一、主机与主机之间通信的三个要素

```
- IP地址(IP address);
- 子网掩码(subnet mask);
- IP路由(IP router)涉及路由器;
```

IP地址的分类，只需要看IP地址第一个数字，进行判别分类

二、IPV4地址组成(点分十进制)

```
一共32个二进制位(0和1)
    11111111.11111111.11111111.11111111(二进制表示)
    255.255.255.255(上面的二进制转换成十进制)
```

表示为4个十进制数,以 . 隔开。点分十进制

三、Linux目录作用

```
/boot : 存放的是启动Linux时使用的一些核心文件，包括一些连接文件以及镜像文件;
/etc : 存放所有的系统管理所需要的配置文件和子目录;
/media : Linux系统会自动识别一些设备，例如U盘，光驱等，当识别后，Linux会把识别
的设备挂在到这个目录下;
/lib : 存放着系统最基本的动态连接共享库，其作用类似于Windows李的DDL文件。
    几乎所有的应用程序都需要用到这些共享库;
/opt : 安装第三方软件存放的目录;
/root : 系统管理员的目录，也称为超级权限这的用户主目录;
/usr : 用户的很多应用程序和文件都存放在此目录下，类似于windows下的program files目录;
/bin : bin-Binary的缩写，保存着经常使用的命令;
 /home : 用户主目录，在Linux中，每个用户都有一个自己的目录，一般该目录名是以用户的
         账号名称命名的;
```

四、用户操作

```
  [root@server0 ~]#groupadd  lisizu     //创建组
  [root@server0 ~]#useradd  -G  tarena  -u 10049  -s  /sbin/nologin   kenji
  [root@server0 ~]#gpasswd  -a  kenji  lisizu     //将用户Kenji添加到lisizu中
  [root@server0 ~]#gpasswd  -d  kenji  lisizu       //将用户kenji从lisizu组中删除
  [root@server0 ~]#usermod  -G  zhangsan,lisi,wangwu   kenji      //给用户添加多个附加
组
```

五、修改用户密码

```
[root@server0 ~]#passwd  用户名        //交互式修改用户密码
[root@server0 ~]#echo  "密码" | passwd  --stdin  用户名      //无交互式给用户设置密码
```

六、查看文本内容

```
head和tail默认显示10行
[root@server0 ~]#head -n  15   /etc/passwd    //显示前15行，可简写成-15
[root@server0 ~]#tail  -n  15   /etc/passwd     //显示后15行，可简写成-15
[root@server0 ~]#cat  -n  /etc/passwd          //显示行号
[root@server0 ~]#cat  /etc/passwd |  head  -15    //显示前15行
[root@server0 ~]#cat  /etc/passwd |  tail  -15     //显示后15行
[root@server0 ~]#cat  /etc/passwd | less  或者 less  /etc/passwd
  //分屏浏览，q键退出，上下键翻动(n,N切换)
[root@server0 ~]#more  /etc/passwd   //未显示的部分以百分比显示
```

七、查找文本中的内容

```
[root@server0 ~]#grep  ^root    /etc/passwd    //以root开头
[root@server0 ~]#grep   root$    /etc/passwd      //以root结尾
[root@server0 ~]#grep      ^$     /etc/passwd       //查找空行
[root@server0 ~]#grep   -v  ^$   /etc/passwd        //查找不包含空行的行
```

八、别名的定义

```
(alias 名称='命令')------定义别名后，生效需要重开一个终端
[root@server0 ~]#vim /root/.bashrc        //对于单个用户生效
[root@server0 ~]#vim /etc/bashrc          //对所有用户都生效
```

九、挂载

```
[root@server0 ~]#mount  /dev/cdrom   /root/test      //临时挂载
[root@server0 ~]#umount  /root/test        //取消挂载
[root@server0 ~]#vim  /etc/fstab          //配置文件永久挂载
/var/lib/libvirt/images/iso/CentOS7-1804.iso   /var/ftp/centos-1804   iso9660
defaults   0   0
[root@server0 ~]#mount  -a         //检测挂载是否成功
```

十、拷贝命令

```
[root@server0 ~]#cp   /etc/passwd    /root    //拷贝单个文件
[root@server0 ~]#cp   /etc/passwd   /root/haha    /opt/      //拷贝多个文件
[root@server0 ~]#\cp  /etc/passwd  /opt/      //强制拷贝，不再提示，当前命令生效
[root@server0 ~]#cp   -r  /home    /root/test       //递归持续复制，用于目录，文件也可以，排除掉链接文件
[root@server0 ~]#cp  -p  /home   /opt       //连同档案的属性(属主，属组)一起复制过去，而非使用预设属性
[root@server0 ~]#cp -r  -v  /etc/passwd   /opt/      // -v 显示拷贝的过程
```

十一、find查找文件数据命令

常用条件表示：

```
    -type 类型----f(文件)、d(目录)、l(快捷方式)
    -name "文档名称"
    -size +或-文件大小(k[小写]、M、G)-----》严格区分大小写
    -user 用户名
    -mtime +n或-n    +n代表多少天以前的文件，-n表示文件更改时间距离现在多少天以内
```

例子：

```
[root@server0 ~]#find  /boot  -type  l    //查找/boot下的所有快捷方式
[root@server0 ~]#find / -user student -type f      //找到根下所有student用户的文件
[root@server0 ~]#find  /etc/  -name  "*conf"  -type f  -exec  cp  {}  /opt/test  \;
[root@server0 ~]#find  /etc/  -name  "*conf"  -type f  -exec  ls  -l {} \;
[root@server0 ~]#find  /  -size  +500M  -exec  rm -rf  {} \;     //删除找到的所有大于500M的文件
[root@server0 ~]#find  /boot  -size  +10M  -type  d    //找到/boot下所有大于10M的目录
[root@server0 ~]#find /boot  -mtime  +10      //boot目录下查找十天前创建或修改的数据
[root@server0 ~]#find   /    -mtime   -10      //在根下查找10天以内创建或修改的文件
```

十二、linux权限详述

linux一切皆文件，多个用户登录操作系统，系统有默认文件，root文件，每个用户都有自己的文件，如何对文件进行权限分配和管理？

每个文件的权限属性分为：属主(owner)，属组(group),其他人(other)

linux系统中，文件创建后的三种访问方式：

```
[root@server0 ~]# ls -ld /opt/kingsoft/
drwxrwxrwx. 3 root root 4096 1月  24 2019 /opt/kingsoft/
```

