# LINUX基础知识

## 查看系统PCI设备

lspci

安装lspci命令：

yum install -y pciutils 安装lspci命令

## LINUX图像界面安装

首先安装X(X Window System)命令

yum groupinstall “X Window System”

检查一下我们已经安装的软件以及可以安装的软件

yum grouplist

安装GNOME Desktop

yum groupinstall “GNOME Desktop”

开启图像界面

startx

### 查看CPU信息

more /proc/cpuinfo more读取/proc/cpuinfo 或者cat读取都行

lscpu 命令

### 查看系统内存信息

more /proc/meminfo

### 查看磁盘分区情况

fdisk -l

## 目录功能介绍

etc：这个目录主要用于存放系统管理相关配置文件以及子目录

usr目录：此目录主要用于存放应用程序和文件，软件包默认会安装到此目录下

var目录：此目录主要用于存放系统运行以及软件运行的日志信息；

dev目录：此目录包含系统所有的设备文件

proc目录：虚拟目录，目录中所有的信息是内存的映射，存在于内存中，而不是硬盘上。（进程目录）当前进程相关信息

## 启动，停止，重启服务？？

要通过systemctl命令启动一个服务

systemctl start ……开启

systemctl stop …….. 停止

systemctl restart ……. 重启

## 开机自启开启和关闭

enable/disable

systemctl enable …….. 开启开机自启

systemctl disable ………禁止开机自启

LINUX的基本常用命令

## date命令

显示或修改系统时间

语法：date [选项] 显示时间的格式（以+开头，后门接时间格式）

date ‘+%Y-%m-%d %H:%M:%S’显示为当前的时间

## Mkdir 命令

参数-m创建时设置权限，-p递归创建目录

-m：配置文件的权限！直接配置，不需要看默认权限（umask）的脸色

-p：帮助你直接将所需要的目录（包涵上一级目录）递归创建起来

例如：

Mkdir -p test1/test2/test3/test4进入目录中的下一个目录可用 cd xx 假如我在test1目录中我现在要去test1目录中的test2目录则可以cd test2进入test2目录

