**Linux学习笔记**

# 1 查看硬件信息

查看CPU信息： **lscpu** 或 **cat /proc/cpuinfo**

查看内存信息： **cat /proc/meminfo**

查看硬盘信息： **fdisk -l**

查看网卡信息： **ifconfig**

用于查看无线连接信息： **iwconfig**

查看USB信息：**lsusb**

单独查看内存使用情况：**free -m**

查看内存及cpu使用情况：**top**

# 2 查看操作系统信息

查看当前操作系统内核信息：**uname -a**

查看当前操作系统发行版信息：**cat /etc/issue**

**或cat /etc/centos-release**

显示开机信息：**dmesg**，通常与grep命名来使用管道来找待查信息。

修改主机名：打开配置/etc/hostname修改，然后修改/etc/hosts文件的主机名：127.0.0.1 主机名称

查看系统语言：**locale**

修改系统语言：**export LANG=zh\_CN.UTF-8**

**export LC\_ALL=zh\_CN.UTF-8**

# 3 组和用户

1. 用户（user）的概念

通过前面对Linux 多用户的理解，我们明白Linux 是真正意义上的多用户操作系统，所以我们能在Linux系统中建若干用户（user）。比如我们的同事想用我的计算机，但我不想让他用我的用户名登录，因为我的用户名下有不想让别人看到的资料和信息（也就是隐私内容）这时我就可以给他建一个新的用户名，让他用我所开的用户名去折腾，这从计算机安全角度来说是符合操作规则的；

当然用户（user）的概念理解还不仅仅于此，在Linux系统中还有一些用户是用来完成特定任务的，比如nobody和ftp 等，我们访问LinuxSir.Org 的网页程序，就是nobody用户；我们匿名访问ftp 时，会用到用户ftp或nobody ；如果您想了解Linux系统的一些帐号，请查看 /etc/passwd 。

1. 用户组（group）的概念

用户组（group）就是具有相同特征的用户（user）的集合体；比如有时我们要让多个用户具有相同的权限，比如查看、修改某一文件或执行某个命令，这时我们需要用户组，我们把用户都定义到同一用户组，我们通过修改文件或目录的权限，让用户组具有一定的操作权限，这样用户组下的用户对该文件或目录都具有相同的权限，这是我们通过定义组和修改文件的权限来实现的；

举例：我们为了让一些用户有权限查看某一文档，比如是一个时间表，而编写时间表的人要具有读写执行的权限，我们想让一些用户知道这个时间表的内容，而不让他们修改，所以我们可以把这些用户都划到一个组，然后来修改这个文件的权限，让用户组可读，这样用户组下面的每个用户都是可读的；

1. 用户和用户组的对应关系

**一对一**：某个用户可以是某个组的唯一成员；

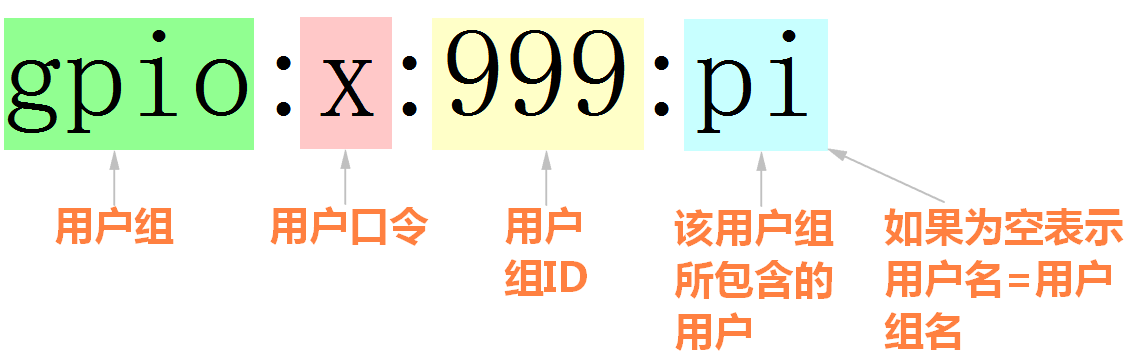
**多对一**：多个用户可以是某个唯一的组的成员，不归属其它用户组；比如beinan和linuxsir两个用户只归属于beinan用户组；