属主：用户若是文件的主人，则匹配属主的权限，权限在文件的左三位，即第一个rwx

属组：用户与属组在同一个组，则匹配属组的权限，权限在文件的中三位，即第二个rwx

其他人：用户既不是属主也不是属组，则匹配其他的权限，权限在文件的右三位，即第三个rwx

要判断一个用户对文件的权限，按照 属主->属组->其他人 顺序进行匹配，先匹配属主，如果不匹配则继续匹配属组，如果匹配则后面的属组和其他人不再匹配。

1、对r、w、x进行说明

```
r、w、x对于文件而言：
r    可以使用文件查看类工具获取文件中的内容；
w    可以修改文件中的内容；
x    可以把二进制可执行文件向内核申请，将文件启动为一个进程(简单说就是类似让文件成为一个可以执行的程序)

r、w、x对于目录而言：
r    可以使用ls命令查看此目录中的文件列表；
w    可以在此目录中创建、删除文件；
x    可以cd进入此目录，可以使用ls -l查看目录中的文件列表.

判断用户对文件或目录的权限：
    前提条件：用户能够成功进入到文件所在的目录，即对操作的文件所在目录有x权限
```

2、文件特殊权限有三种：set uid、set gid、sticky bit

```
set uid：（存在安全隐患，慎用）
    打破了进程安全上下文法则，针对二进制可执行文件，使执行者在执行文件时可以临时获得可执行文件owner的权限，
    以文件owner的身份去执行文件
    前提条件：用户具备此二进制可执行文件的x权限
    [root@server0 ~]# ls -al executable_file
        -rwxr--r--. 1 root root 0 Jul 11 20:44 executable_file
    [root@server0 ~]# chmod u+s executable_file       //通过chmod命令赋予二进制可执行文件suid权限
    [root@server0 ~]# ls -al executable_file
        -rwsr--r--. 1 root root 0 Jul 11 20:44 executale_file
```

```
set gid：针对目录，使任何用户在此目录下创建文件的group与该目录的group相同
前提条件：用户具备此目录的x权限和w权限
[root@server0 ~]# ls -al dir/
    drwxr-xr-x.  2 root root 4096 Sep 13 19:32 .
[root@server0 ~]#chmod g+s dir/            //通过chmod命令赋予目录sgid权限
[root@server0 ~]# ls -al dir/
    drwxr-sr-x.  2 root root 4096 Sep 13 19:32
```

```
sticky bit：防删除位，用户能添加文件，也可以删除自己的文件，但不能删除该目录下其他用户的文件
前提条件：用户具备此目录的x权限和w权限
[root@server0 ~]#chmod o+t dir/           //通过chmod命令赋予目录sbit权限
[root@server0 ~]# ls -al dir/
    drwxr-xr-t.  2 root root 4096 Sep 13 19:32
```

3、文件默认权限：umask

```
全称：user file-creation mode mask（用户文件创建掩码）是用户在创建文件或目录时默认权限的基础；
    在没有umask时，文件的默认权限是0666（rw-rw-rw-），目录的默认权限是0777（rwxrwxrwx）
    [root@server0 ~]# umask     //umask默认值为0022
        0022
    默认权限（目录0777，文件0666）- umask权限（0022） = 实际权限(目录0755，文件0644)
    目录的权限为   “rwxrwxrwx”-“----w--w-”=“rwxr-xr-x”
    文件的权限为    “rw-rw-rw-”-“----w--w-”=“rw-r--r--”
    默认情况下，创建目录的权限值为755 即rwxr-xr-x ，创建普通文件的权限值为644 即rw-r--r--
```

```
如何设置umask：
当前登录窗口临时修改方式：（退出重新登录umask又变回0022）
[root@server0 ~]# umask              //查看umask值
    0022
[root@server0 ~]# mkdir test1
[root@server0 ~]# ls -ld test1          //查看目录权限
    drwxr-xr-x. 2 root root 6 9月  26 15:25 test1            //数字权限表示：0755

[root@server0 ~]# umask 0002      //设置更改umask的值
[root@server0 ~]# umask              //查看umask值
    0002
[root@server0 ~]# mkdir  test2
[root@server0 ~]# ls -ld test2
    drwxrwxr-x. 2 root root 6 9月  26 15:26 test2         //数字权限表示：0775
```

4、更改文件属主和属组： chown

```
[root@server0 ~]#groupadd  teacher
    [root@server0 ~]#mkdir  testfile
    [root@server0 ~]#chown  teacher     testfile       //修改文件的属主
    [root@server0 ~]#chown  :teacher    testfile       //修改文件的属组(属组名前加：)
    [root@server0 ~]#chown  student:teacher   testfile     //同时修改文件的属主和属组

重要参数：-R ： 递归修改目录及目录下所有文件及其子目录的所有属主和属组
```

5、更改文件的权限：chmod

```
<一>字符修改方式（推荐，不容易错）：
    [root@server0 ~]#chmod u=rwx,g=r-x,o=r-- dir/      //将目录的权限设置为rwxr-xr--
    [root@server0 ~]#chmod u+w dir/              //给owner增加w权限
    [root@server0 ~]#chmod u-w dir/       //给owner减去w权限

<二>数字修改方式
    4 为 SUID = u+s
    2 为 SGID = g+s
    1 为 SBIT = o+t
```

```
[root@server0 ~]#cp   /etc/passwd    /root/
[root@server0 ~]#chmod 664  /root/passwd      //将passwd权限设置为rw-rw-r--
[root@server0 ~]# ls -l passwd
    -rw-rw-r--. 1 root root 2005 9月  26 15:56 passwd

[root@server0 ~]#chmod  2571    /root/passwd    //将passwd权限设置为-r-xrws--x
[root@server0 ~]# ls -l passwd
    -r-xrws--x. 1 root root 2005 9月  26 15:56 passwd
[root@server0 ~]#chmod  1264    /root/passwd          //将passwd权限设置为w-rw-r-T
[root@server0 ~]# ls -l passwd
    --w-rw-r-T. 1 root root 2005 9月  26 15:56 passwd
```

