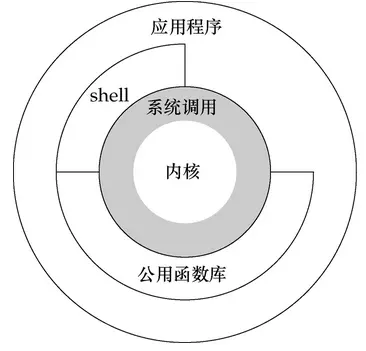
# Linux基础



**内核**分为 进程管理系统 、 内存管理系统 、 I/O管理系统 和文件管理系统 等四个子系统

终端是哪个文件夹下的哪个文件？终端  **/dev/tty**

**DNS实际上是分布在internet上的主机信息的**[数据库](http://www.linux521.com/2009/database/),**其作用是实现 IP地址和主机名之间的转换**

在[Linux系统](http://www.linux521.com/2009/system/)中,**以文件方式访问设备**

全部磁盘块由四个部分组成,分别为引导块 、专用块 、 i节点表块 和数据存储块.

**退出当前命令**： ctrl+c 彻底退出

## 状态

**运行态、就绪态和等待态**

linux上进程有5种:

1. 运行(正在运行或在运行队列中等待)

2. 中断(休眠中, 受阻, 在等待某个条件的形成或接受到信号)

3. 不可中断(收到信号不唤醒和不可运行, 进程必须等待直到有中断发生)

4. 僵死(**进程已终止, 但进程描述符存在**, 直到父进程调用wait()系统调用后释放)

5. 停止(进程收到SIGSTOP, SIGSTP, SIGTIN, SIGTOU信号后停止运行运行)

## 僵尸进程

**在UNIX 系统中，一个进程结束了，但是他的**[**父进程**](https://baike.baidu.com/item/%E7%88%B6%E8%BF%9B%E7%A8%8B/614062)**没有等待(调用wait / wait pid)他， 那么他将变成一个僵尸进程。**

但是如果该进程的父进程已经先结束了，那么该进程就不会变成僵尸进程

原因：

因为每个进程结束的时候，系统都会扫描当前系统中所运行的所有进程， 看有没有哪个进程是刚刚结束的这个进程的**子进程，如**果是的话，**就由Init (pid = 1)来接管他，成为他的父进程。**

## Linux系统的组成

1. **linux内核**（linus 团队管理）
2. **shell、bash（命令解析器）**：用户与内核交互的接口；**本质：根据命令，调用对应的可执行程序**
3. **文件系统**：ext3、ext4等。windows 有 fat32  、ntfs
4. **第三方应用软件**

Shell是**系统的用户界面**，提供了**用户与内核进行交互操作的一种接口(命令解释器)。**

使用type命令可以**区分**内部命令和外部命令，**使用ps命令观察正在执行的shell**

在Linux中，**可执行的文件**也进行了分类：

* **内置命令**：出于效率的考虑，将一些常用命令的解释程序**构造在Shell内部**。
* **外置命令**：存放在**/bin、/sbin目录**下的命令
* **实用程序**：存放在/usr/bin、/usr/sbin、/usr/share、/usr/local/bin等目录下的实用程序
* **用户程序**：用户程序经过**编译生成可执行文件**后，可作为Shell命令运行
* **Shell脚本**：由Shell语言编写的批处理文件，可作为Shell命令运行