如果不加-p就会报错，无法递归的创建，加了-p就可以自行帮我创建多层目录了

## rmdir命令

用来删除目录的命令，只能删除空目录，不能删除非空目录，如要删除，可以先删除文件后成为空目录再删除，，或者使用rm –r命令。

## more命令

参数【-d,-s,-p,-c】

按空白键（space）下一页，按b就会往回（back）一页显示

-c 参数，显示文件时，每屏显示都清理屏幕先前的信息。

-num一次显示的行数

+/pattern在每个文档显示前搜寻该字符串（pattern），然后从该字符串之后开始显示

+num从第num行开始显示

例子：

More +20 -10 testfile从第20行开始显示，每此显示10行testfile文档内容

More +/liu test.log从文件中查找第一个出现“liu”字符串的行，并从该出前两行开始显示输出

## tail和head命令

假如我们只想看部分文件呢？比如只看开头和结尾？

tail查看文件最后几行内容，默认10行。加入-n可以修改行数tail -n 2 file 查看最后2行

head查看文件头几行，默认10行。同上，可以修改行数

## diff命令

diff命令用于来比较文件的差异，以逐行的方式比较文本文件的异同

例如：diff test1 test2 比较test1和test2两个文本内容的差异

## grep命令

LINUX下的过滤工具，筛选

举个简单的例子：grep RPM install.log 列出install.log文件中包含RPM字符串的行

-v :反向搜索

-n :显示行号 grep -n xxx test2

-e :多个匹配模式 grep -e t -e f test2

## rm命令

用来删除某个目录及其下的所有子目录及文件，但是对于链接，只是断开了链接，源文件保持不变的哈。

参数r,f,I需要注意LINUX没有类似windows一样的回收站，一旦文件删除就无法恢复哦。

-r递归删除，且删除目录要加参数-r，删除普通文件可以不用加r

-f 不给提示，适用于大量文件删除，否则一个一个提示效率不高。

-i交互式删除，即在删除前要经过确认。

## touch命令

用来改变指定文件的访问时间和修改时间，若指定文件不存在则创建此文件，如果没有指定时间，则使用当前时间。

参数说明：

a:改变档案的读取时间记录,无需设置时间选项

m:改变档案的修改时间记录，无需设置时间选项

c:假如目的档案不存在，不会建立新的档案，也不提示

R:使用参考档的时间记录

d:设定时间与日期，可以使用各种不同的格式

t设定档案的时间记录

--no-create：不会建立新档案

--version列出版本信息

例如：

创建没有的文件testfile,并修改时间属性为当前系统时间

touch testfile 创建testfile文件（前提是testfile不存在）

ls -l testfile 查看testfile文件的属性

touch testfile 修改testfile文件时间属性为当前系统时间

ls -l testfile 再次查看testfile文件的属性

## ls命令

ls -F(大写)🡪在目录后面加了正斜线（/），可执行文件后加了星号



-a 显示隐藏文件

通常情况下，隐藏文件是文件名以点号开始的文件

-R参数显示递归目录，包含子目录

-l 列出的文件类型中，目录（d）、文件（-）、字符型文件（c）、块设备（b）

-i 参数：查看文件或者目录inode编码

-d：仅列出目录本身，而不是列出目录内的文件数据（常用）

## cat命令

Cat命令是Linux下的文本输出命令、通常是用于观看某个文件的内容、同时cat还可以用于连接合并文件

Cat主要有三大功能：

1. 一次显示整个文件

Cat filename

1. 从键盘创建一个文件

Cat > filename只能创建新文件，不能编辑已有文件、使用cat创建文件时，以系统默认的文件属性作为新文件的属性、并接受键盘输入作为文件的内容。输入结束时按ctrl + d退出并保存文件，如果要清除文件内容、cat /dev/null> /tmp/file4 清除tmp目录下的file4文件中的内容

1. 将几个文件合并为一个文件

Cat file1 file2 > file

## ln命令

ln命令用来在文件或目录之间创建连接；一种是‘硬链接’（Hard Link）一种是‘软连接也叫符号连接’（Symbolic Link）

参数：

-f ：如果在目标位置存在与连接名相同的文件，这个文件将被删除

-s ：进行软连接

-d：允许系统管理员硬链接自己的目录

-b：对将在连接时会被覆盖或者删除的文件进行备份

举例：

将/etc/inittab文件进行硬链接到/opt/backup目录下，此时/opt/backup下就生成一个innttab文件，可以看到两个文件没有任何不同

[root@192 ~]# ln /etc/inittab /opt/backup

## file命令

用来显示文件的类型，对于长度为0的文件，将识别为空文件，对于符号连接文件，默认情况下将显示符号连接引用的真实文件路径。

file 【选项】文件名

参数：

-b 显示文件类型结果，不显示对应文件名称

-L 直接显示符号连接所指向文件的类型

-z 显示压缩文件的信息

-i 如果文件不是常规文件，则进一步对文件类型进行分类

举例：

显示/etc/init.d/halt文件类型

[root@192 etc]# file /etc/init.d/halt

## cp命令

cp命令用来给出文件或者目录复制到另一个文件或者目录中，类似windows下的copy命令

参数：

-a 在复制目录时使用，他保留所有信息，包含文件链接，文件属性，并递归复制目录

-d:若来源文档为连结档的属性（link file）、则复制连结档属性而非文件本身

-f:为强制的意思、若目标文件已经存在且无法开启，则移除后再尝试一次

-i:若目标档已经存在时、在覆盖时会先询问动作的进行

-r 若给出的文件是一目录文件，递归复制该目录下所有的子目录和文件

例如：

用root身份，将root目录下的.bashrc复制到/tmpxia ,并命名为bashrc

Cp .bashrc /tmp/bashrc #复制到tmp目录下并且重命名bashrc

Cp -i ./bashrc /tmp/bashrc 会提示是否覆盖 n:不覆盖、y:覆盖

Rcp:指令用在远端复制文件或目录、使在两台Linux主机之间的文件复制操作更简单。通过适当的配置，在两台Linux主机之间复制文件而无需输入密码，就像本地文件复制一样简单。