十三、用tar工具制作，查看，释放压缩包

```
（这三个个参数是独立的命令，压缩解压都要用到其中一个，可以和别的命令连用，但一次只能用其中一个）
    -c：建立一个压缩文件的参数指令(create 的意思)；
    -x：解开一个压缩文件的参数指令；
    -t：查看 tarfile 里面的文件；
    特别注意，在参数的下达中， c/x/t 仅能存在一个！不可同时存在！因为不可能同时压缩与解压缩。

下面的参数是根据需要在压缩或解压档案时可选的：
    -z：有gzip属性,即需要用 gzip 压缩；
    -j：有bz2属性,即需要用 bzip2 压缩；
    -J：  有xz属性，即需要用xz压缩；
    -v ：压缩的过程中显示文件(显示所有过程)；
    -f ：   使用档案名字，切记，这个参数是最后一个参数，后面只能是要压缩的档案名；
    -p(小写) ：打包时保留原始文件及目录的权限，常常在备份文件时候使用(针对root用户，以普通用户
                的身份对文件进行tar打包，root用户来解压缩包，文件的权限(属主和属组)还是原来文件
的；
                而用普通用户来解压缩包，文件的权限为当前普通用户的权限）

[root@server0 ~]#useradd  zhangsan
[root@server0 ~]#useradd  lisi
[root@server0 ~]#mkdir  /home/share      //创建共享文件夹
[root@server0 ~]#chmod  ugo=rwx  /home/share        //共享文件夹授权，普通用户可以把文件放
进去
```

```
[root@server0 ~]#su  -  zhangsan
[zhangsan@server0 ~]$touch  test       //属主和属组均为zhangsan
[zhangsan@server0 ~]$tar  -zcvpf  test.tar.gz   test      //-p保留权限
[zhangsan@server0 ~]$cp  test.tar.gz    /home/share/test.tar.gz
[zhangsan@server0 ~]$exit     //退出zhangsan登录，返回root界面
[root@server0 ~]#cp  /home/share/test.tar.gz    /root/
[root@server0 ~]#tar   -xvf   test.tar.gz
[root@server0 ~]#ls   -l  test           //文件的属主和属组还是zhangsan用户的
```

```
[root@server0 ~]#su   -  lisi      //用普通用户测试解包文件的权限
[lisi@server0 ~]$cp    /home/share/test.tar.gz    .
[lisi@server0 ~]$tar   -xvf   /test.tar.gz
[lisi@server0 ~]$ls   -l   test       //发现文件的属组和属组为当前lisi用户
```

结论：-p(小写)保留文件原本权限只是针对于root用户，普通用户解包后权限自动更改为当前普通用户的权限

-P(大写) ：可以使用绝对路径来压缩；
注意一点，用-P(大写)压缩的压缩包解压时也需要加上-P(大写)，即保留绝对路径，
如果不加-P也是从当前工作目录解压

压缩比率：bzip2 > gzip > zip
压缩比率=原内容大小/压缩后大小，压缩比率越大，则表明压缩后占用空间的压缩包越小
zip的通用性较好，而现在windows下软件winrar,7zip等对tar.gz的支持也非常好。
推荐用tar.gz，bzip2要耗费更多的cpu

进入相应的目录下进行打包（解包时不会出现目录结构）

```
[root@server0 ~]#tar -tf all.tar.gz         // 这条命令是列出all.tar.gz包中所有文件，-t是列出文件的意思
[root@server0 ~]#tar -xf all.tar.gz   -C   /opt/        // 这条命令是解出all.tar.gz包中所有文件，
                                                    -x是解开的意思，-C指定解压路径

[root@server0 ~]#tar  -zcvf  all.tar.gz    /home   /opt    //gzip格式打包，并显示详细信息
[root@server0 ~]#tar  -zcvpf   all.tar.gz     /home    //解压缩后，文件的属性(属主和属组)不会因为使用者而变。

[root@server0 ~]#tar  -zcvPf  all.tar.gz  /home   /opt    //使用绝对路径压缩
[root@server0 ~]#tar  -xPf   all.tar.gz    //解压缩时，将压缩的文件释放到相应的路径
```

使用特定的打包方式(解包时不会出现目录结构)-----》目录和文件分开，在每个打包的文件的上层目录前加-C

```
[root@server0 ~]#tar  -zcvf   test.tar.gz   -C   /etc/   passwd  -C  /opt/   shadow
```

十四、firewall防火墙

防火墙守护 firewalld 服务引入了一个信任级别的概念来管理与之相关联的连接与接口。它支持 ipv4 与 ipv6，并
支持网桥，采用 firewall-cmd来动态的管理 kernel netfilter临时或永久的接口规则，并实时生效而无需重启服务。
安装：

```
[root@server0 ~]#yum  -y   install   firewalld       //有则不用安装
[root@server0 ~]#systemctl start firewalld          //启动
[root@server0 ~]# systemctl enable firewalld            //开机启动
```

```
[root@server0 ~]#systemctl stop firewalld             //关闭
[root@server0 ~]#systemctl disable firewalld        //取消开机启动

Firewall 能将不同的网络连接归类到不同的信任级别，Zone 提供了以下几个级别：
    public：允许指定的进入连接,ssh,ping命令；
    trusted：信任所有连接；
    block：拒绝所有外部发起的连接，允许内部发起的连接；
    drop：丢弃所有进入的包，而不给出任何响应；

具体的规则管理，可以使用 firewall-cmd：
firewall-cmd   --help          //列出防火墙的帮助信息
查看规则：

[root@server0 ~]#firewall-cmd --state           //查看运行状态
[root@server0 ~]#firewall-cmd --get-active-zones       //查看已被激活的 Zone 信息
[root@server0 ~]# firewall-cmd  --list-all  --zone=block     //检查运行时区域block下的规
则
[root@server0 ~]#firewall-cmd --get-default-zone           //查看防火墙的默认级别
[root@server0 ~]#firewall-cmd --set-default-zone=public           //将防火墙设置为
public，本身永久生效
[root@server0 ~]#firewall-cmd --reload              //让设置的防火墙配置生效
[root@server0 ~]#firewall-cmd  --permanent  --zone
                  =trusted --add-forward-port=port=22:proto=tcp:toport=3753
    //然后转发 tcp 22 端口至 3753，--permanent永久生效
[root@server0 ~]#firewall-cmd  --permanent  --zone=block  --add-source=172.25.0.0/24

    //添加永久配置“阻塞来自网段172.34.0.0/24的任何访问”
[root@server0 ~]# firewall-cmd  --list-all
[root@server0 ~]#firewall-cmd --list-all-zones          //列出所有区域的规则，可以查看自
己设置的防火墙
```

十五、nmcli管理命令的使用

1、使用nmcli添加一个新的网卡

```
[root@server0 ~]#nmcli  connection  add  type ethernet  ifname  eth4  con-name  eth4
//添加以后会生成配置文件    /etc/sysconfig/network-script/ifcfg-eth4
```

2、使用nmcli配置IP地址

```
[root@server0 ~]#nmcli  connection  show
 //看看管理列表中都有哪些网卡，通过ifconfig可以查看当前服务器中都有哪些网卡，
这里有的网卡才可以通过才可以通过nmcli添加到管理列表中，配置IP后才能生效，否则会报错，找不到设备，
虚拟机的话可以先添加一个物理网卡，再将其添加到nmcli管理列表中
```

```
  [root@server0 ~]#nmcli  connection  add  type  ethernet  ifname  eth4   con-name
eth4
 //向管理列表中添加一个网卡

centos7.2版本以前:
[root@server0 ~]#nmcli connection modify 'System eth0' ipv4.method manual
                    ipv4.addresses "172.25.0.12/24 172.25.0.12"
                    connection.autoconnect yes
[root@server0 ~]#nmcli  connection  up  "System  eth0"            //激活网卡

centos7.2版本以后:
[root@server0 ~]#nmcli connection modify  "System  eth0" ipv4.method manual
  ipv4.addresses 172.25.0.12/24  ipv4.gateway 172.25.0.254  ipv4.dns 172.25.254.254
   connection.autoconnect yes
[root@server0 ~]#nmcli  connection  up  "System  eth0"        //激活网卡
```