# Linux目录

|  |  |
| --- | --- |
| 目录 |  |
| /bin | 存放二进制可执行文件(ls,cat,mkdir等)，**常用命令一般都在这里**。 |
| /etc | **存放系统管理和配置文件** |
| /home | **存放所有用户文件的根目录，是用户主目录的基点**，比如用户user的主目录就是/home/user，可以用~user表示 |
| /usr | 用于存放系统应用程序，比较重要的目录/usr/local 本地系统管理员软件安装目录（**安装系统级的应用**）。这是最庞大的目录，要用到的应用程序和文件几乎都在这个目录。  /usr/x11r6 存放x window的目录  **/usr/bin 众多的应用程序**  **/usr/sbin 超级用户的一些管理程序**  /usr/doc linux文档  /usr/include linux下开发和编译应用程序所需要的头文件  /usr/lib 常用的动态链接库和软件包的配置文件  /usr/man 帮助文档  /usr/src 源代码，linux内核的源代码就放在/usr/src/linux里  /usr/local/bin 本地增加的命令  /usr/local/lib 本地增加的库 |
| /opt | 额外安装的可选应用程序包所放置的位置。一般情况下，我们可以把tomcat等都安装到这里。 |
| /proc | 虚拟文件系统目录，**是系统内存的映射。**可直接访问这个目录来获取系统信息。 |
| /root | **超级用户**（系统管理员）的主目录（特权阶级^o^） |
| /sbin | 存放二进制可执行文件，只有root才能访问。这里存放的是系统管理员使用的系统级别的管理命令和程序。如ifconfig等。 |
| **/dev** | **用于存放设备文件。** |
| /mnt | 系统管理员安装临时文件系统的安装点，系统提供这个目录是让用户临时挂载其他的文件系统。 |
| /boot | 存放用于系统引导时使用的各种文件 |
| /lib | 存放跟文件系统中的程序运行所需要的共享库及内核模块。共享库又叫动态链接共享库，作用类似windows里的.dll文件，存放了根文件系统程序运行所需的共享文件。 |
| /tmp | 用于存放各种临时文件，是公用的临时文件存储点。 |
| /var | 用于存放运行时需要改变数据的文件，也是某些大文件的溢出区，比方说各种服务的**日志文件**（系统启动日志等。）等。 |
| /lost+found | 这个目录平时是空的，系统非正常关机而留下“无家可归”的文件（windows下叫什么.chk）就在这里 |

# 常用的文件、目录操作命令

linux有2种用户,一种管理员root,另外一种普通:xz

Root用户登录到linux默认所在的目录是 /root/

其他用户xz登录到linux默认所在的目录是 /home/xz/

tips:**输入命令的时候要常用tab键来补全，按两次显示所有**

## ls命令

* ls：显示文件或目录信息
* 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

## cd命令

* cd ~：当前用户目录
* cd /：根目录
* cd -：上一次访问的目录
* cd ..：上一级目录
* cd ：缺省当前用户目录
* **touch 1.txt** :在当前目录创建一个文件1.txt，文件存在，修改时间

## [wc命令用法](https://www.cnblogs.com/newcaoguo/p/5896491.html)

Linux系统中的wc(Word Count)命令的功能为统计指定文件中的字节数、字数、行数，并将统计结果显示输出。

1．命令格式：

wc [选项]文件...

2．命令功能：

统计指定文件中的字节数、字数、行数，并将统计结果显示输出。该命令统计指定文件中的字节数、字数、行数。如果没有给出文件名，则从标准输入读取。wc同时也给出所指定文件的总统计数。

3．命令参数：

-c 统计字节数。

-l 统计行数。

-m 统计字符数。这个标志不能与 -c 标志一起使用。

-w 统计字数。一个字被定义为由空白、跳格或换行字符分隔的字符串。

-L 打印最长行的长度。

-help 显示帮助信息

--version 显示版本信息

## find命令

**whereis [-bmsu] [BMS 目录名 -f ] 文件名**

**1]根据文件名查找**  
-name    按照文件名查找   
-iname   根据文件名查找，但是不区分大小写

$find / -user fred      //查找在系统中属于FRED这个用户的文件    
-prune    不在当前指定的目录中查找   
-depth    在查找文件时，首先查找当前目录中的文件，然后再在其子目录中查找

**2]根据文件所属用户和组来查找文件**-user     按照文件属主来查找文件

$find / -user fred      //查找在系统中属于FRED这个用户的文件

-group   按照文件所属的组来查找文件

**3]根据uid 和 gid来查找用户**

#find  /tmp -uid 500       //查找uid是500 的文件 

**[4]-a,-o,-not的使用**-a      连接两个不同的条件（两个条件必须同时满足）

#find /tmp -name "\*.sh" -a -user root

-o     连接两个不同的条件（两个条件满足其一即可）

#find /tmp -name "\*.sh" -o -user root

-not 对条件取反的

**[5]根据文件时间戳的相关属性来查找文件**-atime   最近一次访问时间      单位：天  
-mtime 最近一次内容修改时间  单位：天  
-ctime  最近一次属性修改时间  单位：天  
-amin   最近一次访问时间      单位：分钟  
-mmin  最近一次内容修改时间  单位：分钟  
-cmin   最近一次属性修改时间  单位：分钟  
-newer file1 ! file2 查找更改时间比文件file1新但比文件file2旧的文件

