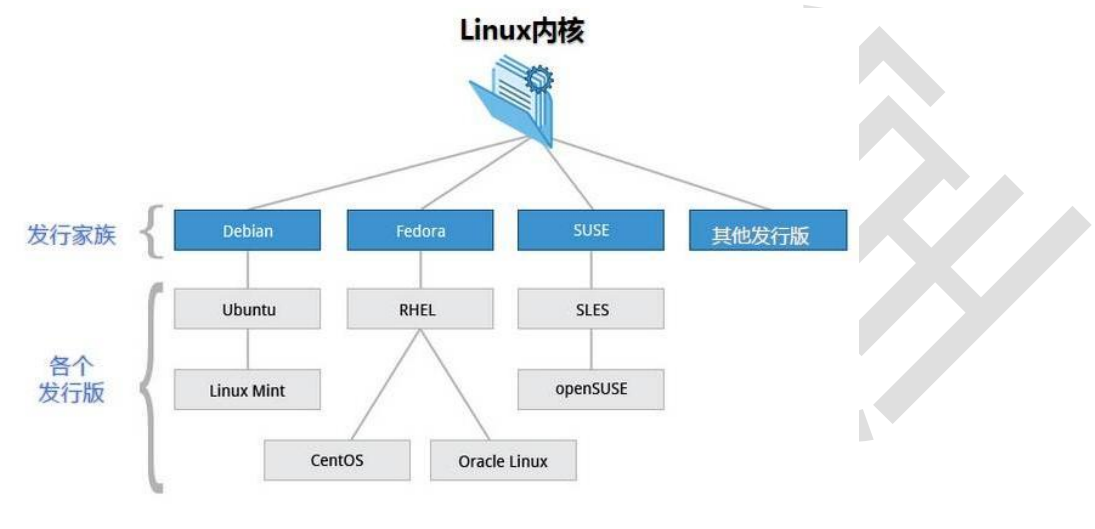


# 一、基础介绍

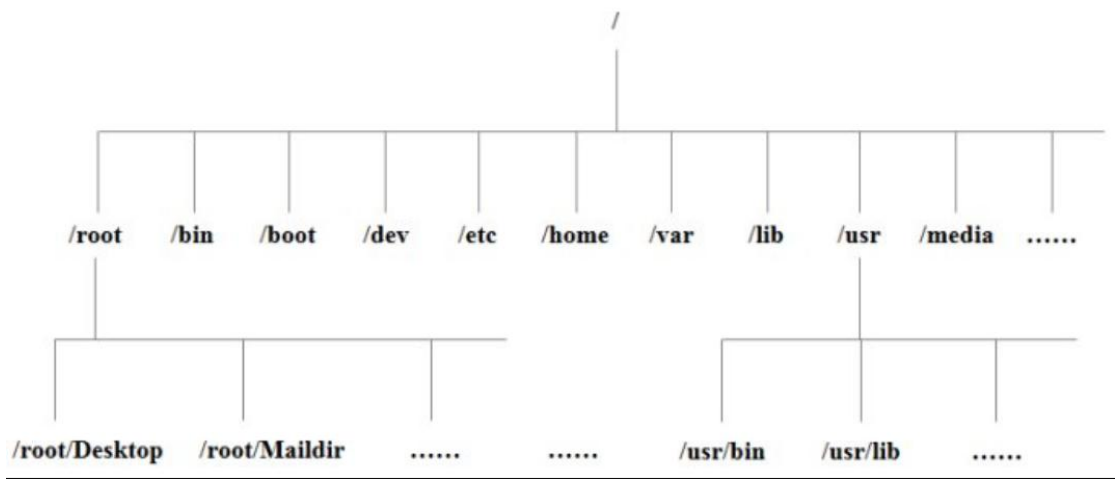
Linux 是一套免费使用和自由传播的类 Unix 操作系统，是一个基于 POSIX 和 UNIX 的多用户、多任务、支持多线程和多 CPU 的操作系统。