3、使用nmcli配置聚合链路

```
1》创建虚拟网卡组:
[root@server0 ~]# nmcli connection add type team con-name team0 ifname team0
autoconnect yes
                         config '{"runner":{"name":"activebackup"}}'
命令格式简单说明:
  nmcli connection  添加    类型为    team (组队)    配置文件名   team0
 网卡名  team0    每次开机自动启用    工作模式为      热备份
    2》给组里添加成员:
  [root@server0 ~]#nmcli connection add type  team-slave  con-name team0-1 ifname
eth1  master team0
  [root@server0 ~]#nmcli connection add type  team-slave  con-name team0-1 ifname
eth1  master team0
  命令简介:
  nmcli connection 添加  类型   team-成员   配置文件名为  team0-1  网卡名  eth1 主设备  为
team0
3》为虚拟网卡team0配置IP地址:
   [root@server0 ~]#nmcli connection modify team0 ipv4.method manual
 ipv4.addresses 192.168.1.1/24 connection.autoconnect yes
4》激活配置:
[root@server0 ~]#nmcli  connection  up  team0
[root@server0 ~]#nmcli  connection   up  team0-1
[root@server0 ~]#nmcli  connection   up  team0-2
5》查看链路聚合状态: (测试效果)
[root@server0 ~]#ifconfig  eth0  down      //如果网卡1 down掉
[root@server0 ~]#teamdctl  team0  state     //查看当前team0的状态
6》删除team0的配置:
[root@server0 ~]#nmcli  connection  delete  team0
7》查看网卡的配置文件的位置:
```

```
[root@server0 ~]# ls /etc/sysconfig/network-scripts/
```

十六、windows里可以直接将文件拖拽到xshell的黑窗口中就可以上传到linux系统的软件

```
yum -y install lrzsz       //安装到xshell需要远程的linux服务器上
```

十七、分区方式

什么是分区?
　　分区是将一个硬盘驱动器分成若干个逻辑驱动器，分区是把硬盘连续的区块当做一个独立的磁硬使用。
分区表是一个硬盘分区的索引,分区的信息都会写进分区表。
为什么要有多个分区?
防止数据丢失：如果系统只有一个分区，那么这个分区损坏，用户将会丢失所的有数据。
增加磁盘空间使用效率：可以用不同的区块大小来格式化分区，如果有很多1K的文件，
而硬盘分区区块大小为4K，那么每存储一个文件将会浪费3K空间。这时我们需要取这些文件大小的
平均值进行区块大小的划分。
数据激增到极限不会引起系统挂起：将用户数据和系统数据分开，可以避免用户数据填满整个硬盘，
引起的系统挂起。

<一>分区工具fdisk用法介绍：（fdisk工具他对分区是有大小限制的，它只能划分小于2T的磁盘）

```
fdisk命令参数介绍
p    打印分区表。
n    新建一个新分区。
d    删除一个分区。
q    退出不保存。
w    把分区写进分区表，保存并退出。
[root@server0 ~]#lsblk              //查看系统磁盘分区情况
[root@server0 ~]#fdisk   /dev/vdb
 //可以分出三个主分区，第四个分区不要指定大小，后面可继续创建逻辑分区
.................................
 [root@server0 ~]#mkfs.xfs   /dev/vdb1         //格式化文件系统
 [root@server0 ~]#blkid      //查看磁盘的文件系统
 [root@server0 ~]#mkdir    /mnt/test          //创建挂载点
 [root@server0 ~]#mount   /dev/vdb1  /mnt/test        //临时挂载，重启失效
 [root@server0 ~]#umount  /mnt/test         //取消挂载
 [root@server0 ~]#vim   /etc/fstab
     /dev/vdb1    /mnt/test    xfs    defaults      0      0
 [root@server0 ~]#mount   -a          //检测挂载是否成功
 [root@server0 ~]#df   -h         //查看是否挂载成功
```

<二>逻辑卷LVM:

```
LVM的最重要的优点在与于弹性调整文件系统的容量，lvm可以将多个物理分区在一起，
像组成了一块完整的可伸缩的硬盘。
```

第一步：

将物理硬盘格式化成PV(物理卷)-----》使用 pvcreate 命令(如果是fdisk刚划分出来的分区不需要执行，

已经使用过的，有了文件系统的磁盘需要执行此操作)

```
[root@server0 ~]#pvcreate  /dev/vdb   /dev/vdc
[root@server0 ~]#pvs        //查看当前pv的信息
```

第二步：

创建卷组(VG)，并将PV加入到卷组中：

```
[root@server0 ~]#vgcreate   -s  1M  basename  /dev/vdb   /dev/vdc     //-s指定PE的大小
[root@server0 ~]#vgs      //查看卷组的详细信息
[root@server0 ~]#lvcreate  -L   2G   -n  vo  basename       //创建逻辑卷,也可以是“-l  40(PE的个数)”
[root@server0 ~]#lvs         //查看创建好的逻辑卷的信息
```

我们创建LV(逻辑卷)的大小是根据当前VG(卷组)的大小来决定的，不能超过当前VG(卷组)的剩余大小。

每创建好一个逻辑卷，都会在 /dev 目录下出现一个以该卷组命名的文件夹，基于该卷组创建的所有的逻辑卷都是存放在这个文件夹下面，我们可以查看一下：

```
[root@server0 ~]#ls   /dev/basename/vo
```

第三步：

格式化逻辑卷：

```
[root@server0 ~]#mkfs.ext4   /dev/basename/vo     //将逻辑卷vo文件系统格式化为ext4(ext3, xfs)
[root@server0 ~]#blkid          //获取要挂载的磁盘名
[root@server0 ~]#mkdir   /mnt/base     //创建挂载点
[root@server0 ~]#vim /etc/fstab         //永久挂载
/dev/mapper/basename-vo   /mnt/base    ext4    defaults      0      0
[root@server0 ~]#mount  -a
[root@server0 ~]#df    -h         //检测挂载是否成功
```

第四步：

扩展LVM逻辑卷：(分为两种情况)

第一种情况，需要扩展的pe数量或磁盘大小，vg(卷组)可以提供

```
[root@server0 ~]#lvs    //查看逻辑卷信息
[root@server0 ~]#lvextend   -L  +100M   /dev/basename/vo    //在原基础上再扩展100M
[root@server0 ~]#lvextend  -L   10G    /dev/basename/vo      //卷的大小扩展到10G
[root@server0 ~]#df   -h       //磁盘容量没改变，需要对文件系统进行扩展
[root@server0 ~]#resize2fs    /dev/basename/vo      //扩展ext3或ext4的文件系统
[root@server0 ~]#xfs_growfs   /dev/basename/vo    //扩展xfs的文件系统
```