**一对多**：某个用户可以是多个用户组的成员；比如beinan可以是root组成员，也可以是linuxsir用户组成员，还可以是adm用户组成员；

**多对多**：多个用户对应多个用户组，并且几个用户可以是归属相同的组；其实多对多的关系是前面三条的扩展；理解了上面的三条，这条也能理解；

查看当前登录用户名：**whoami** 或 **who am i**

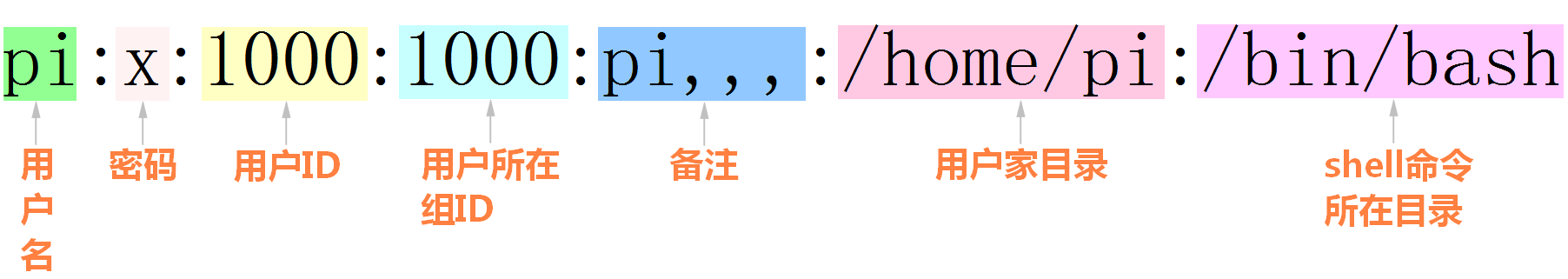
查看所有用户组：**cat /etc/group**



查看组内有哪些用户：**groups**

查看用户信息：**id 用户名**

查看所有的用户信息：**cat /etc/passwd**



**对组操作**

修改组名称： **groupmod -n 新组名 旧组名**

添加组：**groupadd 组名**

把用户增加到工作组：**usermod -G 组名 用户名**

删除组：**groupdel 组名**

从用户组中删除用户：**gpasswd -d 用户名 组名**

修改文件文件的所有者：**chown 用户名 文件名**

修改文件夹下所有文件用户组和组名：**chown -R 用户组.组名 文件夹**

**对用户操作**

添加新用户并添加到组： **useradd -g 用户名 组名**

修改用户密码： **passwd 用户名**

修改用户名： **usermod -l 新用户名 -d /home/新用户名 -m 旧用户名**

|  |
| --- |
| -l 修改用户名  -d 修改登录后使用的路径  -m 修改登录名称的同时将目录名称一同修改 |

**例如：usermod -l vison -d /home/vison -m zhufuyi**

将用户添加到组：**gpasswd -a 用户名 组名**

把用户从组移除：**gpasswd -d 用户名 组名**

删除用户：**userdel 用户名**

用户影子文件：存放各用户的密码，使用不可逆加密方式SHA加密口令

查看用户影子文件内容：cat /etc/shadow

|  |
| --- |
| root:$6$w1kELZF86qOSwzbt$/Zy0C09Dy4XSNGGejyQ1RSjRs1tr6Gp5yGx3IiVNKmI.S0AQXPQVr3dMae2FE16NftxXOwT92b4AEZSrTvKKK/::0:99999:7::: |

etc/shadow文件中域的含义

|  |  |
| --- | --- |
| 域 | 含 义 |
| username | 用户登录名 |
| password | 加密的用户口令 |
| lastchg | 表示从1970年1月1日起到上次修改口令所经过的天数 |
| min | 表示两次修改口令之间至少经过的天数 |
| max | 表示口令还会有效的最大天数，如果是99999则表示永不过期 |
| warn | 表示口令失效前多少天内系统向用户发出警告 |
| inactive | 表示禁止登录前用户名还有效的天数 |
| expire | 表示用户被禁止登录的时间 |
| flag | 保留域，暂未使用 |