**[6]根据文件类型来查找文件**-type 查找某一类型的文件   
文件类型：   
f 普通文件   
d 目录   
l 符号链接文件   
c 字符设备文件   
p 管道文件   
b 块设备文件  
s socket文件

**[7]根据大小来查找文件**-size n[c] 查找文件长度为n块的文件，带有c时表示文件长度以字节计

#find /tmp -size  2M           //查找在/tmp 目录下等于2M的文件

**[8]根据文件权限查找文件**

-perm

#find /tmp -perm 755           //查找在/tmp目录下权限是755的文件

**命令举例**

**搜索/etc目录下的文件名包含del的文件。**

**$find /etc -name '\*del\*'**

系统查找到"config.py"文件后即时在屏幕上显示"config.py"文件信息。

$find . -name "config.py" -ls

搜索当前目录中，所有过去10分钟中更新过的普通文件。如果不加-type f参数，则搜索普通文件+特殊文件+目录。

$ find . -type f -mmin -10

## ps命令（查看进程）

* 查看进程：显示所有进程信息**ps -A**
* 显示所有进程信息，连同命令行 **命令：ps -ef**
* **ps -ef :相当于任务管理器**,对进程的一次查看
* **ps 与grep 常用组合用法，查找特定进程 命令：ps -ef|grep ssh**
* 在**超级用户**下显示Linux系统中**正在运行的全部进程**,应使用的命令及参数是 **ps** -aux .
* grep: 过滤任务 经常和其他命令联合使用
* grep -i 名称 **-i** 忽略字符大小写的差别

## netstat命令

**显示网络连接、路由表和网络接口信息，可以让用户得知目前都有哪些网络连接正在运作。**

**该命令的一般格式为：  
      netstat [选项]  
      命令中各选项的含义如下：  
      -a 显示所有socket，包括正在监听的。  
      -c 每隔1秒就重新显示一遍，直到用户中断它。  
      -i 显示所有网络接口的信息，格式同“ifconfig -e”。  
     -n 以网络IP地址代替名称，显示出网络连接情形。  
      -r 显示核心路由表，格式同“route -e”。**

**-l : 仅显示监听套接字(所谓套接字就是使应用程序能够读写与收发通讯协议(protocol)与资料的程序)**

**-p : 显示进程标识符和程序名称，每一个套接字/端口都属于一个程序。**

**-t 显示TCP协议的连接情况。  
      -u 显示UDP协议的连接情况。  
      -v 显示正在进行的工作。**

* **端口是否被占用**

**1.netstat  -anp  |grep   端口号**

**如下，我以3306为例，netstat  -anp  |grep  3306**

**查看具体端口时候，必须要看到tcp，端口号，LISTEN那一行，才表示端口被占用了**

**2.netstat   -nultp（此处不用加端口号）**

**该命令是查看当前所有已经使用的端口情况**

**3. 查看端口被某一进程占用可以使用 netstat -ano|findstr "port" 命令，此命令可以找到占用进程的pid。查询占用了8080端口的进程：netstat -ano|findstr "8080"**

**然后使用 tasklist|findstr "pid" 找到进程名称;**

* **然后使用 taskkill /f /t /im 进程名称 。**
* **taskkill /f /PID 5769    杀PID**

## Awk

是一个强大的文本分析工具

## mkdir

* **pwd:打印工作目录，**PrintWorkingDirectory
* **which 只能查可执行文件**
* **whereis 只能查二进制文件、说明文档，源文件等**
* 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，放在后面也可以-p。

**其父目录不存在时先创建父目录-p**

* rmdir：要求目录为空

## 文件管理

**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

**cat f1.txt > f2.txt 把f1.txt复制为f2.txt**

**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：**移动文件或目录、文件或目录改名**，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

**硬链接(hard link)与软链接(symbolic link)**，硬链接的意思是一个档案可以有多个名称，而软链接的方式则是产生一个特殊的档案(指向同一个文件节点)，该档案的内容是指向另一个档案的位置。

硬链接是存在同一个文件系统中，而软链接却可以跨越不同的文件系统。

**软链接：**

1.**软链接，以路径的形式存在。类似于Windows操作系统中的快捷方式(该档案的内容是指向另一个档案的位置。)**2.软链接可以 跨文件系统 ，硬链接不可以  
3.软链接可以对一个不存在的文件名进行链接  
4.软链接可以对目录进行链接