## find命令

find命令用来在指定路径下查找指定文件。

find path-name 【-options】【-print –exec –ok命令{} \;】

具体选项说明：

path-name ：find命令查找的目录路径，例如：可以用”.”来表示当前目录，用”/”来表示系统目录

-options：find命令的这个选项主要用来控制搜索方式，列出一些常用的几种格式如下：

-name’字符串’：查找文件名匹配所给字符串的所有文件，

-type文件类型：按照指定文件类型搜索，文件类型包括b（块设备文件）、c（字符设备文件）、f（普通文件）、l（符号连接）、d（目录）、

P（管道）、s（socket）文件

-mtime +n或-n：按时间搜索，+n表示n天之前，-n表示今天到n天前之间的

-empty：查找大小为0的目录或文件

…………….（还有很多不一一写出来）

-print：将搜索结果输出到标准输出

-exec：对搜索出符合条件的文件执行所给出的LINUX命令，而不询问用户是否需要执行该命令。{}表示shell命令的选项即为所查找到的文件。命令末尾必须以”;”结束。注意{ }\，在}和\之间一定要有空格哦

-ok：对搜索出符合条件的文件执行所给出的LINUX命令，与-exec不同的是他会询问用户是否需要执行该命令

举例：

找出/file路径下的普通文件，并列出他们的完整路径

[root@192 file]# find /file -type f -exec ls -al {} \;

找出/file路径下文件名以cat开头的文件，并打印出来

[root@192 file]# find /file -name 'cat\_\*' -print

找出/file路径下文件名为’cat\_test’的文件，并通过交互的方式删除

[root@192 file]# find /file -name 'cat\_test' -ok rm {} \;

找出系统/var/log目录下修改时间在7天以前的普通文件，然后以交互的方式删除

[root@192 file]# find /var/log -type f -mtime +7 -OK rm {} \;

好了！！！不一一举例了，，，，大概就是这样的用法

## split命令

用来分割文档，将一个文件分成数个。

split 【选项】【input\_file】【output\_file】

命令选项：

-b size：指定分割出来的文件大小，size加入单位，b代表512字节、k代表1KB、m代表1MB

-n：指定分割的每个文件的长度、默认1000行

-d：将split命令生成的文件序列以数字形式命名

-a：指定split命令生成的文件序列的长度，默认长度2

Input\_file：需要分割的文件

output\_file：分割出来的文件序列，split命令分割后生成的文件序列名依次为output\_fileaa、output\_fileab、output\_fileac等。如果没有指定output\_file、系统默认以X作为输出文件名

举例：

分割cat\_test1文件

[root@192 file]# split cat\_test1🡪没有指定文件名则生成了文件xaa

将access\_log文件进行分割、设定每个分割文件大小为10M

[root@192 file]# split -b 10M access\_log access\_log\_bak🡪生成文件access\_log\_bakaa, access\_log\_bakab等

## mv命令

用于将文件或者目录改名，或者将文件由一个目录移入另外一个目录中

参数：

-f: force强制的意思，如果目标文件已经存在，不会询问而直接覆盖

-i:若目标文件已经存在，就会询问是否覆盖

比价简单不举例说明

# 压缩与解压

## zip/unzip命令

将一般的文件或者目录进行压缩或者解压，默认生成以”zip”为后缀的压缩包

zip【选项】压缩文件名

unzip【选项】压缩文件名

zip参数：

-r：递归压缩，将制定目录下的所有文件以及子目录全部压缩

-d：从压缩文件内删除指导文件

-x”文件列表”：压缩时排除文件列表中指定的文件

………..（不一一列举参数）

unzip参数：

-n：解压时不覆盖已存在的文件

-o：解压时覆盖已存在的文件，并且不要求用户确认

-d目录名：把压缩文件解压到指定目录下

1、将/etc目录下的所有文件以及目录进行压缩，设置压缩级别为最高，保存到文件/opt/etc.zip

zip -9r /opt/etc.zip /etc

2、将/var目录中所有文件以及子目录压缩，除了”\*.log”文件之外，将压缩文件保存到/opt目录下

zip –r /opt/var.zip /var –x “\*.log”