切换用户：**su 用户名**

# 4 进入U盘管理文件

1)先用：sudo fdisk -l命令查看u盘所在分区。

2)假设如上分区在/dev/sdb4，使用命令创建usb目录:

mkdir /mnt/usb

使用:mount /dev/sdb4 /mnt/usb 命令载u盘到/mnt/usb目录

使用:umount /mnt/usb命令卸载u盘挂载分区

(注意:卸载的时候不能在当前目录，不然系统将识别为/mnt/usb目录忙碌，无法卸载)。

# 5 解压和压缩

**.tar文件**

解包：tar xvf FileName.tar

打包：tar cvf FileName.tar DirName

**.gz文件**

解包1：gunzip FileName.gz

解包2：gzip -d FileName.gz

打包：gzip FileName

**.tar.gz文件 或 .tgz文件**

解包：tar zxvf FileName.tar.gz

打包：tar zcvf FileName.tar.gz DirName

**.bz2文件**

解包1：bzip2 -d FileName.bz2

解包2：bunzip2 FileName.bz2

打包： bzip2 -z FileName

**.tar.bz2文件**

解包：tar jxvf FileName.tar.bz2

打包：tar jcvf FileName.tar.bz2 DirName

**.bz文件**

解包1：bzip2 -d FileName.bz

解包2：bunzip2 FileName.bz

打包：未知

**.tar.bz文件**

解包：tar jxvf FileName.tar.bz

打包：未知

**.Z文件**

解包：uncompress FileName.Z

打包：compress FileName

**.tar.Z文件**

解包：tar Zxvf FileName.tar.Z

打包：tar Zcvf FileName.tar.Z DirName

**.zip文件**

解压：unzip FileName.zip

压缩：zip FileName.zip DirName

**.rar文件**

解包：rar x FileName.rar

打包：rar a FileName.rar DirName

# 6 查看文件或文件夹空间大小

查看文件大小：du 文件名

查看文件占有空间：du -sh 文件夹

统计当前目录大小，并按大小排序：-sk单位为k，-sm单位为m

du -sk \* | sort -n

统计文件夹下文件个数：du ./golang/\*/\*/\* | wc -l

# 7 时间设置和输出格式化

设置系统时间：

date -s "2016-01-01 12:34:56"

获取系统时间并格式化输出

|  |  |
| --- | --- |
| date "+%Y-%m-%d" | 2016-02-19 |
| date "+%H:%M:%S" | 13:13:59 |
| date "+%Y-%m-%d %H:%M:%S" | 2013-02-19 13:14:19 |
| date "+%Y\_%m\_%d %H:%M:%S" | 2013\_02\_19 13:14:58 |
| date -d today | Tue Feb 19 13:10:38 CST 2016 |
| date -d now | Tue Feb 19 13:10:43 CST 2016 |
| date -d tomorrow | Wed Feb 20 13:11:06 CST 2016 |
| date -d yesterday | Mon Feb 18 13:11:58 CST 2016 |

# 8 设置软件开机启动和监控

首先准备一个执行文件和写好的脚本文件，脚本的意义是判断进程是否存在，不存在则启动程序，存在则忽略，内容如下：

|  |
| --- |
| #! /bin/sh    proc\_name="hello"  proc\_num()  {  num=`ps -A | grep $proc\_name | wc -l`  return $num  }    proc\_num  number=$?  #echo $number  if [ $number -eq 0 ]  then  /home/vison/work/golang/project/src/hello/hello &  fi |