## 1、发行的 linux 版本

Linux 的发行版说简单就是将 Linux 内核与应用软件做一个打包。市面上较知名的发行版有：Ubuntu、RedHat、CentOS、Debian、Fedora、SuSE、OpenSUSE、Arch Linux、SolusOS



## 2、linux 系统目录结构



系统启动必须：

- /boot: 存放的启动 Linux 时使用的内核文件，包括连接文件以及镜像文件。
- /etc: 存放所有的系统需要的配置文件和子目录列表，更改目录下的文件可能会导致系统不能启动。
- /lib: 存放基本代码库（比如 c++库），其作用类似于 Windows 里的 DLL 文件。几乎所有的应用程序都需要用到这些共享库。

- **/sys:** 这是 linux2.6 内核的一个很大的变化。该目录下安装了 2.6 内核中新出现的一个文件系统 sysfs。sysfs 文件系统集成了下面 3 种文件系统的信息：针对进程信息的 proc 文件系统、针对设备的 devfs 文件系统以及针对伪终端的 devpts 文件系统。该文件系统是内核设备树的一个直观反映。当一个内核对象被创建的时候，对应的文件和目录也在内核对象子系统中

#### 指令集合：

- **/bin:** 存放着最常用的程序和指令
- **/sbin:** 只有系统管理员能使用的程序和指令。

#### 外部文件管理：

- **/dev :** Device(设备)的缩写, 存放的是 Linux 的外部设备。**注意:** 在 Linux 中访问设备和访问文件的方式是相同的。
- **/media:** 类 windows 的**其他设备**, 例如 U 盘、光驱等等, 识别后 linux 会把设备放到这个目录下。
- **/mnt:** 临时挂载别的文件系统的, 我们可以将光驱挂载在/mnt/上, 然后进入该目录就可以查看光驱里的内容了。

#### 临时文件：

- **/run:** 是一个临时文件系统, 存储系统启动以来的信息。当系统重启时, 这个目录下的文件应该被删掉或清除。如果你的系统上有 /var/run 目录, 应该让它指向 run。
- **/lost+found:** 一般情况下为空的, 系统非法关机后, 这里就存放一些文件。
- **/tmp:** 这个目录是用来存放一些临时文件的。

#### 账户：

- **/root:** 系统管理员的用户主目录。
- **/home:** 用户的主目录, 以用户的账号命名的。
- **/usr:** 用户的很多应用程序和文件都放在这个目录下, 类似于 windows 下的 program files 目录。
- **/usr/bin:** 系统用户使用的应用程序与指令。
- **/usr/sbin:** 超级用户使用的比较高级的管理程序和系统守护程序。
- **/usr/src:** 内核源代码默认的放置目录。

#### 运行过程中要用：

- **/var:** 存放经常修改的数据, 比如程序运行的日志文件 (/var/log 目录下)。
- **/proc:** 管理**内存空间!** 虚拟的目录, 是系统内存的映射, 我们可以直接访问这个目录来, 获取系统信息。这个目录的内容不在硬盘上而是在内存里, 我们也可以直接修改里面的某些文件来做修改。

#### 扩展用的：

- **/opt:** 默认是空的, 我们安装额外软件可以放在这个里面。
- **/srv:** 存放服务启动后需要提取的数据 (**不用服务器就是空**)

## 3、linux 基础命令

shutdown 会给系统计划一个时间关机。它可以被用于停止、关机、重启机器。shutdown 会给系统计划一个时间关机。它可以被用于停止、关机、重启机器。

```
# shutdown -P now ### 关闭机器

# shutdown -H now ### 停止机器

# shutdown -r09:35 ### 在 09:35am 重启机器
```