第二种情况，vg(卷组)的大小不足，需要先扩展卷组

```
[root@server0 ~]#pvcreate   /dev/sdg      //创建物理卷，新建磁盘不需要
[root@server0 ~]#vsextend  basename   /dev/sdg
```

剩下的步骤跟第一种情况相同，参照上例子

第五步：
删除逻辑卷：
【注意】对于创建物理卷、创建卷组以及创建逻辑卷我们是有严格顺序的，同样，
对于删除逻辑卷、删除卷组以及删除物理卷也是有严格顺序要求的：
①首先将正在使用的逻辑卷卸载掉　通过 umount 命令
[root@server0 ~]#umount /mnt/base //永久挂载需要注释相应的配置文件
②将逻辑卷先删除　通过 lvremove 命令
[root@server0 ~]#lvremove /dev/basename/vo
③删除卷组　通过 vgremove 命令
[root@server0 ~]#vgremove basename
④最后再来删除我们的物理卷　通过 pvremove 命令
[root@server0 ~]#pvremove /dev/vdb /dev/vdc

<三>parted分区方式：(实现对超过2T磁盘的进行分区操作)
GPT分区方式没有四个主分区的限制，最多可达到128个主分区

通过parted工具来实现磁盘分区：
[root@system ~]# parted /dev/vdb //选择要分区的硬盘
(parted) help //获取帮助信息，忘记的命令都可以在这里找
(parted) mkpart //开始进行分区命令
分区名称? []? data2
文件系统类型? [ext2]? ext4 //不起实际作用，只是一个声明
起始点? 10G //从磁盘的多少大小开始划分
结束点? 20G //划分到多大，实际大小=结束点 - 起始点
(parted) p //查看硬盘分区状态
(parted)rm 3 //删除分区3
(parted) q //退出

十八、DNS的服务器解析过程

访问网站的过程：
我们访问网站时候，首先访问的是dns服务器，询问dns服务器是否知道这个网站的确切地址。dns服务器如果有该网站的详细地址会发送给客户端，客户端才能正常访问网站，不然是打不开网站的，好的dns服务器在解析响应时间上就比较有优势，也影响这我们访问一个网站的打开速度，甚至决定了我们能不能打开这个网站。

DNS服务器解析过程：
1.电脑访问qq.com这个网站，电脑首先访问自己的hosts文件,如果有对应的ip就直接进行访问。
2.如果hosts没有对应的地址，就访问DNS解析器缓存，如果有qq.com对应的ip是
119.147.15.13这个地址就进行访问。
3.如果dns缓存还没有，就访问自己在ip地址中指定的dns服务器询问是否有对应ip地址。
(当如果得到结果并且访问成功，电脑会把它存入自己的hosts文件以备日后访问)

1、配置DNS的主从同步

```
主服务器:
yum  -y install  bind-chroot   bind
vim  /etc/named.conf
options {
        directory       "/var/named";
    allow-transfer { 192.168.4.207; };      //授权从服务器IP，通过man  named.conf的
/allow获取
};

地址库:
    2019092001  ;serial      //数据版本号，由10位数字组成，年    月    日    次数（主要）
                                                从服务器通过数据版本号来进行同步，时间越大版本
越新
        1D                  ;refresh      //主从同步数据的时间
        1H                  ;retry          //失联之后，主从同步数据的时间
        1W                 ;epire         //彻底失联的时间
        3H                  ;minimum           //无效记录的时间

从服务器:
yum   -y install  bind-chroot   bind
vim  /etc/named.conf
zone  "tedu.cn" {             //负责解析的域名
    type slave;               //设置从服务器
    file  "/var/named/slaves/tedu.cn.slave";      //同步过来的数据存放的路径和名称
     masters { 192.168.4.7; };          //指定主DNS服务器的位置
};
systemctl restart  named      //重启服务后，自动在/var/named/slaves/产生一个文件
tedu.cn.slave
                                          //以乱码的形式显示

/var/named/slaves/      //默认的存放数据同步的目录
```

十九、图形模式和命令行模式的切换

```
[root@server0 ~]#systemctl get-default              //查看每次开机默认进入模式
[root@server0 ~]# systemctl isolate graphical.target         //切换到图像模式
[root@server0 ~]#systemctl isolate multi-user.target            //切换到字符模式

设置永久策略，每次开机自动进入字符模式
[root@server0 ~]#systemctl set-default multi-user.target
[root@server0 ~]#reboot
```

#进程常用命令：# 我们通常会列出所有进程，然后通过grep命令来进行过滤，获取实际需要的信息

```
ps命令：用于查看当前正在运行的进程
a          显示终端上所有用户的进程
x          显示无终端进程
u          显示详细信息

[root@server0 ~]#ps             //显示进程信息，参数可省略
[root@server0 ~]#ps  -aux       //显示进程
[root@server0 ~]#ps  -elf       //显示进程，与aux的显示风格不同
[root@server0 ~]#ps  -aux | grep  JAVA        //aux会列出所有进程，通过管道查找我们需要的

pstree命令：查看进程与进程之间的树型关系结构
            （使用pstree命令，可以提供用户名或PID值作为参数）
[root@server0 ~]#pstree  -p  1584       //列出PID为1584的进程的进程树结构

[root@server0 ~]#jobs  -l        //列出当前用户当前终端的后台任务

[root@server0 ~]#bg  进程编号(num)              //激活后台编号为num的进程
[root@server0 ~]#fg   进程编号(num)               //将后台编号为num的进程恢复到前台

[root@server0 ~]#sleep  600          //睡眠，等待600秒
[root@server0 ~]#ps  -aux  | grep  sleep       //查找sleep的进程信息，PID号1724
    root      1724  0.0  0.0 107892   360  pts/0    S+   15:54   0:00   sleep   600

kill杀死进程命令：
[root@server0 ~]#kill   1724          //杀死进程，一些顽固，杀不掉的加 -9
[root@server0 ~]#kill   -9   1724      //强制杀死指定PID的进程
[root@server0 ~]#killall   -9   vim     //强制杀死所有名为vim的进程
```