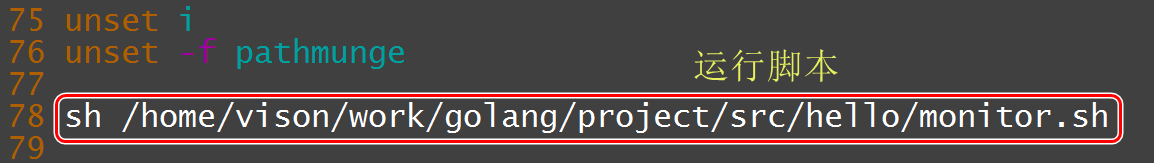
**(1) 设置开机启动**

/etc/profile文件为系统的每个用户设置环境信息，当用户第一次登录时，该文件被执行。

编辑/etc/profile文件

**vim /etc/profile**

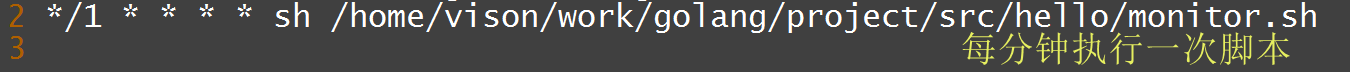
在文件最后添加运行脚本命令即可



然后source /etc/profile

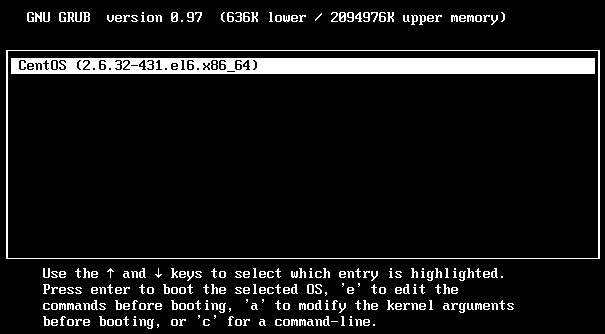
**(2) 设置监控程序是否结束**

**crontab -e**

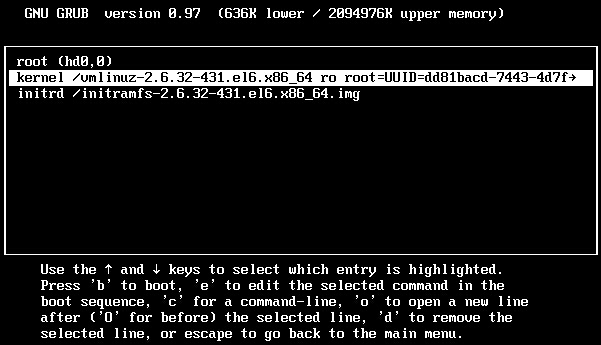


# 9 linux忘记root密码解决办法

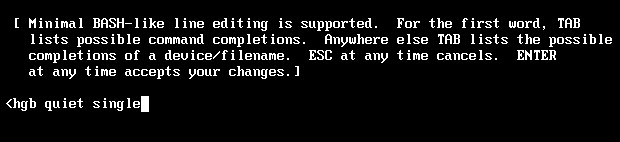
(1) 重启服务器，在读秒的时候按任意键，就会出现如下界面，在此界面中按下键盘中的‘e’，从而进入grub模式



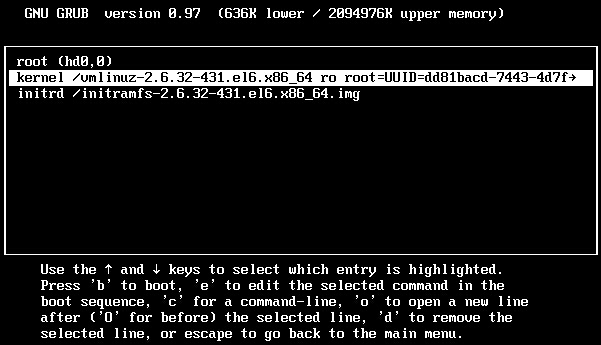
(2) 将光标移动到kernel那一行，然后再一次按‘e’，进入kernel该行的编辑界面



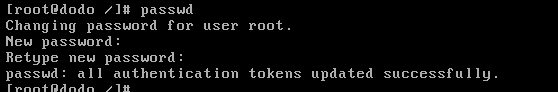
(3) 进入kernel编辑界面，在kernel编辑界面，按一下空格键，然后在后面输入single，同时按下回车键enter退出kernel编辑界面。



(4) 退出kernel界面后会回到grub模式界面，在此界面再次将光标移动到kernel那一行，然后按下‘b’来启动系统



(5) 在单用户模式下免密码进入系统，然后修改密码passwd root



# 10 多虚拟机共用一对公私钥登陆远程主机

前提是有一组能够使用ssh免密码登陆服务器的公私钥，说明公钥已经在被服务器储存

|  |  |
| --- | --- |
| 切换到普通用户运行的帐号 | su vison  cd ~ |
| 生成ssh私钥和公钥 | ssh-keygen -t rsa  注：生成的公私钥过程中一路按确认键默认即可，公私钥存放位置在生成过程中有说明，id\_rsa(私钥) id\_rsa.pub(公钥) |
| 复制公共的公私钥内容替换新生成的公私钥 | 把公共的公私钥复制到目录~  替换公钥：  cat ~/id\_rsa.pub > ~/.ssh/id\_rsa.pub  替换私钥：  cat ~/id\_rsa > ~/.ssh/id\_rsa |
| 把公钥复制为已认证的公钥keys里 | cd ~/.ssh  cp id\_rsa.pub authorized\_keys  注意authorized\_keys这个文件的权限问题，不能让所有者之外的用户对authorized\_keys文件有写权限，否则会造成设置免密码登录无效。 |