要取消即将进行的关机, 只要输入下面的命令:

```
# shutdown -c
```

halt 命令通知硬件来停止所有的 CPU 功能, 但是仍然保持通电。你可以用它使系统处于低层维护状态。注意在有些情况它会完全关闭系统。

```
# halt          ### 停止机器

# halt -p       ### 关闭机器

# halt --reboot  ### 重启机器
```

poweroff 会发送一个 ACPI 信号来通知系统关机。

```
# poweroff      ### 关闭机器

# poweroff --halt  ### 停止机器

# poweroff --reboot  ### 重启机器
```

reboot 命令 reboot 通知系统重启。

```
# reboot        ### 重启机器

# reboot --halt  ### 停止机器

# reboot -p      ### 关闭机器
```

## 二、Linux 用户/用户组

Linux 系统是一个多用户多任务的分时操作系统，任何一个要使用系统资源的用户，都必须首先向系统管理员申请一个账号，然后以这个账号的身份进入系统。

用户账号的管理工作主要涉及到用户账号的添加、修改和删除。

### 1、增加一个用户组

groupadd 选项 用户组

可以使用的选项有：

-g GID 指定新用户组的组标识号（GID）。

-o 一般与-g 选项同时使用，表示新用户组的 GID 可以与系统已有用户组的 GID 相同。

例 1：

```
# groupadd group1          ##此命令向系统中增加了一个新组 group1
```

例 2：

```
# groupadd -g 1003 group2  ##此命令向系统中增加了一个新组 group2，同时指定新组的组标识号是 1003
```

### 2、删除一个用户组

groupdel 用户组

例：

```
# groupdel group1          ##此命令从系统中删除组 group1。
```

## 3、修改用户组的属性

**groupmod** 选项 用户名

常用的选项有：

-g GID 为用户组指定新的组标识号。

-o 与 -g 选项同时使用，用户组的新 GID 可以与系统已有用户组的 GID 相同。

-n 新用户组 将用户组的名字改为新名字

例 1：

```
# groupmod -g 10000 -n group3 group2 ##此命令将组 group2 的标识号改为 10000，组名修改为 group3。
```

## 4、添加用户账号

**useradd** 选项 用户名

参数说明：

选项：

-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。

用户名：

指定新账号的登录名。

例 1

```
# useradd -d /usr/peter -m peter ##创建用户 peter，其中-d和-m选项用来为登录名 peter 产生一个主目录/usr/peter。
```

例 2

```
# useradd -s /bin/sh -g group1 -G group2,root peter2 ##新建用户 peter2，登录 Shell 是 /bin/sh，属于 group1，又属于 group2 和 root。
```

这里可能新建组：#groupadd group1 及 groupadd group2

增加用户账号就是在/etc/passwd 文件中为新用户增加一条记录，同时更新其他系统文件如/etc/shadow，/etc/group 等。

## 5、删除帐号

**userdel** 选项 用户名

常用的选项是 -r，它的作用是把用户的主目录一起删除。

例如：

```
# userdel -r peter2
```

##删除用户 **peter2** (主要是/etc/passwd, /etc/shadow, /etc/group 等) 的记录, 同时删除用户的主目录。

## 6、修改帐号

**usermod** 选项 用户名

常用的选项包括 **-c**, **-d**, **-m**, **-g**, **-G**, **-s**, **-u** 以及 **-o** 等, 这些选项的意义与 **useradd** 命令中的选项一样, 可以为用户指定新的资源值。

例如:

```
# usermod -s /bin/sh -d /home/p -g group2 peter ##将用户 peter 的登录 Shell 修改为 sh, 主目录改为/home/p, 用户组改为 group2。
```

## 7、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使用, 即使是指定空口令。

**passwd** 选项 用户名

可使用的选项:

**-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。

如果默认用户名, 则修改当前用户的口令。例如, 假设当前用户是 **peter**, 则下面的命令修改该用户自己的口令:

```
$ passwd
```

```
Old password:*****
```

```
New password:*****
```

```
Re-enter new password:*****
```