二十、scp命令(scp 可以在 2个 linux 主机间复制文件或目录)

```
从 本地 复制到 远程：
命令格式：
            scp 本地文件   远程主机用户名@远程主机IP：远程主机路径
        scp   -r  本地目录   远程主机用户名@远程主机IP：远程主机路径

[root@server0 ~]#scp   /etc/passwd   student@172.25.0.11:/opt/
    //将本机的/etc/passwd远程复制到远程主机student用户的opt目录下
[root@server0 ~]#scp  -r   /home   student@172.25.0.11:/opt/
    //将本机的/home目录远程复制到远程主机student用户的opt目录下

从 远程 复制到 本地：
    scp 本地文件   远程用户名@远程IP：远程主机路径
    scp 本地文件   远程用户名@远程IP：远程主机路径
[root@server0 ~]#scp  student@172.25.0.11:/etc/passwd     /opt/
    //将远程主机student用户下的/etc/passwd复制到本机的opt目录下
```

```
[root@server0 ~]#scp  student@172.25.0.11:/home     /opt/
    //将远程主机student用户下的/home目录复制到本机的opt目录下

-v     用来显示进度,可以用来查看连接,认证，或是配置错误;
-P(大写)     选择端口;

[root@server0 ~]#scp -P 7920  /etc/passwd  student@176.121.212.140:/home/student/
    //如果远程主机的SSH端口发生改变，需要指定端口
```

二十一、SSH常用命令

SSH是一种网络协议，用于计算机之间的加密登录，SSH之所以能够保证安全，原因在于
它采用了公钥加密。
整个过程是这样的:

（1）远程主机收到用户的登录请求，把自己的公钥发给用户。
（2）用户使用这个公钥，将登录密码加密后，发送回来。
（3）远程主机用自己的私钥，解密登录密码，如果密码正确，就同意用户登录。

```
[root@server0 ~]#ssh pika@192.168.0.111      //以用户名pika，登录远程主机host
[root@server0 ~]#ssh  -X  pika@192.168.0.111
     //-X 作业是远程以后可以使用远程主机的图形化，例如: firefox
[root@server0 ~]#ssh -p 2222 pika@192.168.0.111
    //SSH的默认端口是22，你要远程的主机端口发生改变(2222),这时要登录远程主机，
       需要使用p参数，指定这个端口
```

SSH远程操作:

SSH不仅可以用于远程主机登录，还可以直接在远程主机上执行操作

```
[root@server0 ~]#ssh  pika@192.168.0.111  ‘mkdir   /opt/test01’     //远程创建文件夹
[root@server0 ~]#ssh  pika@192.168.0.111  'mkdir -p .ssh && cat
                     >> .ssh/authorized_keys' < ~/.ssh/id_rsa.pub
    //单引号中间的部分，表示在远程主机上执行的操作；后面的输入重定向，  表示数据通过SSH传向远程主
机。
[root@server0 ~]# ssh  pika@192.168.0.111   'ps -aux | grep httpd'
    //查看远程主机是否运行进程httpd
```

二十二、Rsync实时同步

rsync (remote sync)是一个远程数据同步工具，使用与unix/Linux/windows等多种平台,使本地和
远程两个主机之间的文件达到同步,由于只传送两个文件的不同部分，而不是每次都整份传送，
因此速度相当快。

Rsync优点:
1>支持增量备份;
2>选择性的保持 : 符号链接，硬链接，文件属性，权限 及时间等;
3>传输前执行压缩。适用于异地备份 ，镜像服务器等应用;

4>使用ssh做为传输端口。

Rsync和scp的区别：
　　当文件数据很大时候：scp无法备份大量数据；
　　scp每次拷贝都是完整拷贝。无法增量拷贝。rsync 边复制，边比较，边统计。

Rsync同步备份的原理：
　　在远程同步任务中，负责发起rsync同步操作的客户机称为发起端，而负责响应来自客机的rsync步操作的服务器称为备份源。
　　首先服务器B（发起源）向服务器A（同步源）进行数据备份，将自己的数据备份到服务器A中。当服务器B中的数据遭到损失或者增量的时候，都会从服务器A中进行数据同步。服务器B数据丢失则从服务器A中同步数据丢失的部分。当服务器B数据增多了，就会再次向服务器A进行数据备份，但是备份的不是完整备份，而是增量备份，即备份同步源中没有的数据

```
[root@server0 ~]#which  rsync       //查看这条命令是否安装
[root@server0 ~]#yum   provides   rsync        //查看这条命令是由那个包提供的
    rsync-3.0.6-5.el6_0.1.x86_64
```

二十四、rsync命令的基本用法

```
    -a 参数，相当于-rlptgoD (-r 是递归 -l 是链接文件，意思是拷贝链接文件；-p 表示
    保持文件原有权限；-t 保持文件原有时间；-g 保持文件原有用户组；-o 保持文件原
    有属主；-D 相当于块设备文件)；
    -z  --compress 表示压缩传输；
    -P(大写)   --显示传输进度；
    --delete    --删除那些目标位置有而原始位置没有的文件；
    --exclude       --忽略文件或目录；
    -v     --显示同步过程的详细信息

格式：
    rsync  -avzP  --delete  root@{远程主机}:{远程文件夹}   {本地文件夹}
      //把远程主机上的数据同步到本地文件夹

[root@server0 ~]#   rsync -avzP --delete root@192.168.1.100:/tmp/rtest1   /tmp/
    //将远程主机(100)的/tmp下的rtest1整个文件夹同步到本地的/tmp下
[root@server0 ~]#   rsync -avzP --delete root@192.168.1.100:/tmp/rtest1/   /tmp/
    //区别是只同步目录下的数据，不会把rtest1也同步过去

向远程主机推送数据：
格式：
    rsync -avzP --delete   {localDir}  root@{remoteHost}:{remoteDir}
[root@server0 ~]#rsync  -avzP  --delete  /tmp/rtest1  root@192.168.1.101:/tmp/
    //将本地主机的/tmp下的rtest1整个文件夹同步到远程主机(100)的/tmp下
[root@server0 ~]#rsync  -avzP  --delete  /tmp/rtest1/  root@192.168.1.101:/tmp/
    //区别是只同步rtest1目录下的数据，不会把rtest1也同步过去
```

一台服务器，不同目录之间进行同步：
格式：
    rsync -avzP --delete   新数据目录   旧数据目录
    [root@server0 ~]#rsync -avzP --delete /tmp/test_gitlab/web-godPen/
/opt/jenkins/rsync_d/
        //将新数据目录下的数据，同步至旧数据目录下，实现两个目录之间相同内容。
        （包括在新数据目录下，文件内容的增删改查及目录的增删改查）
    注：如果不加--delete，只会同步增加的内容，新目录中删除的内容，并不会同步到旧目录中。

忽略某个目录(忽略旧数据中某个目录或文件不进行同步)：
使用Rsync同步的时候往往会要求对某个文件夹或者文件进行忽略，客户端可以使用--exclude参数
来实现对目录或者文件的忽略

[root@server0 ~]#rsync   -avzP   --exclude   "a.txt"   --delete   /tmp/test_gitlab/
   /opt/jenkins/
注：必须是忽略在前，删除在后，否则没有意义!!!