**硬链接:**

1.**硬链接，以文件副本的形式存在。但不占用实际空间。**  
2.不允许给目录创建硬链接  
3.硬链接只有在同一个文件系统中才能创建

* ln：建立链接文件，ln -s 源文件 目标文件。
* file/stat：查看文件类型或文件属性信息

### chomd（四段）

文件类型 文件所有者的权限  文件所有者所在组的权限 其他用户

**文字设定法**

chmod ［who］ ［+ | – | =］ ［mode］ **文件名**

操作对象who可是下述字母中的任一个或者它们的组合：

* **u 表示“用户（user）”，即文件或目录的所有者。**
* **g 表示“同组（group）用户”，即与文件属主有相同组ID的所有用户。**
* **o 表示“其他（others）用户”。**
* a 表示“所有（all）用户”。它是系统默认值。

操作符号可以是：

* + 添加某个权限。
* – 取消某个权限。
* = 赋予给定权限并取消其他所有权限（如果有的话）。
* 设置mode所表示的权限可用下述字母的任意组合：
* r 可读。
* w 可写。
* x 可执行。

例子：chmod a+x sort

chmod ug+w，o-x text

**即设定文件text的属性**为：

文件属主（u） 增加写权限

与文件属主同组用户（g） 增加写权限

其他用户（o） 删除执行权限

**数字设定法**

其中每一位的权限用数字来表示。具体有这些权限：

* r(Read，读取，权限值为4)：对文件而言，具有读取文件内容的权限；对目录来说，具有浏览目 录的权限。
* w(Write,写入，权限值为2)：对文件而言，具有新增、修改文件内容的权限；对目录来说，具有删除、移动目录内文件的权限。
* x(eXecute，执行，权限值为1)：对文件而言，具有执行文件的权限；对目录了来说该用户具有进入目录的权限。

**如果你的umask设置为022。默认777,就是755（文件夹权限），文件权限再减111。**

[**chown**](http://yusi123.com/tag/chown)**命令**

功能：更改某个文件或目录的**属主和属组（可以用： ）**。这个命令也很常用。例如root用户把自己的一个文件拷贝给用户yusi，为了让用户yusi能够存取这个文件，root用户应该把这个文件的属主设为yusi，否则，用户yusi无法存取这个文件。

**语法：**[chown](http://yusi123.com/tag/chown) ［选项］ **用户或组文件**

[**chgrp**](http://yusi123.com/tag/chgrp)**命令**

功能：改变文件或目录所属的组。

语法：[chgrp](http://yusi123.com/tag/chgrp) ［选项］ group filename¼

## 文件浏览命令

* **cat：查看文本文件内容，如果文件一页之内可以显示完毕，cat 文件名称**
* **more：可以分页看，more 文件名称，只能向后翻页；**
* **less：不仅可以分页，还可以方便地搜索（/ 文件），回翻等操作，less -mN 文件名称。q:退出**
* tail -10： 查看文件的尾部的10行，tail -10 /root/install.log 查看install.log文件中后10行内容
* head -20：查看文件的头部20行
* dd:删除当前行（0：返回行首）  
  ndd:删除光标行往下n行(含光标行)内容  
  **dgg:删除光标当前行及以上内容**  
  dG:删除光标当前行及以下内容  
  dH:删除当前页面第1行至光标行  
  p:粘贴到光标下一行  
  u:撤销一次操作  
  Ctrl + r:反撤销（一次）

## 解压

tar是操作.tar的命令

gzip是压缩.gz压缩包的命令

compress：压缩.Z文件

uncompress：解压缩.Z文件

tar

**-c: 建立压缩档案**

**-x：解压**

-t：查看内容

# tar -tf all.tar

这条命令是列出all.tar包中所有文件，-t是列出文件的意思

-r：向压缩归档文件末尾追加文件

# tar -rf all.tar \*.gif

这条命令是将所有.gif的文件增加到all.tar的包里面去。-r是表示增加文件的意思。

-u：更新原压缩包中的文件

# tar -uf all.tar logo.gif

这条命令是更新原来tar包**all.tar中logo.gif文件**，-u是表示更新文件的意思。

这五个是独立的命令，压缩解压都要用到其中一个，可以和别的命令连用但只能用其中一个。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属性的