如果是超级用户, 可以用下列形式指定任何用户的口令:

```
# passwd peter
```

```
New password:*****
```

```
Re-enter new password:*****
```

为用户指定空口令时, 执行下列形式的命令:

```
# passwd -d peter
```

**passwd** 命令还可以用 **-l(lock)** 选项锁定某一用户, 使其不能登录, 例如:

```
# passwd -l peter
```

## 8、切换用户组

```
$ newgrp root
```

这条命令将当前用户切换到 root 用户组，前提是 root 用户组确实是该用户的主组或附加组。类似于用户账号的管理，用户组的管理也可以通过集成的系统管理工具来完成。

## 9、与用户账号有关的系统文件

与用户和用户组相关的信息都存放在一些系统文件中，这些文件包括/etc/passwd、/etc/shadow、/etc/group 等

### 9.1、/etc/passwd 文件

Linux 系统中的每个用户都在/etc/passwd 文件中有一个对应的记录行，它记录了这个用户的一些基本属性。

例子：

```
# cat /etc/passwd

root:x:0:0:Superuser:/:
daemon:x:1:1:System daemons:/etc:
bin:x:2:2:Owner of system commands:/bin:
sys:x:3:3:Owner of system files:/usr/sys:
adm:x:4:4:System accounting:/usr/adm:
uucp:x:5:5:UUCP administrator:/usr/lib/uucp:
auth:x:7:21:Authentication administrator:/tcdb/files/auth:
cron:x:9:16:Cron daemon:/usr/spool/cron:
listen:x:37:4:Network daemon:/usr/net/nls:
lp:x:71:18:Printer administrator:/usr/spool/lp:
```

每一行记录对应着一个用户（其中 bin/sys/adm/uucp/lp/nobody 是伪用户），每行记录又被冒号(:)分隔为 7 个字段，其格式和具体含义如下：

用户名:口令:用户标识号:组标识号:注释性描述:主目录:登录 Shell

### 9.2、/etc/group 文件

用户组的所有信息都存放在/etc/group 文件中。

例子：

```
root::0:root
bin::2:root,bin
sys::3:root,uucp
adm::4:root,adm
daemon::5:root,daemon
lp::7:root,lp
users::20:root,sam
```

此文件的格式类似于/etc/passwd 文件，由冒号(:)隔开若干个字段，这些字段有：

组名:口令:组标识号:组内用户列表

# 三、 文件与目录

## 1、文件基本属性

### 1.1、文件的属性概貌。

文件 类型	属主 权限			属组 权限			其他用户 权限		
0	1	2	3	4	5	6	7	8	9
d	rwx			r-x			r-x		
目录 文件	读	写	执行	读	写	执行	读	写	执行

第一个字符：代表这个文件的类型，是目录、文件，还是一个链接等等

- [ d ] 目录
- [ - ] 文件
- [ l ] 链接文档(link file)
- [ b ] 可供储存的接口设备(可随机存取装置)
- [ c ] 串行端口设备，例如键盘、鼠标(一次性读取装置)

