二：**配置文件：

/etc/sysconfig/network-scripts

vi ifcfg-eth0:1

DEVICE=eth0:1

BOOTPROTO=static

IPADDR=192.168.253.130

NETMASK=255.255.0.0

ONBOOT=yes

**配置文件有两类（/etc/sysconfig/network-scripts/）**

配置IP、掩码、网关

以太网：ifcfg-ethX

PPP：ifcfg-pppX

**匹配路由**

route-ethX

route-pppX

**配置IP、掩码、网关:ethX**

DEVICE="eth0"

此配置关联至的设备。设备名要与文件ifcfg-后ude内容保持一致

BOOTPROTO=none

引导协议：{none|static|dhcp|bootp}

HWADDR="00:0C:29:26:62:92"

MAC地址：要与真实MAC地址保持一致，可省略

NM\_CONTROLLED="yes"

是否接受NetworkManager脚本控制：{yes|no}

ONBOOT="yes"

是否开机自动启动此网络设备{yes|no}

TYPE="Ethernet"

设备类型Etheraget Bridge（桥接）

UUID="14351f7f-a726-4dfc-966e-dfb1f352f226"

唯一标识，可省略

IPADDR=

ip地址

NETMASK=

掩码

GATEWAY=

默认网关

DNS1=

DNS1服务地址

IPV6INIT=no

是否开启ipv6

USERCTL=no

是否允许普通用户操作网卡

PEERDNS={yes|no}

是否允许DHCP服务分配地址时直接更新/etc/resolv。conf中的DNS服务器地址

**配置路由：route-ethX**

配置文件的格式1：每行一个要配置的路径条目

目标 via 下一跳

配置文件的格式2：每一组一个路由条目

ADDRESS#=目标

NETMASK#=掩码

GATEWAY#=下一跳

ADDRESS0=192.168.0.0

NETMASK0=255.255.255.0

GATEWAY0=172.16.100.1

**如何实现在单网卡上配置多个地址：**

通过网络接口别名来实现

eth0X

eth0：X eth0:0 eth0:1....

配置1：

ifconfig ethX:Y IP/mask

配置2：

ifcfg-ethX:Y

DEVICE=ethX:Y

BOOTPROTO={none|static}

IPADDR=

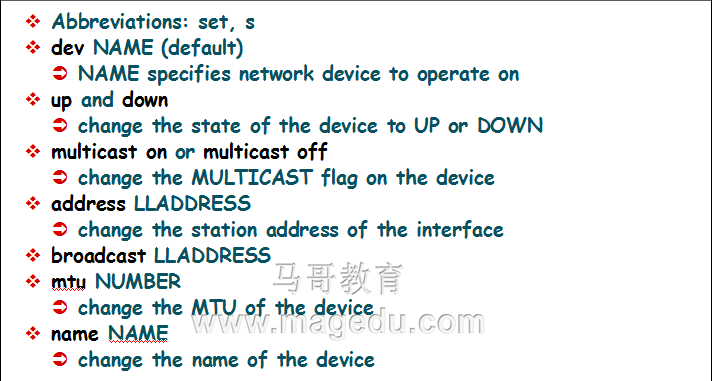
NETMASK=

GATEWAY=

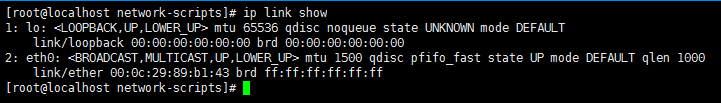
非主地址不支持DHCP协议获取

## iproute2

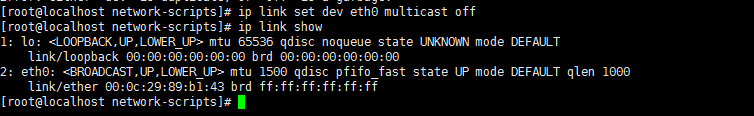
ip link： 网络设备配置命令，如可以启用/禁用某个网络设备，改变mtu及mac地址等。