3、解压file.zip到/file1目录下，并且在解压过程中如果出现相同的文件就直接覆盖，不要求用户确认

[root@192 file1]# unzip -od /file1 file.zip

## gzip/gunzip命令

将一般文件进行压缩或者解压，压缩文件预设的扩展名为”gz”，其实gunzip就是gzip的硬链接，因此无论是压缩还是解压都可以通过gzip来实现

注意：gzip只能对文件进行压缩、不能压缩目录、即使指定压缩目录、也只能压缩目录内的所有文件

gzip【选项】压缩（解压缩）的文档名

参数说明：

-d：对压缩的文件进行解压

-r：递归压缩，指定目录以及子目录下的所有文档

-t：检查压缩文档的完整性

-v：对于每个压缩和解压的文档，显示相应的文件名和压缩比

-num：用指定的数字num配置压缩比例、”-1”或”-fast”表示最低压缩比、”-9”或者”-best”表示最高压缩比、系统默认压缩比为6

举例：

1、将etc.zip文件进行gzip压缩，设置gzip的压缩级别为9

gzip -9v etc.zip

2、将etc.zip.gz文件解压到当前目录

gzip -d etc.zip.gz

gunzip etc.zip.gz

## bzip2/bunzip2命令

对文件进行压缩和解压、类似gzip和gunzip命令，只能对文件进行压缩、压缩生成”.bz2”为后缀名的压缩包。bunzip2其实就是bzip2的符合链接，即软连接，因此压缩和解压都可以通过bzip2实现

参数：

-d：执行解压缩、此时选项后面跟要解压缩的文件

-v：压缩或者解压缩时，显示详情信息

-k：bzip2压缩或者解压缩后，会删除原文件，若要保留原文件，可以使用此选项

-f：bzip2压缩或者解压时，不会覆盖相同的文件，若要覆盖就用此选项

不要举例说明

## tar命令

tar命令是LINUX下经常使用的归档工具、对文件或者目录进行打包归档，归档成一个文件，但是并不进行压缩哦！

tar【主选项+辅助选项】文件或者目录

主选项参数一些：

-c：新建文件

-r：把要归档的文件追加到档案文件的末尾

-t：列出档案文件中已经归档的文件列表

-x：从打包的档案文件中还原出文件

-u：更新档案文件，用新建文件替换档案中的原始文件

辅助选项：

-z：调用gzip命令在文件打包过程中压缩/解压文件

-w：在还原文件时、把所有文件的修改时间设定为现在时间

-j：调用bzip2命令在文件打包的过程中压缩/解压文件

-Z：调用compress命令过滤档案

-f：选项后紧跟档案文件的存储设备，默认是磁盘、需要指定档案文件名

-v：指定在创建归档文件过程中，显示各个归档文件的名称

-p：在文件归档的过程中，保存文件的属性不发生变化

举例：

1、将/etc目录下的所有文件打包，并显示打包的详细文件，设置打包文件名为etc.tar同时保存在/opt目录下

tar -cvf /opt/etc.tar /etc

2、将/etc目录下的所有文件打包并压缩，然后显示打包的详细文件、设置打包文件名为etc.tar.gz，同时保存在/opt目录下。

tar -zcvf /opt/etc.tar.gz /etc

tar -jcvf /opt/etc.tar.bz2 /etc

“zcvf”选项表示在打包完成后调用gzip命令对档案文件进行压缩，压缩名为etc.tar.gz，命名上没有硬性规定哦

3、查看我们刚刚压缩的压缩包内容

tar -ztvf etc.tar.gz

注意：etc.tar.gz压缩包可能含有很多文件，无法在一个屏幕显示完毕，这时可以使用more命令，比如：tar -zxvf /etc.tar.gz | more