接下来的字符：以三个一组分成三组，用 r、w、x 三个参数的组合表示，位置不会改变

- [ r ] 代表可读(read)
- [ w ] 代表可写(write)
- [ x ] 代表可执行(execute)
- [ - ] 没有权限

### 1.2、文件的属主与属组

每一个文件，它都有一个特定的所有者，也就是对该文件具有所有权的用户。

同时，在 Linux 系统中，用户是按组分类的，一个用户属于一个或多个组。

文件所有者以外的用户又可以分为文件所有者的同组用户和其他用户。

因此，Linux 系统按文件所有者、文件所有者同组用户和其他用户来规定了不同的文件访问权限。

#### 1.2.1、chgrp：更改文件属组

语法：

chgrp [-R] 属组名 文件名

参数选项

-R：递归更改文件属组，就是在更改某个目录文件的属组时，如果加上-R的参数，那么该目录下的所有文件的属组都会更改。

## 1.2.2、chown：更改文件属主，也可以同时更改文件属组

语法：

`chown [-R] 属主名 文件名`

`chown [-R] 属主名:属组名 文件名`

进入 /root 目录 (^) 将 install.log 的拥有者改为 bin 这个账号：

例：

`chown root:root aa`      ##更改 aa 的属主与属组

## 1.2.3、chmod：更改文件 9 个属性

先复习一下刚刚上面提到的数据：文件的权限字符为：『**-rwxrwxrwx**』，这九个权限是三个三个一组的！其中，我们可以使用数字来代表各个权限，各权限的分数对

照表如下：

r:4   w:2   x:1

每种身份(owner/group/others)各自的三个权限(r/w/x)分数是需要累加的，例如当权限为： [-rwxrwx---] 分数则是：

owner = rwx = 4+2+1 = 7

group = rwx = 4+2+1 = 7

others= --- = 0+0+0 = 0

所以等一下我们设定权限的变更时，该文件的权限数字就是 770 啦！

变更权限的指令 chmod 的语法是这样的：

`chmod [-R] xyz 文件或目录`

选项与参数：

xyz：就是刚刚提到的数字类型的权限属性，为 rwx 属性数值的相加。

-R：进行递归(recursive)的持续变更，亦即连同次目录下的所有文件都会变更

举例来说，如果要将 bashrc 这个文件所有的权限都设定启用，那么命令如下：

`chmod +x aa`      ##更改 aa 的属性,加 x 权限

`chmod 777 aa`      ##更改 aa 的属性，三组权限都设置成 7

## 2、文件与目录管理命令

使用 `man [命令]` 来查看各个命令的使用文档，如：`man cp`

`ls`：列出目录 ---ll

`cd`：切换目录



```
pwd: 显示目前的目录

mkdir: 创建一个新的目录

rmdir: 删除一个空的目录

cp: 复制文件或目录  -----scp 网络复制

rm: 移除文件或目录

mv: 移动文件与目录, 或修改文件与目录的名称
```

## 3、硬链接与软链接

### 3.1 硬连接

硬连接指通过索引节点来进行连接。在 Linux 的文件系统中，保存在磁盘分区中的文件不管是什么类型都给它分配一个编号，称为索引节点号(Inode Index)。在 Linux 中，**多个文件名指向同一索引节点是存在的**。比如：A 是 B 的硬链接（A 和 B 都是文件名），则 A 的目录项中的 inode 节点号与 B 的目录项中的 inode 节点号相同，即一个 inode 节点对应两个不同的文件名，两个文件名指向同一个文件，A 和 B 对文件系统来说是完全平等的。删除其中任何一个都不会影响另外一个的访问。

硬连接的作用是允许一个文件拥有多个有效路径名，这样用户就可以建立硬连接到重要文件，以防止“误删”的功能。其原因如上所述，因为对应该目录的索引节点有一个以上的连接。只删除一个连接并不影响索引节点本身和它的连接，只有当最后一个连接被删除后，文件的数据块及目录的连接才会被释放。也就是说，**文件真正删除的条件是与之相关的所有硬连接文件均被删除**。

### 3.2 软连接

另外一种连接称之为符号连接（Symbolic Link），也叫软连接。**软链接文件有类似于 Windows 的快捷方式**。它实际上是一个特殊的文件。在符号连接中，文件实际上是一个文本文件，其中包含的有另一文件的位置信息。比如：A 是 B 的软链接（A 和 B 都是文件名），A 的目录项中的 inode 节点号与 B 的目录项中的 inode 节点号不相同，A 和 B 指向的是两个不同的 inode，继而指向两块不同的数据块。但是 A 的数据块中存放的只是 B 的路径名（可以根据这个找到 B 的目录项）。A 和 B 之间是“主从”关系，如果 B 被删除了，A 仍然存在（因为两个是不同的文件），但指向的是一个无效的链接。