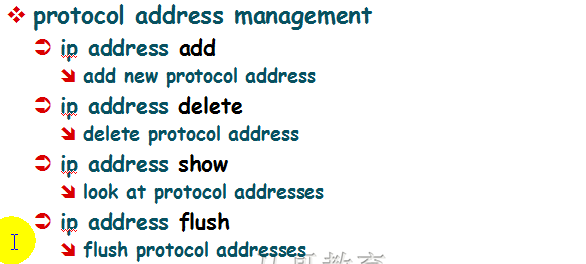
Ip link show：显示网卡接口信息



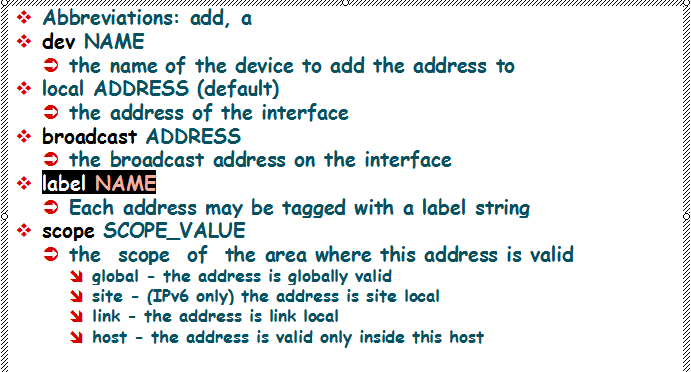
Ip link set dev NAME（default）



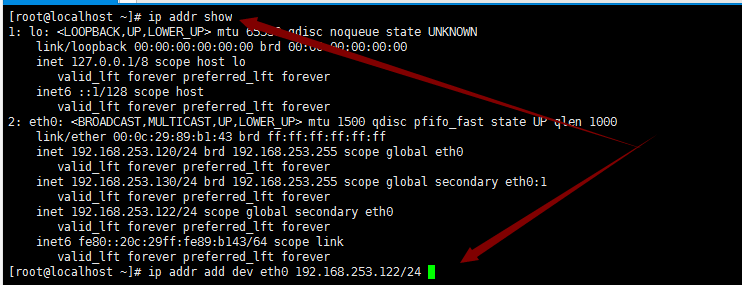
ip addr：用于管理某个网路设备与协议（IP或IPv6）有关的地址，与Ip link类似，不过增加了协议有关的管理。



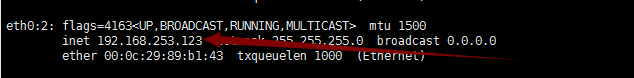
Ip addr add



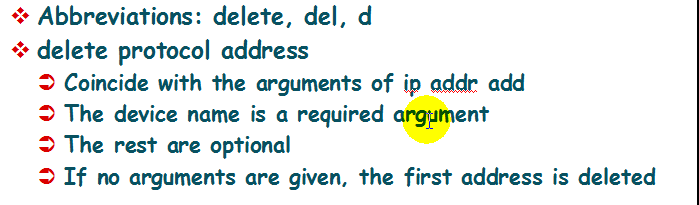
添加地址：ip addr add eth0 192.168.253.122/24



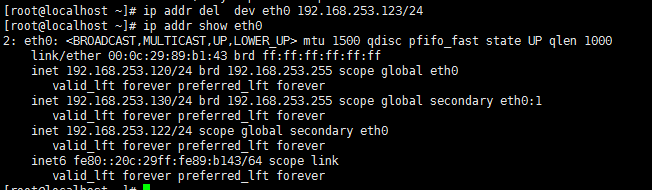
添加地址并指定别名：ip addr add dev eth0 192.168.253.123/24 label eth0:2