注：第一次登陆需要手动确认

例如：ssh root@192.168.8.201 -p 22

# 11修改系统时间

**手动修改时区：**

cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

修改为中国的东八区：

vim /etc/sysconfig/clock

|  |
| --- |
| ZONE="Asia/Shanghai"  UTC=false  ARC=false |

设定日期时间：

date -s "2016-12-13 11:03:30"

## 自动更新时区

安装ntpdate

apt-get install ntpdate

使用命令crontab -e

|  |
| --- |
| 10 5 \* \* \* root ntpdate us.pool.ntp.org;hwclock -w |

每天5:10自动进行网络校时，并同时更新BIOS的时间

# 12 网络相关

(1) ping命令，用来检查网络是否可达状态，但不能ping通，不代表网络没有连接，有可能对方防火墙禁止。

ping <ip>

(2) 检测服务是否可达，先要安装telnet工具

telnet <ip> <port>

## 网络命名空间

**查看网络命名空间列表**

ip netns list

**删除网络命名空间**

ip netns del <名称>

**查看网络命名空间ip地址和链接**

ip netns exec <名称> ip a

ip netns exec <名称> ip link

**添加一个网络命名空间**

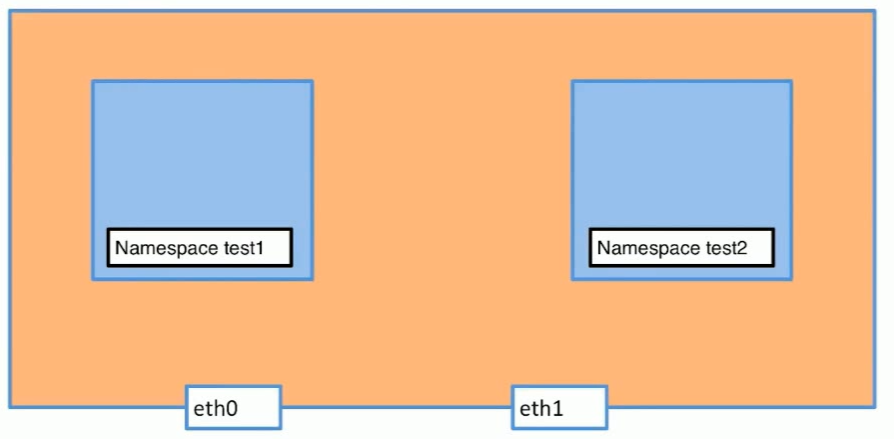
ip netns add <名称>

示例：

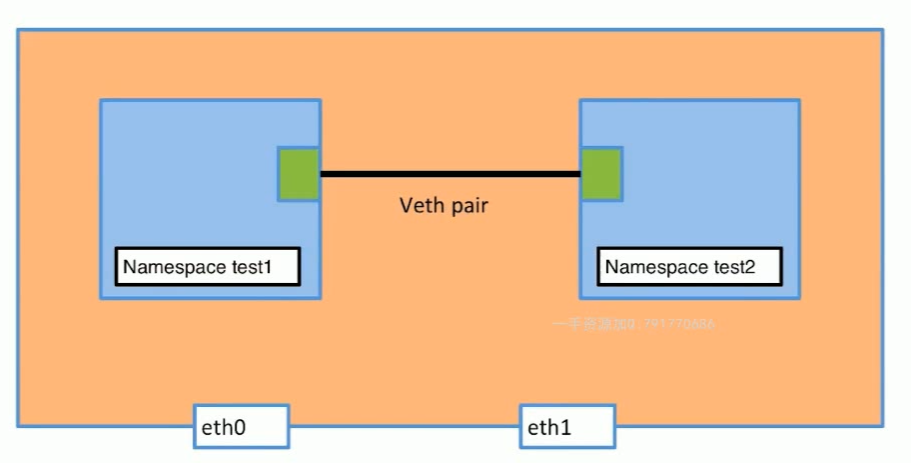
ip netns add nns01

ip netns add nns02

添加新的两个网络命名空间，暂时没有ip地址，也没有启动，是down的状态，如图所示:



要想连接，需要创建veth pair(相当于给两台主机添加ip和一条线)，如果再给两个端口各自配ip地址，就可以通信了。



**创建veth pair，类似给两个端口准备一条网线**

ip link add <veth01名称> type veth peer name <veth02名称>

例如：

ip link add veth\_nns01 type veth peer name veth\_nns02

使用ip link命令查看，发现多了两个@链接，此时都是down。

**把端口设置到命名空间**

ip link set <veth名称> netns <网络命名空间名称>

例如：

ip link set veth\_nns01 netns nns01

ip link set veth\_nns02 netns nns02

使用ip link命令发现新建的两个端口消失了，是转移到了网络命名空间里，可以使用ip netns exec <名称> ip link查看，此时两个命名空间的两个端口都是down状态，也还没有ip地址。

**设置网络命名空间端口ip地址**

ip netns exec <名称> ip addr add <ip地址> dev <端口名>

例如：

ip netns exec nns01 ip addr add 192.168.1.101/24 dev veth\_nns01

ip netns exec nns02 ip addr add 192.168.1.102/24 dev veth\_nns02

**启动或关闭网络命名空间端口**

ip netns exec <名称> ip link set dev <dev名称> <up|down>

例如：

ip netns exec nns01 ip link set dev veth\_nns01 up

ip netns exec nns02 ip link set dev veth\_nns02 up

两个网络命名空间就可以相互ping通了，但是不能ping通到其他ip，需要桥接和网络地址转换。

完整示例

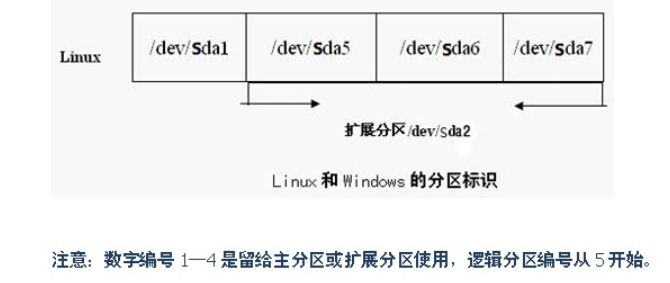
|  |
| --- |
| (1) 添加两个网络命名空间  ip netns add nns01  ip netns add nns02  (2) 创建一条连接的两个端口  ip link add veth\_nns01 type veth peer name veth\_nns02  (3) 设置网络命名空间端口  ip link set veth\_nns01 netns nns01  ip link set veth\_nns02 netns nns02  (4) 设置ip地址  ip netns exec nns01 ip addr add 192.168.1.101/24 dev veth\_nns01  ip netns exec nns02 ip addr add 192.168.1.102/24 dev veth\_nns02  (5) 启动端口  ip netns exec nns01 ip link set dev veth\_nns01 up  ip netns exec nns02 ip link set dev veth\_nns02 up  测试是否可以相互ping通  ip netns exec nns01 ping 192.168.1.102  ip netns exec nns02 ping 192.168.1.101 |

# 13 systemctl命令

systemctl命令是系统服务管理器指令

|  |  |
| --- | --- |
| 任务 | 新指令 |
| 使某服务自动启动 | systemctl [enable](http://man.linuxde.net/enable) httpd.service |
| 使某服务不自动启动 | systemctl disable httpd.service |
| 检查服务状态 | systemctl status httpd.service （服务详细信息） systemctl is-active httpd.service （仅显示是否 Active) |
| 显示所有已启动的服务 | systemctl list-units --[type](http://man.linuxde.net/type)=service |
| 启动某服务 | systemctl start httpd.service |
| 停止某服务 | systemctl stop httpd.service |
| 重启某服务 | systemctl restart httpd.service |