例：

```
$touch a          #创建一个测试文件 a

$ ln a b          #创建 a 的一个硬连接文件 b    --echo 123 > a ##a,b 的内容同时变为 123

$ ln -s a c       #创建 a 的一个符号连接文件 c

$ ls -li         # -i 参数显示文件的 inode 节点信息
```

## 四、 安装程序 rpm/yum

### 1. rpm 方式

需要去下载相应的 rpm 格式的安装包 xxxx.rpm

```
rpm [参数] rpm 包
```

-i 安装

-U 更新（很少用）

-e 卸载

-v 显示安装信息

-h 显示安装进度

例:

```
rpm -ivh jdk-8u172-linux-x64.rpm 安装 jdk8 并显示安装进度和安装信息
```

```
rpm -qa|grep jdk 查看 jdk 的安装包
```

```
rpm -e jdk1.8-1.8.0_221-fcs.x86_64 卸载 jdk8 , 只适用于 rpm 包安装的软件
```

### 2. yum 方式

yum ( Yellow dog Updater, Modified) 是一个在 Fedora 和 RedHat 以及 SUSE 中的 Shell 前端软件包管理器。

基於 RPM 包管理, 能够从指定的服务器自动下载 RPM 包并且安装, 可以自动处理依赖性关系, 并且一次安装所有依赖的软件包, 无须繁琐地一次次下载、安装。

yum 提供了查找、安装、删除某一个、一组甚至全部软件包的命令, 而且命令简洁而又好记。

#### 2.1、yum 语法

```
yum [options] [command] [package ...]
```

**options:** 可选, 选项包括-h(帮助), -y(当安装过程提示选择全部为"yes"), -q(不显示安装的过程)等等。

**command:** 要进行的操作。

**package** 操作的对象。

例:

```
yum search jdk 搜索 jdk 安装包
```

```
yum install -y java-1.8.0-openjdk.x86_64 安装 openjdk, -y 表示安装过程中的询问自动选 y
```

```
yum list installed |grep jdk 列出安装的 jdk 软件包
```

```
yum remove java-1.8.0-openjdk.x86_64 java-1.8.0-openjdk-headless.x86_64 卸载 jdk
```

#### 2.2、yum 常用命令

1. 列出所有可更新的软件清单命令: `yum check-update`

2. 更新所有软件命令: `yum update`

```
3.仅安装指定的软件命令: yum install <package_name>
4.仅更新指定的软件命令: yum update <package_name>
5.列出所有可安装的软件清单命令: yum list
6.删除软件包命令: yum remove <package_name>
7.查找软件包 命令: yum search <keyword>
8.清除缓存命令:
yum clean packages: 清除缓存目录下的软件包
yum clean headers: 清除缓存目录下的 headers
yum clean oldheaders: 清除缓存目录下旧的 headers
yum clean, yum clean all (= yum clean packages; yum clean oldheaders) :清除缓存目录下的软件包及旧的 headers
```

## 2.3、yum 源

官方的 yum 源在国内访问效果不佳。需要改为国内比较好的阿里云或者网易的 yum 源

在/etc/yum..repos.d/下进行如下操作(请做好相应备份):

```
wget http://mirrors.163.com/.help/CentOS7-Base-163.repo
mv CentOS7-Base-163.repo CentOS-Base.repo
```

重建缓存:

```
yum clean all
yum makecache
```

## 3. 配置环境变量

在 Linux 系统下, 安装应用程序后, 在安装目录之外, 可能仍然会出现 “command not found”。要解决所有地方都能使用应用程序, 就涉及到环境变量 path 的设置问题