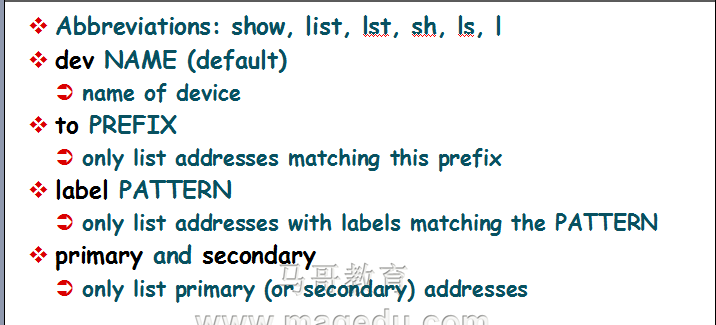
Ip addr delete



删除指定的IP:



Ip addr show



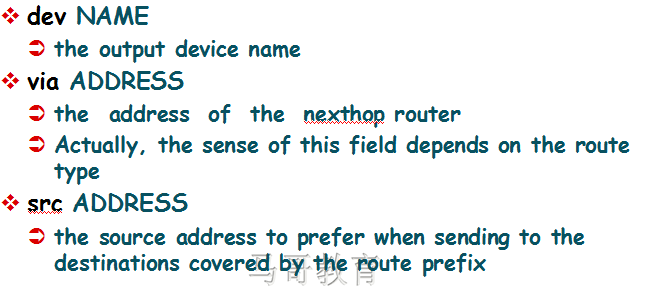
Ip add flush eth0 to 192.168.253.122

ip addrlabel：IPV6的地址标签，主要用于RFC3484中描述的IPV6地址的选择，RFC3484主要介绍了两个算法，用于IPV6地址的选择策略。

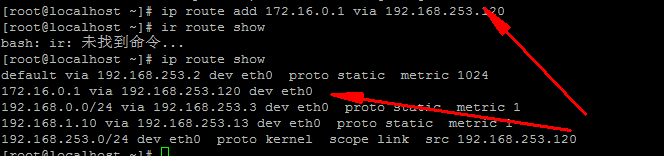
ip route：管理路由，如添加，删除等。



ip route add



添加路由信息：添加172.16.0.1网路 ，下一跳192.168.253.20



Ip route add DESTINATION [via NEXT\_HOP] [src SOURCE\_ADDR][dev DEVICE]

ip rule：管理路由策略数据库，这里面有一个算法，用于控制路由的选择策略。

ip neigh：用于neighbor/arp表的管理，如显示插入，删除等。

ip tunnel：隧道配置，隧道的作用是将数据封装成ip包然后在互联网上讲包发出

ip maddr：多波地址管理

ip mroute：多波路由管理。

ip monitor：状态监控，如可以持续监控ip地址和路由的状态。

**启用/禁用接口：**

ip link set 接口 up|down

ifconfig 接口 up|down

ifdown 接口，ifup 接口

重置网络连接

**网络管理或检测等相关的工具：**

ping：ICMP

ping [options] ip

-c 次数

-w 测试执行时长

traceroute HOST:

获取当前主机到目标主机所经过的网关

mtr HOST

netstat

-t：tcp协议的连接

-u：udp协议的链接

-l：监听状态的连接

-a：所有状态的连接

-p：连接相关的进程

-n：数字格式显示

-r: 显示路由表，类似于route或ip route show

常用组合：

netstat -tan

netstat -tunl

netstat -rn

ss,用于取代netstat

-t：tcp协议的连接

-u：udp协议的链接

-l：监听状态的连接

-a：所有状态的连接

-e：显示扩展信息

-m：显示套接连接使用的内存信息

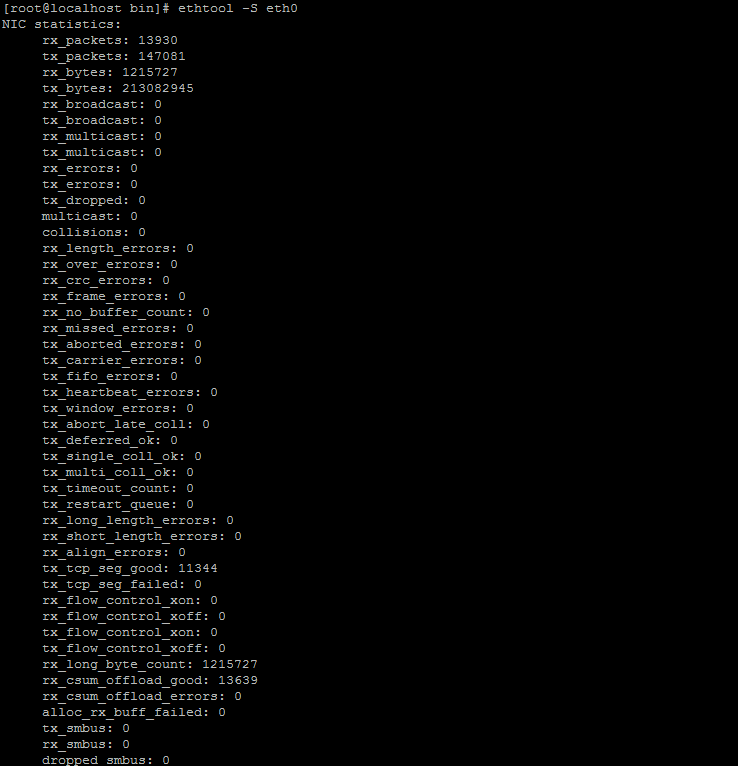
-p：进程及UDP

-n：数字格式显示

-o state (established)

ethtool 接口: 显示接口设备属性

ethtool -S 接口：显示设备接口的统计数据



**修改hostname**

修改 /proc/sys/kernel/hostname文件

# 进程

**进程管理命令**

ps pstree pidof top htop pmap vmstat dstat

## top命令

M:按内存占用百分比大小排序

P：按cpu占用百分比大小排序，默认

T：按累计占有时长进行排序

1：分开显示

l: 是否显示负载信息

t：是否显示cpu和进程的统计信息

m：是否显示内存和交互分区的信息

q：退出

k：kill，杀掉进程

s：改变top刷新频率

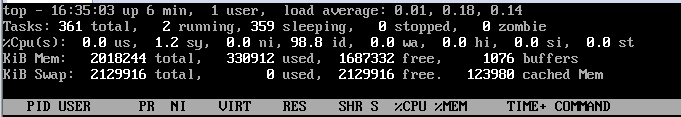
**常用选项：**

-d #:指定刷新时间间隔

-b：以批次显示top的刷新

-n # 指定指定的次数

top命令是Linux下常用的性能分析工具。



top命令输出可以分为两部分：前部分是系统统计信息，后部分是进城信息。

**在统计信息中：**

第一行是任务队列信息，它的记过等同于uptime命令，从左到右一次表示，系统当前时间，系统运行时间，当前登录用户数，最后的load average表示系统的平均负载，即任务队列的平均长度，这3个值分别表示：1分钟，5分钟，15分钟。

第二行是进程统计信息，分别为正在运行的进程数，睡眠进程数，停止的进程数，僵尸进程数。

第三行是CPU统计信息，us表示用户空间的CPU占用率，sy表示内核空间CPU占用率，ni表示用户进程空间改变过优先级的进程CPU的占用率，id表示空闲CPU占用率，wa表示等待输入输出的CPU时间百分比，hi表示硬件中断请求，si表示软件中断请求。