二十五、磁盘阵列RAID模式优缺点

RAID0模式：

优点：在RAID 0状态下，存储数据被分割成两部分，分别存储在两块硬盘上，此时移动硬盘的理论存储速度是单块硬盘的2倍，
      实际容量等于两块硬盘中较小一块硬盘的容量的2倍。
    缺点：任何一块硬盘发生故障，整个RAID上的数据将不可恢复。
    备注：存储高清电影比较适合。

RAID1模式：

优点：此模式下，两块硬盘互为镜像。当一个硬盘受损时，换上一块全新硬盘(大于或等于原硬盘容量)替代原硬盘即可自动恢复资料和继续使用，移动硬盘的实际容量等于较小一块硬盘的容量，存储速度与单块硬盘相同。RAID 1的优势在于任何一块硬盘出现故障是，所存储的数据都不会丢失。
缺点：该模式可使用的硬盘实际容量比较小，仅仅为两颗硬盘中最小硬盘的容量。
备注：非常重要的资料，如数据库，个人资料，是万无一失的存储方案。

RAID 0+1模式：

RAID 0+1 是磁盘分段及镜像的结合，采用2组RAID0的磁盘阵列互为镜像，它们之间又成为一个RAID1的阵列。
硬盘使用率只有   50%，但是提供最佳的速度及可靠度。

RAID5 模式：

RAID5 不对存储的数据进行备份，而是把数据和相对应的奇偶校验信息存储到组成RAID5的各个磁盘上，并且奇偶校验信息和相对应的数据分别存储于不同的磁盘上。当RAID5的一个磁盘数据发生损坏后，利用剩下的数据和相应的奇偶校验信息去恢复被损坏的数据。

RAID 10模式：

RAID10 最少需要4块硬盘才能完成。把2块硬盘组成一个RAID1，然后两组RAID1组成一个RAID0。虽然RAID10方案造成了50%的磁盘浪费，但是它提供了200%的速度和单磁盘损坏的数据安全性。

二十六、watch命令

```
作用：
    用来监听数据的变换， 当数据模型 (data选项 M) 发生改变时， watch就会触发；
    周期性的执行指定命令，并以全屏方式显示结果。
参数：
    -n   指定周期长度，单位为秒，默认为2秒
    -d   高亮显示两次输出中不同的部分，这个功能相当实用
用法：
    [root@server0 ~]#watch   'cat   /etc/passwd'   //则每隔两秒执行一次cat /etc/passwd命令
    [root@server0 ~]#watch    -n  5  date         //每隔5秒执行一次date命令
    [root@server0 ~]#watch   -d  -n  3   date    //每隔3秒执行一次date命令，变化的地方高亮标示出来
```

二十七、crontab周期性计划任务

```
    crontab命令被用来提交和管理用户的需要周期性执行的任务，crond进程每分钟会定期检查是否有要执行的任务，如果有要执行的任务，则会自动执行。

    安装crontab：
        [root@server0 ~]#yum -y install crontabs cronie
        [root@server0 ~]#systemctl start crond    //centos7以上版本使用
        [root@server0 ~]#/sbin/service start crond    //centos7以下版本使用
        [root@server0 ~]#systemctl reload crond     //重新载入配置
        [root@server0 ~]#ntsysv       //查看crontab服务是否已设置为开机启动

    系统cron任务配置文件：
        /etc/crontab
    参数：
    -e  编辑该用户的时间计划任务；
    -l  列出该用户的时间计划任务；
    -r  删除该用户的时间计划任务；
    -u  指定要设定计划任务的用户；

    用户crond配置文件：
```

/var/spool/cron/zhangsan //可以直接创建用户名文件来设置时间计划任务，非交互式(用于脚本)

文件格式:

minuter(0~59) hour(0~23) day(1~31) month(1~12) week(0~7[0和7代表星期天]) command

在以上的各个字段中，还可以使用以下特殊字符:

星号(*):代表所有可能的值，例如:month字段如果是星号，则表示在满足其它字段的制约条件后每月都执行该命令的操作;

逗号(,):可以用逗号隔开的值指定一个列表范围;例如:“1，2，5，7，9”;

中杠(-):可以用整数之间的中杠表示一个整数范围，例如:“2-6”表示“2，3，4，5，6”;

正斜线(/):可以用正斜线指定时间的间隔频率，例如:“0-23/2”表示每两小时执行一次。同时正斜线可以和星号一起使用，例如:*/10,如果在minute字段，表示每十分钟执行一次;

案例:

```
[root@server0 ~]#crontab -e  -u  root
0,15,30,45  18-06  *  *  *  /usr/bin/echo  `date`  >>  /root/a.txt
```

保存退出

解析:

系统将在晚上6点到早上06点这段时间内，每隔15分钟向a.txt文件中追加一次当前时间;

新创建的配置文件的路径为:/var/spool/cron/zhangsan

```
[root@server0 ~]#crontab  -l     //列出当前用户的计划任务
[root@server0 ~]#crontab  -l  -u  zhangsan   //列出张三用户的计划任务
```

实例1:每1分钟执行一次command

命令:

```
* * * * * command
```

实例2:每小时的第3和第15分钟执行

命令:

```
3,15 * * * * command
```

实例3:在上午8点到11点的第3和第15分钟执行

命令:

```
3,15 8-11 * * * command
```

实例4:每隔两天的上午8点到11点的第3和第15分钟执行

命令:

```
3,15 8-11 */2 * * command
```

实例5:每个星期一的上午8点到11点的第3和第15分钟执行

命令:

```
3,15 8-11 * * 1 command
```

实例7:每月1、10、22日的4:45重启http

命令:c

```
45 4 1,10,22 * * /usr/bin/systemctl restart httpd
```

```
实例8：每周六、周日的1:10重启httpd
命令：
    10 1 * * 6,0 /usr/bin/systemctl restart httpd

实例9：每天18:00至23:00之间每隔30分钟重启http
命令：
    0,30 18-23 * * * /usr/bin/systemctl restart httpd

实例10：每星期六的晚上11:00 pm重启http
命令：
    0 23 * * 6 /usr/bin/systemctl restart httpd

实例11：每一小时重启httpd
命令：
    * */1 * * * /usr/bin/systemctl restart httpd

实例12：晚上11点到早上7点之间，每隔一小时重启http
命令：
    * 23-7/1 * * * /usr/bin/systemctl restart httpd
```

二十八、yum搭建

1、使用iso镜像配置本地yum源：

```
1、上传iso镜像到任意目录下（可自定义路径名称）；
2、挂载iso到/iso目录下（可以自定义路径）：
    [root@server0 ~]#mount /var/ftp/centos-1804.iso  /iso    //挂载镜像光盘到/iso目录下
3、将 /etc/yum.repos.d/ 下的repo文件都备份后删除，新建一个repo文件 比如：Centos-Base.repo
    [root@server0 ~]#vim  /etc/yum.repos.d/Centos-Base.repo
修改该文件：
[rhel7]
name=rhel7.5
baseurl=file:///iso      //指定yum仓库的路径
baseurl=http://content.example.com/rhel7.0/x86_64/dvd    //yum仓库路径也可以是挂载在http下的
gpgcheck=0               //0为不检测，1为检测，要更改为0，检测只是检测红帽官方打包的rpm包
enable=1                 //是否立即生效，1为是
[root@server0 ~]#yum clean all     //清空yum仓库缓存
[root@server0 ~]#yum repolist

如果想要每次开机都自动挂载，可以修改/etc/fstab 或者修改/etc/rc.d/rc.loacl文件。
```