### 3.1、配置环境变量

系统环境变量文件在/etc 下的 profile 文件, 我们可以用 vi profile 命令来编辑该文件, 将变量添加进去

一般需要将安装文件的 bin 目录加入 path 中, 可在 profile 中加入下面一行代码, 如:

```
vim /etc/profile
# 末尾追加以下内容
export JAVA_HOME=/usr/java/default
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

## 3.2、环境变量生效

```
source /etc/profile
```

# 五、 Shell 脚本

shell 语言是一门 linux 系统下的工具语言，主要用于写一些 linux 系统下的操作命令，实际上 Shell 是一个命令解释器，它解释由用户输入的命令并且把它们送到内核。或者直接理解为 shell 命令是可以执行多个 linux 命令的脚本。Shell 种类繁多，有以下种类（一般使用的 Bash，就是 Bourne Again Shell，它是大多数 Linux 系统默认的 Shell）：

- Bourne Shell (/usr/bin/sh 或 /bin/sh)
- Bourne Again Shell (/bin/bash)
- C Shell (/usr/bin/csh)
- K Shell (/usr/bin/ksh)
- Shell for Root (/sbin/sh)

shell 并不复杂，有编程基础的话，简单入门两三个小时就可以入门，主要是把一些重复操作的 linux 命令写成 shell 脚本来执行一下。以下列出 shell 常规的一些要素：

## 1、解释器与执行 shell

解释器

```
#!/bin/sh    Bourne shell 版本
#!/bin/bash  Bourne Again Shell 版本
```

执行 shell

```
chmod +x ./test.sh #使脚本具有执行权限
./test.sh #执行脚本 “./”表示当前目录下
```

## 2、演示

```
#!/bin/bash    ##声明 bash 脚本

##demo        ##注释
echo $PATH     ##打印 PATH 环境变量
name="Peter"   ##定义变量 name
echo $name     ##打印 name 值
echo "I am ${name}'s friend"    ##字符串拼接
echo ""

name2="I am ${name}'s good friend"    ##字符串拼接
```

```

echo $name2
echo ""

names=("Peter" "james" "deer")          ##定义数组
echo ${names[@]}                        ##遍历数组
echo "I am ${names[1]}'s friend" ##第二个元素
echo "I have ${#names[@]} friends"      ##数组长度
echo ""

for var in ${names[@]};                  ##循环数组
do
    if test $var = 'Peter'                ##字符串相等
    then
        echo "I am Peter"
    else
        echo "I am ${var}'s friend"
    fi
done

echo ""

if [ $(ps -ef | grep -c "ssh") -gt 1 ]; ##查找是否有 ssh 服务
then echo "ssh service open";
fi

echo ""

echo "sh arg: $0 $1"                    ##sh 的传参

重定向:
1、test 'aa' -eq "bb" > out    ##命令输出到 out 文件，报错信息并不会进入 out
2、test 'aa' -eq "bb" > out 2>&1 ##将 stderr 合并到 stdout，则报错信息进入了 out

```

## 六、常用命令

### 1. 最最常用的是 cd 命令