在Men行中，从左到右，依次表示物理内存容量，已使用的物理内存，空闲物理内存，内核缓冲使用量。

Swap行依次表示交换区总量，空闲交换区大小，缓冲交换区大小。

top命令的第2部分是进程信息区，显示了系统内各个进程的资源使用情况：

PID:进程ID

USER：进程所有者

PR：优先级

NI：nice值，负值表示高优先级，正值表示优先级底

%CPU：上次更新到现在CPU时间占用百分比

TIME+：进程使用CPU时间总计，单位1/100秒

VIRT：进程使用的虚拟内存总量，单位kb，VIRT=SWAP+RES

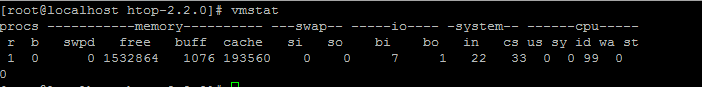
%MEM：进程使用的物理内存百分比

RES：进程使用的，未被换出的物理内存大小，单位kb，RES=CODE+DATA

SHR：共享内存大小，单位kb。

COMMAND：命令名/命令行

## vmstat命令



procs

r：运行或等待cpu时间片的进程的个数

b：被阻塞（通常为等待I/O完成）的进程的长度

memory

swpd：从物理内存交互至swap中的数据量，单位KB

free：未使用的内存大小，单位KB

buffer：buffer空间大小，通常与缓存写操作相关（被用来作为缓存的内存数），单位KB

cache：cache空间大小，通常与缓存读操作相关

swap

si：swap in 数据进入swap中的数据量（从磁盘交换到内存的交换页数量），通常是速率。kb/s

so：swap out 数据离开swap中的数据量（从内存交换到磁盘的交换页数量），通常是速率。kb/s

io

bi：block in：从块设备读入的数据量，通常是速率，kb/s

bo：block out：保存至块设备中的数据量，通常是速率，kb/s

system

in：中断发生频率。每秒的中断数

cs：context switch 上下文切换，进程切换（即每秒的上下文切换的次数），通常是速率，kb/s

cpu：

us：用户空间的使用率

sy：内核空间的使用率

id：空闲

wa：等待的

st：被偷走的

### 使用格式

vmstat [delay [counts]]

vmstat 1 每一秒刷新一次

vmstat 1 4：每一秒刷新四次

-s:显示内存统计数据

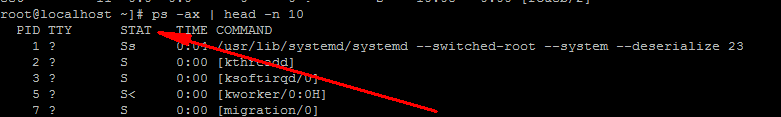
3．dstat

## ps命令

与终端相关的进程：a

与终端无关的进程：x

**ps –ax**



STAT:

R:运行和可运行

S:可中断睡眠

D:不可中断睡眠

T:停止

Z:僵死

s：session leader 有子进程

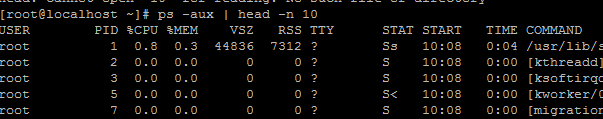
+：前台进程

l：多线程进程

N：低优先级进程

<: 高优先级进程

**ps –aux**



u：以用户为中心显示进程相关信息

VSZ:虚拟内存集，Virutal menmory Size

RSS：常驻内存集，

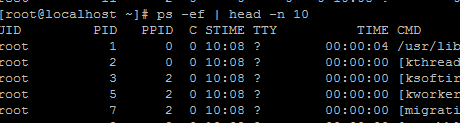
STIME：启动时间

TIME:累计占用的内存时间 []表示内核线程

**ps -ef**

-e：显示所有进程

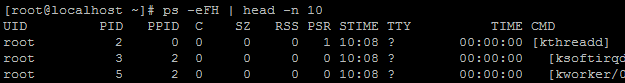
-f：以进程的长格式显示



**ps -eFH**

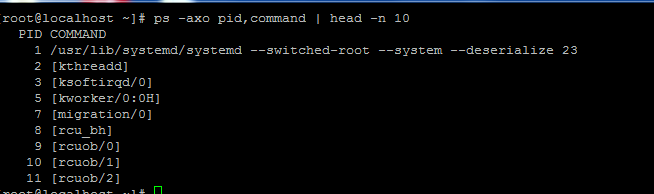
-F：显示额外信息

-H：显示进程的层次信息



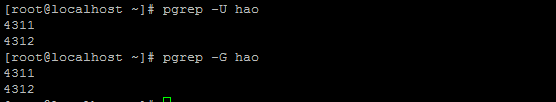
-o：要显示的字段：自定义显示格式

ps –axo pid,comman

  **pgroup：**

-U Username：仅显示指定用户的进程的进程的PID

-G Groupname：仅显示指定用户组的进程的进程的PID



**pidof：**显示指定命令所启动的pid

pidof command

## nice命令

主要用来手动调整进程优先级

默认nice为0，其默认优先级为120

普通用户只能调大此值

对于尚未启动的程序

nice -n # COMMAND 表示进程以#进程启动

对于运行中的进程：

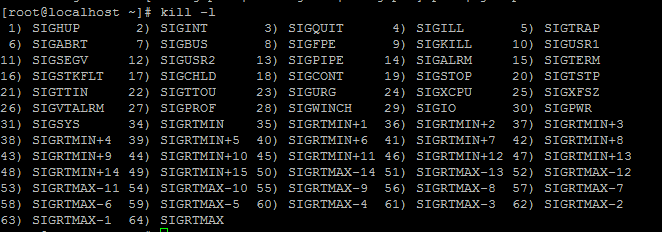
renice # PID 调整PID的nice值

查看 ps axo ni command pid

**进程间通信（IPC）：**

kill命令：可以实现向其他进程发送信息

kill -l ：可以查看向进程发送的所有信息



man 7 sigual

kill -SIGNAL PID

-SIGNAL:

数字代号：比如1，9等

信号名称：如SIGHUP等

简写的信号名称：如HUP等

1) SIGHUP：让程序重读配置文件，而不用重启程序

2) SIGINT：中断信号。Ctrl+c即发送次信号

9) SIGKILL：杀死进程

15) SIGTERM：终止进程，比较优雅（默认）

killall -SIGNAL 进程名

## pmap

## Linux的作业控制

前台作业：占据着一个终端

后台作业：作业执行时不占据终端，作业启动后就释放终端

非守护进程类的程序，启动以后都在前台工作

如果已经启动：前台-->后台。ctrl+z把前台作业送往后台，作业被”停止“

如果尚未启动：COMMAND &

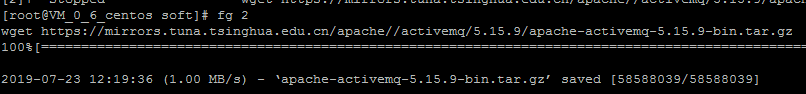
退出当前会话，作业也会终止，因为作业与当前终端相关，如果把作业送往后台后，不期望作业随终止结束而

停止

nohup COMMAND &

如何让送往后台的作业继续执行：

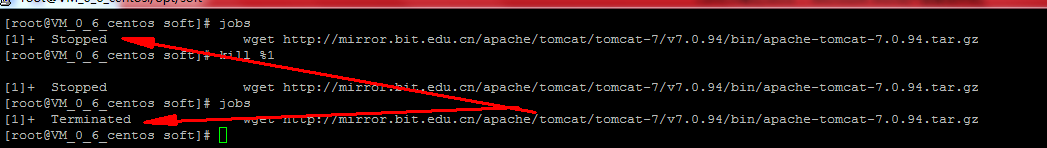
fg [[%]作业号码]：将作业调回前台继续进行



bg [[%]作业号码]：让作业在后台继续进行

默认的为最后一个进入后台的任务

kill %作业号码：终止作业



查看作业号：jobs