4、将/opt/etc.tar.gz解压到/usr/local/src下

先切换到/usr/local/src目录下，然后

tar -zxvf /opt/etc.tar.gz

一般情况下，可以把opt/etc.tar.gz在任何地方解压

5、将/etc目录下的所有文件打包备份到/opt目录下，并保存每个文件的权限

tar -zcvpf /opt/etc.tar.gz /etc

…………

## cpio命令

cpio命令是通过重定向的方式将文件进行打包、备份、还原、恢复的工具、他还可以解压”.cpio”或者”.tar”结尾的文件

格式：

cpio【选项】>文件名或者设备名

cpio【选项】<文件名或者设备名

说说参数：

-o：将文件复制、打包成文件或者将文件输出到设备上

-i：将打包文件解压或者将设备上的备份还原到系统中

-t：查看cpio打包的文件内容或者输出到设备上的文件内容

-v：显示打包过程中的文件名称

-d：在cpio还原文件的过程中，自动建立相应的目录

-c：一种较新的存储方式

-B：让默认块可以增大到5120字节，默认块为512字节，这样做的好处是可以加快存取速度

举例：

1、将/etc下的所有普通文件度备份到/opt/etc.cpio中

find /etc –type f | cpio –ocvB > /opt/etc.cpio

2、查看上面我们备份的etc.cpio文件

cpio -icdvt < etc.cpio

3、将备份还原到相应的位置，如果有相同文件就进副盖

cpio -icduv < /opt/etc.cpio

4、如果备份文件太多，一个屏幕无法显示完毕，此时可以利用下面的命令，让备份文件输出到文件

cpio -icdvt < /opt/etc.cpio > /tmp/st\_content

cpio无法直接读取文件，他需要每个文件或者目录的完整路径才能识别，而find命令的输出刚好做到了这一点，所以cpio一般和find命令配合使用

## sort命令

此命令用语排序，就像python列表排序一样

数字排序？

sort -n test2 #必须要加-n参数，因为默认情况下sort命令会把数据当做字符串来处理，加上-n参数就会告诉sort命令把数字识别成数字而不是字符串

时间戳排序？

-M参数

--reverse按照降序排列

参数过多………..

# 磁盘管理和维护

## df命令

df命令用来检查linux系统的磁盘空间占用情况

参数：

-h：以容易理解的格式输出文件系统分区占用情况，如32KB、120MB

-k：以KB大小单位输出文件系统分区占用情况

-m：以MB大小单位输出文件系统分区占用情况

-a：列出所有的文件系统分区，包含0大小的文件系统分区

-i：列出文件系统分区的inode信息

-T：显示磁盘分区文件系统类型

## du命令

du命令用来显示文件或者目录所占用的磁盘空间情况、指定某个文件或者目录查看其占用的空间

参数：

-s：显示文件或者整个目录的大小，单位是KB

-b：以字节为单位显示文件大小或者显示目录下所有文件的大小

-sh：以人性化的格式显示文件或者目录大小，如300MB、1.2GB等

-sm：以MB为单位显示文件或者目录大小

例如：

显示LINUX系统 /tmp目录的总大小，

du –sh /tmp

查看/file目录下的cat\_test文件大小

du -sh /file/cat\_test

## fsck命令

fsck命令用来检查文件系统并尝试修复错误

fsck【选项】【-t <文件系统类型>】【设备名】

## sync命令

用来强制把内存中的数据写回硬盘，以免数据的丢失。

在LINUX中，修改过的操作会缓存到内存中，不会马上写入硬盘，等待合适的时候再从内存写入硬盘，这样可以提高系统读写效率，LINUX默认每隔3秒系统自动执行一次sync操作，系统正常关机时也会执行sync操作，保证所有操作全部从内存写入磁盘。