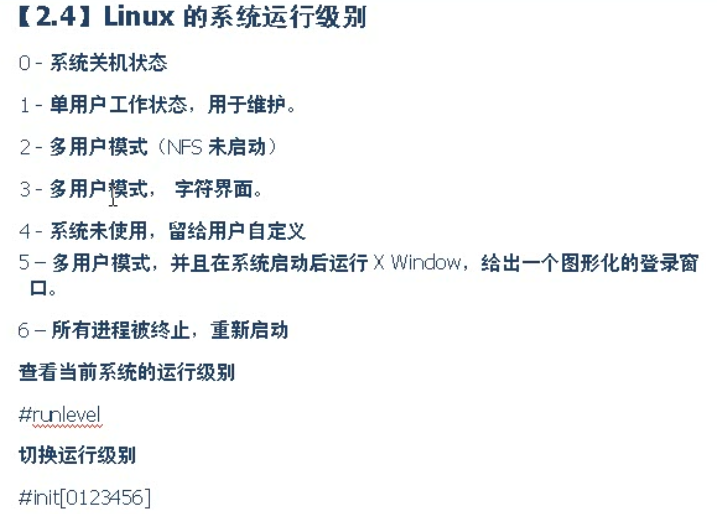
系统分区



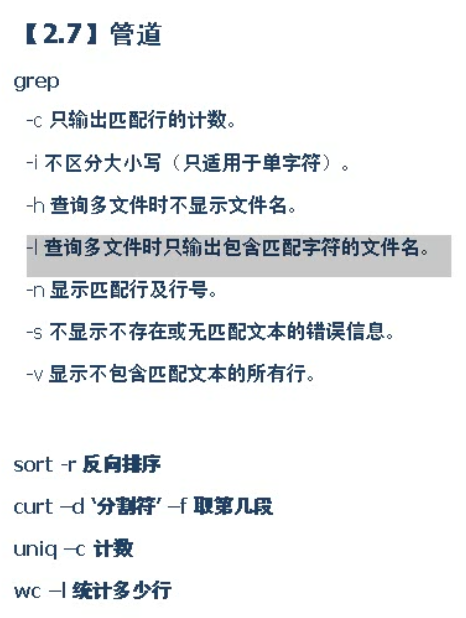


swap一般为内存的两倍

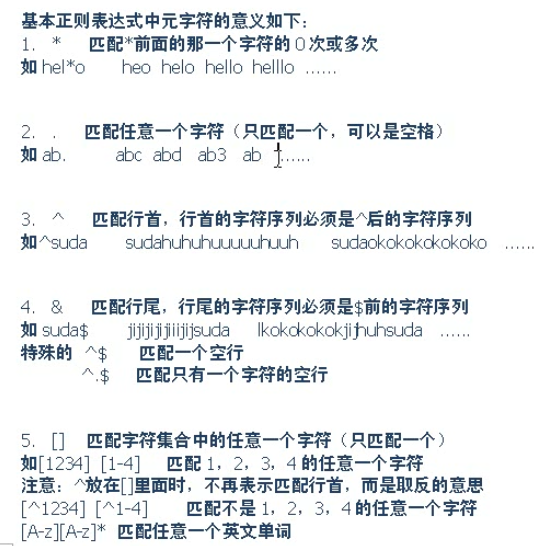
大部分安装软件都安装在/usr/local上







# 正则：



# 其他命令

检查，一个软件包是否安装：**dpkg -s 软件名**

查看已经加载的模块：**lsmod** 或 **cat /proc/modules**

查看模块详细信息：**modinfo 模块名称**

移除模块：**rmmod 模块名称**

查看可使用的设备：**cat /proc/devices**

打印目录树：**tree -a**

打印2层目录树：tree /usr/local -L 2

整个开机流程是

(1) 载入BIOS的硬件信息，并取得第一个开机装置的代号

(2) 读取第一个开机装置的MBR的boot Loader (grub)的开机信息

(3) 载入OS Kernel信息，解压Kernel，尝试驱动硬件

(4) Kernel执行init程序并获得run-lebel信息(如3或5)

(5) init执行/etc/rc.d/rc.sysinit

(6) 启动内核外挂模块(/etc/modprobe.conf)（/etc/profile）

(7) init执行run-level的各种Scripts，启动服务

(8) init执行/etc/rc.d/rc.local

(9) 执行/bin/login，等待用户Login

(10) Login后进入Shell