2、自定义yum仓库

```
[root@server0 ~]# createrepo --version       //检测createrepo的软件包是否安装
未安装，需要下载相应的软件包和依赖包：
    1>先使用yum install命令安装libxml2-python-2.9.1-5.el7_0.1.x86_64.rpm
    2>再安装createrepo-0.4.11-3.el5.noarch.rpm（一定要按顺序安装，它们存在依赖关系）命令如下：
[root@server0 ~]# yum -y install libxml2-python-2.9.1-5.el7_0.1.x86_64.rpm
[root@server0 ~]# yum -y install createrepo-0.4.11-3.el5.noarch.rpm
    3>在根目录root下创建/pk目录，用于存放所需依赖包：
[root@server0 ~]# mkdir /pk     //将需要放入yum仓库的rpm包放入pk目录下
[root@server0 ~]# cd /pk        //切换到pk目录下
[root@server0 ~]# createrepo /pk/      //配置成功后pk目录下会有一个依赖关系的文件
[root@server0 ~]#vim  /etc/yum.repos.d/Centos7.repo
[rhel7]
name=rhel7.5
baseurl=file:///pk       //指定yum仓库的路径
gpgcheck=0                //0为不检测，1为检测，要更改为0，检测只是检测红帽官方打包的rpm包
enable=1                  //是否立即生效，1为是
```

二十九、监控系统的命令 (free ， df，top，uname，uptime)

1、free命令

free - 显示系统已用及空余物理内存量、交换分区使用情况（swap memory）、内核占用的缓存、及共享内存；

free命令显示了当前系统内存使用情况，其数据取自/proc/meminfo文件，数据以kb为单位；

```
其命令形式为 free + options(可多个参数)；
参数：
-h   (-human)自动将数值转换为人类易读的形式；
-c   (-count)展示结果次数，需与-s配合使用；
-s   (-secounds)动态刷新内存使用时间的间隔；
-m   以MB为单位显示当前内存的使用情况；
实例：
[root@server0 ~]#free  -h     //数据后面带有单位
[root@server0 ~]#free  -m     //以Mb显示，不带单位
[root@server0 ~]#free -h -c 2 -s 4     //每隔四秒显示一次内存使用情况，显示两次
[root@server0 ~]#cat /proc/meminfo
```

2、df 命令

df 命令用于显示目前在Linux系统上的文件系统的磁盘使用情况;

```
[root@server0 ~]#df  -h    //以人类易读的方式显示

文件系统        容量  已用  可用 已用% 挂载点
/dev/vda1        10G  3.1G  7.0G   31% /
devtmpfs        906M     0  906M    0% /dev
tmpfs           921M   80K  921M    1% /dev/shm
tmpfs           921M   17M  904M    2% /run
tmpfs           921M     0  921M    0% /sys/fs/cgroup
```

3、top命令

top命令可以查看各个进程的CPU使用率和内存使用率，类似Windows的任务管理器；

```
[root@server0 ~]#top
top - 14:57:01 up  4:13,  2 users,  load average: 0.00, 0.01, 0.05
Tasks: 124 total,   2 running, 122 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2.7 us,  0.3 sy,  0.0 ni, 97.0 id,  0.0 wa,  0.0 hi,  0.0 si,
KiB Mem:    836720 total,   706356 used,   130364 free,      352 buffer
KiB Swap:        0 total,        0 used,        0 free.   236568 cached

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+
 1573 gdm       20   0 1350976 104660  33288 S  2.7 12.5   0:24.47
  597 root      20   0  160860  17968   6468 S  0.3  2.1   0:01.90

注意:
    load average: 0.00, 0.01, 0.05  ---》 load average后面的三个数三个数值分别为  1分钟、5分钟、15分钟前到现在的平均值。


//显示内容动态刷新，每3秒一次，默认按cpu占用率排序（按进程对cpu的占用率从高到低排序），RES：使用内存（单位KB），PID：进程号（如需结束进程使用 kill 进程号）

键盘按键操作：（区分大小写）
M(大写)：按内存占用率排序（从高到低）
P(大写)：按cpu占用率排序
q(小写)：退出

[root@server0 ~]#top  -bn1   //静态显示(打印所有进程)
```

4、uname命令

```
[root@server0 ~]#uname -r   //查看内核版本
[root@server0 ~]#uname -v
    #1 SMP Wed Nov 19 10:24:30 CST 2014
SMP：对称多处理机，表示内核支持多核、多处理器
Wed Nov 19 10:24:30 CST 2014 ：内核的编译时间(build date)为(2014/11/19 10:24:30)

[root@server0 ~]#uname -p     //该属性表示该机器处理器的类型(CPU)
[root@server0 ~]#uname -m     //查看硬件名称，【x86_64:64位系统】，【ix86:32位系统(x表示3、4、5、6)】
[root@server0 ~]#uname -o     //查看操作系统类型
[root@server0 ~]#uname -n      //查看主机名
[root@server0 ~]#uname -a   //显示全部信息
```

5、uptime命令

每当系统变慢时，我们做的第一件事就是执行top或uptime，来了解下负载情况

```
[root@server0 ~]#uptime
16:02:29 up  5:18,  2 users,  load average: 0.00, 0.01, 0.05
16:02:29        //当前时间
up 2 days, 5:18    //系统运行时间
3 user    //正在登录用户数
最后三个数呢？依次是 1分钟，5分钟，15分钟的平均负载（load average）
//一个单核系统上，平均负载为1.78，0.60，6.56，说明1分钟内，系统有78%的超载，而在15分钟内有556%的超载；
//平均负载高于CPU数量70%时，就应该分析排查负载高问题了。
```

三十、linux常用命令

1、ping命令

```
ping  -c  测试次数   -i  间隔时间  -W  测试失败后反馈时间
-c   ping指定次数后停止ping；
-i   设定间隔几秒发送一个ping包，默认一秒ping一次；
-W(大写)  等待回复的时间，单位是毫秒，这个选项只在没有接到任何的回复的情况下有效；
[root@server0 ~]#ping  -c 3 -i 0.3  -W 1  172.25.0.11
```

2、file命令

```
file命令用来查看文件的类型；
[root@server0 ~]# file /etc/shadow
/etc/shadow: ASCII text
```

3、du命令

```
```