## eject命令

用来退出抽取式设备，如光驱或者磁带

## mount/umount命令

挂载以及卸载指定的文件系统

这里不解释，，需要用的时候慢慢再百度哦。

# 网络设置与维护

## ifconfig命令

用来配置网络或者显示当前网络接口状态、类似windows下的ipconfig命令

配置ip地址，百度

## scp命令

用于将文件或者目录从一个LINUX系统复制到另一个LINUX系统下，scp传输数据用SSH协议，保证了数据传输的安全

格式？？

scp 远程用户@ip地址：文件的绝对路径本地LINUX系统路径（远程LINUX文件复制到本地）

scp 本地LINUX系统文件路径远程用户名@ip地址：远程系统文件绝对路径名（本地LINUX文件复制到远程LINUX）

例如：

目前我们的IP为”192.168.60.133”的LINUX系统下、计划将此系统下的/home/ixdba/etc.tar.gz文件复制到IP为”192.168.60.168”远程LINUX

root用户下的/tmp目录下

scp /home/ixdba/etc.tar.gz [root@192.168.60.168:/tmp](mailto:root@192.168.60.168:/tmp)

## netstat命令

用来显示本机网络连接，运行端口和路由表等信息

netstat【选项】

参数：

-a：显示本机所有连接和监听端口

-n：以网络ip地址的形式显示当前建立的有效连接和端口

-r：显示路由表信息

-t：显示有所TCP协议连接情况

-u：显示所有UDP协议连接情况

## man命令

man命令用来访问存储在LINUX系统上的手册页面，（bash）

man xxx命令

例如：

查看cp命令手册man cp (q)退出

## ps命令

查看进程命令；

参数过多：

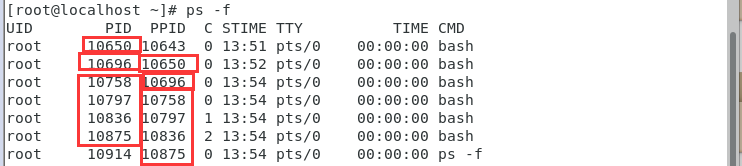
-a 显示所有进程

-f 详细显示程序执行的路径群

-e 将除内核进程以外所有进程的信息写到标准输出

查看shell子类父类关系，ps –forest

ps –f也可以通过PPID显示出谁是谁的父进程



exit 退出子shell

exit也可以退出父类shell，退出终端

## top命令

类似上面的ps命令，但是ps命令只能显示某个特定时间点的信息，

而top命令实时显示进程信息所以就比ps安逸啊。

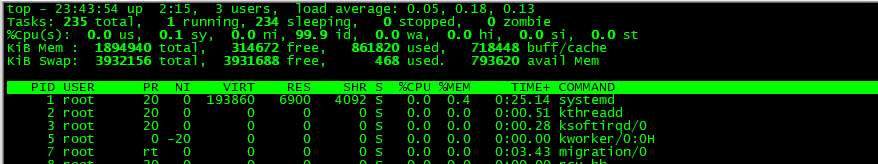
参数：过多

-i ：不显示闲置或者僵尸的进程信息

-c : 显示进程的整个命令路径，而不只是显示命令名称

……….太多

基本信息？



PID：进程的ID

USER：进程属主的名字

PR：进程的优先级

NI：进程的谦让度值

VIRT：进程占用的虚拟内存总量

RES：进程占用的物理内存总量

SHR：进程和其他进程共享的内存总量

S：进程状态（D代表可中断的休眠状态，R代表在运行状态，S代表休眠状态，T代表跟踪状态或停止状态，Z代表僵死状态）

%CPU：进程使用的CPU时间比例

%MEM：进程使用的内存占可用内存的比例

TIME+：自进程启动到目前为止的CPU时间总量

COMMAND：进程所对应的命令名称，也就是启动的程序名

那如何结束进程呢？？？杀死进程

## kill和killall命令

kill通过进程ID（PID）给进程发信号，结束进程

kill 5005 结束5005号进程

-s 参数强制终止

killall支持通过进程名而不是进程ID来结束进程

## 进程列表

可以在一行中指定要依次运行的一系列命令，并用分号（;）隔开

[root@localhost ~]# pwd;ls;cd /file;pwd #不是在子shell中运行的

如果是要将依次运行变成进程列表，需要加个括号

[root@localhost ~]# (pwd;ls;cd /file;pwd) 看起来效果并没有什么不同

进程列表会创建子shell来执行这些命令

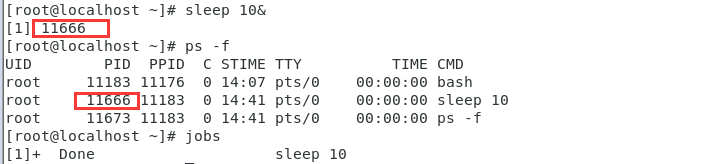
sleep 睡眠

sleep 10睡眠10秒

末尾加上&，命令置入后台模式

sleep 10& 植入后台

可以用ps –f来查看一番，也可以用jobs查看后台作业



## history命令

可以查看最近使用过的命令列表

# 环境变量

## env或者printenv命令

查看全局变量命令（很多）