-j：有bz2属性的

-Z：有compress属性的

-v：显示所有过程

-O：将文件解开到标准输出

**下面的参数-f是必须的**

-f: 使用档案名字，切记，这个参数是最后一个参数，后面只能接档案名。

## 系统命令

* **echo：把内容重定向到指定的文件中 ，有则打开，无则创建**

**echo "cmd1" | at midnight**

* 管道命令 | ：**将前面的结果给后面的命令**，例如：ls -la | wc，将ls的结果给wc命令来统计字数
* 重定向 **> 是覆盖模式，>> 是追加模式，**例如：echo "zhen de hen xihuan ni" > qingshu.txt把左边的输出放到右边的文件里去。
* **杀死进程 kill -9 进程id**
* **查看IP: ifconfig。首先使用ifconfig查看以太网地址和硬件地址，然后根据以太网的inet地址进行ping。（ping inet地址）**
* **查看机器是否连接通畅: ping 对方机器ip**

**sends SIGTERM to the process whose PID IS 9.**

* **wc命令 查看文件有多少行**
* **我（客户端）上传，你（服务器）接收（RZ），我下载（客户端），你（服务器）发送（SZ）。**

# Vim

基本上vi可以分为三种状态，分别是命令模式（command mode）、插入模式（Insert mode）和底行模式（last line mode），各模式的功能区分如下：   
    1) 命令行模式command mode）   
　　控制屏幕光标的移动，字符、字或行的删除，移动复制某区段及进入Insert mode下，或者到 last line mode。   
    2) 插入模式（Insert mode） **按i键进入**  
　　只有在Insert mode下，才可以做文字输入，按「ESC」键可回到命令行模式。   
    3) 底行模式（last line mode）   
　　将文件保存或退出vi，也可以设置编辑环境，如寻找字符串、列出行号……等。

**:wq 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。**

vim filename **打开vim并创建名为filename的文件**

**/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

Ctrl + r 重做（Redo），即撤销的撤销。

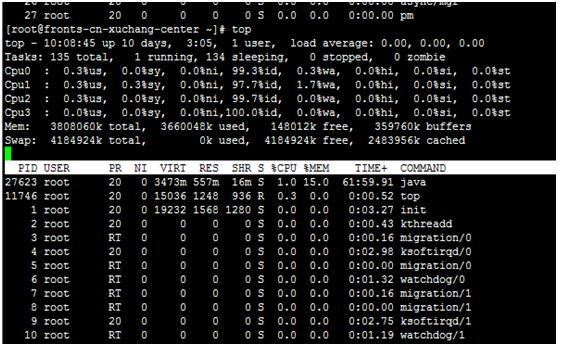
yy 拷贝当前行

nyy 拷贝当前后开始的n行，比如2yy拷贝当前行及其下一行。

# Top（性能分析工具）

TOP命令是Linux下常用的性能分析工具，能够实时显示系统**中各个进程的资源占用状况。**

op进入视图



**第一行**

10:08:45 — 当前系统时间；10 days, 3:05 — 系统已经运行了10天3小时5分钟（在这期间没有重启过）； 1 users — 当前有1个用户登录系统；

**load average**: 0.00, 0.00, 0.00 — load average后面的三个数分别是**1分钟、5分钟、15**分钟的**负载情况。**

load average数据是**每隔5秒钟**检查一次活跃的进程数，然后按特定算法计算出的数值。如果这个数**除以**逻辑**CPU**的数量，**结果高于5的时候就表明系统在超负荷运转了。**

所谓**CPU负载**指的是一段时间内任务队列的长度，通俗的讲，就是一段时间内一共有多少任务在使用或等待使用CPU。  
  
  **第二行：进程**  
    Tasks — 任务（进程），系统现在共有135个进程，其中处于运行中的有1个，134个在休眠（sleep），stoped状态的有0个，zombie状态（僵尸）的有0个。  
  
   **第三行：cpu状态**  
    0.3% us — 用户空间占用CPU的百分比。  
    0.0% sy — **内核空间占用CPU的百分比。**  
    0.0% ni — 改变过优先级的进程占用CPU的百分比  
    99.7% id — 空闲CPU百分比  
    0.0% wa — IO等待占用CPU的百分比  
    0.0% hi — 硬中断（Hardware IRQ）占用CPU的百分比  
    0.0% si — 软中断（Software Interrupts）占用CPU的百分比  
在这里CPU的使用比率和windows概念不同，如果你不理解用户空间和内核空间，需要充充电了。

**第四行：内存状态**  
    3808060k total — 物理内存总量（4GB）  
    3660048k used — 使用中的内存总量（3.6GB）  
    148012k free — 空闲内存总量（148M）  
    359760k buffers — 缓存的内存量 （359M）

**第五行：swap交换分区**  
    4184924k total — 交换区总量（4G）  
    0k used — 使用的交换区总量（0M）  
    4184924k free — 空闲交换区总量（4G）  
    **2483956k cached — 缓冲的交换区总量（2483M）**

第四行中使用中的内存总量（used）指的是现在**系统内核控制的内存数**，空闲内存总量（free）是内核还未纳入其管控范围的数量。纳入内核管理的内存不见得都在使用中，还包括过去使用过的现在可以被重复利用的内存，内核并不把这些可被重新使用的内存交还到free中去，因此在linux上free内存会越来越少，但不用为此担心。

**如果出于习惯去计算可用内存数**，这里有个近似的计算公式：第四行的free + 第四行的buffers + 第五行的cached，按这个公式此台服务器的可用内存：

148M+359M+2483M = 2990M。

对于内存监控，在top里我们要时刻监控第五行swap交换分区的used，如果这个数值在不断的变化，说明内核在不断进行**内存和swap的数据交换**，这是真正的内存不够用了。

**第七行以下：各进程（任务）的状态监控**  
    PID — 进程id  
  USER — 进程所有者  
  **PR — 进程优先级**  
    NI — **nice值**。负值表示高优先级，正值表示低优先级  
    VIRT — 进程使用的**虚拟内存总量**，单位kb。VIRT=SWAP+RES  
    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  
    S — 进程状态。D=不可中断的睡眠状态 R=运行 S=睡眠 T=跟踪/停止 Z=僵尸进程  
    %CPU — 上次更新到现在的CPU时间占用百分比  
    %MEM — **进程使用的物理内存百分比**  
    TIME+ — 进程使用的CPU时间总计，单位1/100秒  
    COMMAND — 进程名称（命令名/命令行）

# 面试

* **查看计算机内存：**

free -h 或者  free -m

　　-h是人性化显示，-m是以M为单位显示

* **查看CPU的核数**

**cat /proc/cpuinfo | grep "cpu cores"**

* **禁止所有普通用户登录 touch /etc/nolgoin**

**MIN HOUR DAY MONTH DAYOFWEEK** COMMAND

* **export命令：为其它应用程序设置环境变量**
* **当前目录下还有多大空间：Use du .**
* **不重启系统就可转换到 level 5 ：telinit 5**
* **Mount：装载文件系统**
* **make install：需要root权限**

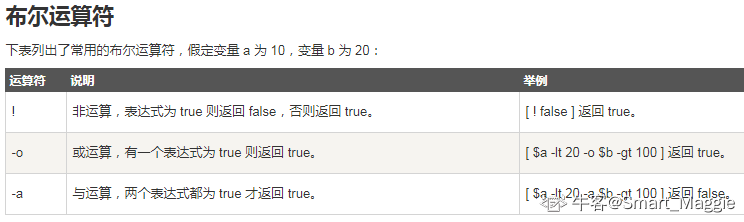
**rpm命令是RPM软件包的管理工具。**

* **文件属于哪个包 ：rpm -qf /etc/my.conf**

Linux系统大致启动流程是：**uEFI-> MBR -> grub -> initrd**

**Linux的启动信息：dmesg**

查找当前目录一个月(30天)以前大于100M的日志文件(.log)并删除：find . -name "\*.log" –mtime +30 –type f –size +100M |xargs **rm –rf {}**



## 用shell统计访问日志里每个ip访问次数

访问日志情况：

[root@qular ~]# cd /usr/local/nginx/logs/

[root@qunlar logs]# head access.log

开始写脚本：

 [root@qunar logs]# cat a.sh

#!/bin/bash

#将28/Jan/2015全天的访问日志放到a.txt文本

cat access.log |sed -rn '/28\/Jan\/2015/p' > a.txt

#统计a.txt里面有多少个ip访问

cat a.txt |awk '{print $1}'|sort |uniq > ipnum.txt

#通过shell统计每个ip访问次数

for i in `cat ipnum.txt`

do

iptj=`cat  access.log |grep $i | grep -v 400 |wc -l`

echo "ip地址"$i"在2015-01-28日全天(24小时)累计成功请求"$iptj"次，平均每分钟请求次数为："$(($iptj/1440)) >> result.txt

done

执行脚本：

 [root@qunar logs]#sh a.sh

查看结果:

 [root@qular logs]# cat result.txt