要查看个别环境变量的值，用printenv命令，不要用env

例如：

printenv HOME

printenv MAIL

………………..

也可以使用echo显示变量的值，这种情况下引用某个环境变量的时候，必须在变量前面加上一个美元符号（$）

echo $HOME

echo $MAIL

…………………

在echo命令中在变量前面加上$不仅仅是要显示当前的值，它能够让变量作为命令行参数

ls $HOME

ls $MAIL

………………..

## 局部变量

LINUX没有特定的命令来只显示局部变量

set命令会显示为某个特定进程设置的所有环节变量，包括局部变量，全局变量以及用户定义变量

## 设置局部用户定义变量

通过等号（=）给环境变量赋值，值可以是数字或者字符串

my\_variable=Hello （变量名和等号，值之间没有空格）

echo $my\_variable

Hello

如果是含有空格的字符串，必须要加引号

My\_varibale=”Hello world”

设置了局部变量后可以在hell进程中任何一个地方使用，但是如果新生成另外一个shell它在子shell是不可以用的

bash 新生成一个子shell

echo $my\_variable （这个时候就不可用了）

exit 退出子shell

echo $my\_variable （这个时间就可以用了）

## 设置全局环境变量

设置全局变量的方法：先创建一个局部环境变量，然后再把它导出到全局环境中，这个过程通过export命令来完成，变量名前面不需要加$

my\_variable=’I am global now’

export my\_variable 导入全局环境中

echo $my\_variable (I am global now)

bash 生成一个子shell

echo $my\_variable (I am global now)

my\_variable=”Null” 修改子shell中的全局变量my\_variable的值

echo $my\_variable (Null)

export my\_variable 子shell重定义并导出环境变量,也不会影响父类的环境变量

exit 退出子shell

echo $my\_variable (I an global now)修改子shell中全局变量的值不会影响到父shell中该变量的值

## 删除环境变量

unset命令完成删除环境变量（引用环境变量时不加$）

echo $my\_variable (hello)

unset my\_variable

echo $my\_variable (无)已删除环境变量

且子shell中删除环境变量不会影响父类shell的环境变量

## 默认的shell环境变量

系统环境变量

（太多，太多）

## 设置PATH环境变量

PATH环境变量定义了用于进行命令和程序查找的目录

echo $PATH

/usr/lib64/qt-3.3/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin

PATH中的目录用冒号（:）隔开

如果命令或者程序的位置没有包括在PATH变量中，那么如果不使用绝对路径的话就会产生错误

把新的搜索目录加入到现在的PATH变量中

PATH=$PATH:/home/file 添加目录/home/file到环境变量

这种方式只能持续到退出或重启系统

## 定位系统环境变量

接上如何让环境变量的作用持久化？？？？

bash shell几种不同的启动方式

### 登录shell

当你登录LINUX系统时，bash shell会作为登录shell启动。

读取命令：

1、/etc/profile

2、$HOME/.bash\_profile

3、$HOME/.bashrc

4、$HOME/.bash\_login

5、$HOME/.profile

/etc/profile文件是系统上默认的bash shell的主启动文件，系统上的每个用户登录时都会执行这个启动文件

另外4个启动文件是针对用户的，可根据个人需要定制

1、/etc/profile文件

cat /etc/profile 查看一下

2、$HOME目录下的启动文件

该目录下，提供一个用户专属的启动文件来定义该用户所用到的环境变量。

以点号开头说明是隐藏文件哦

我们可以