```

cd 进入用户主目录;
cd ~ 进入用户主目录;
cd - 返回进入此目录之前所在的目录;
cd .. 返回上级目录（若当前目录为"/"，则执行完后还在"/"；".."为上级目录的意思）；
cd ../../ 返回上两级目录；

```

---

ls, ll, wget, curl, history

## 2. 新建文件夹和文件：mkdir touch

📄

**mkdir** 创建文件夹

**mkdir** **dirname** 直接跟文件夹名，可在当前目录下创建文件夹

**mkdir** /opt/lamp/**dirname** 可跟路径

**mkdir** -p /opt/lamp/**dirname** 假如 lam 不存在，需要用 -p 才可以创建该文件夹

**touch** 新建文件

**touch** **dilenam** 可直接跟文件名在当前目录下创建新的文件

## 3. cat/less/more/tail 等文件查看命令

📄

-n 或 -number: 有 1 开始对所有输出的行数编号;

-b 或 --number-nonblank: 和 -n 相似，只不过对于空白行不编号;

-s 或 --squeeze-blank: 当遇到有连续两行以上的空白行，就代换为一行的空白行;

-A: 显示不可打印字符，行尾显示“\$”;

-e: 等价于“-vE”选项;

-t: 等价于“-vT”选项; 📄

-e: 文件内容显示完毕后，自动退出;

-f: 强制显示文件;

-g: 不加亮显示搜索到的所有关键词，仅显示当前显示的关键字，以提高显示速度;

-l: 搜索时忽略大小写的差异;

-N: 每一行行首显示行号;

-s: 将连续多个空行压缩成一行显示;

-S: 在单行显示较长的内容，而不换行显示;

-x<数字>: 将 TAB 字符显示为指定个数的空格字符。

📄

-<数字>: 指定每屏显示的行数;

-d: 显示“[press space to continue, 'q' to quit.]”和“[Press 'h' for instructions]”;

-c: 不进行滚屏操作。每次刷新这个屏幕;

-s: 将多个空行压缩成一行显示;

-u: 禁止下划线; +<数字>: 从指定数字的行开始显示。

按 Space 键: 显示文本的下一屏内容。

按 Enter 键: 只显示文本的下一行内容。

按斜线符|: 接着输入一个模式，可以在文本中寻找下一个相匹配的模式。

按 h 键: 显示帮助屏，该屏上有相关的帮助信息。

按 B 键：显示上一屏内容。

按 Q 键：退出 rmore 命令。

--retry: 即是在 tail 命令启动时, 文件不可访问或者文件稍后变得不可访问, 都始终尝试打开文件。使用此选项时需要与选项“--follow=name”连用;

-c 或 --bytes=: 输出文件尾部的 N (N 为整数) 个字节内容;

-f 或; --follow: 显示文件最新追加的内容。“name”表示以文件名的方式监视文件的变化。“-f”与“-fdescriptor”等效;

-F: 与选项“-follow=name”和“--retry”连用时功能相同;

-n 或 --line=: 输出文件的尾部 N (N 位数字) 行内容。

--pid=<进程号>: 与“-f”选项连用, 当指定的进程号的进程终止后, 自动退出 tail 命令;

-q 或 --quiet 或 --silent: 当有多个文件参数时, 不输出各个文件名;

-s<秒数>或 --sleep-interval=<秒数>: 与“-f”选项连用, 指定监视文件变化时间间隔的秒数;

-v 或 --verbose: 当有多个文件参数时, 总是输出各个文件名;

一般 tail 命令最常用的 -n 和 -f, 例:

```
tail filename 读取 filename 最后 10 行内容
```

```
tail -f filename 实时动态读取 filename 最后 10 行内容
```

```
tail -20f filename 实时动态读取 filename 最后 20 行内容
```

cat 和 less 及 more 指令相似, cat 是一次性读取所有内容, 文件内容较多时速度较慢。less 与 more 的区别在于: less 可以前后翻页查看, more 只能向前翻页查看。

tail 则用于实时获取 log 信息, 从后向前读取内容

上述四种命令均可跟 grep 搭配使用 cat/less/tail |grep mysql

## 4. 查看文件大小的命令 du/df

```
du -sh * 显示当前目录下所有文件的大小
```

```
du -sh filename 显示该文件大小
```

```
du -sh 显示当前目录所占空间大小
```

-s 或 仅显示总计, 只列出最后加总的值。

-h 或 以 K, M, G 为单位, 提高信息的可读性。

```
df 显示磁盘占用信息
```

直接 df 默认 -k 为单位

```
df -lh 显示本地系统的占用信息, 以 K, M, G 为单位
```

## 5. 大重点--文本编辑器 vi

vi 命令是 UNIX 操作系统和类 UNIX 操作系统中最通用的全屏幕纯文本编辑器。

Linux 中的 vi 编辑器叫 vim, 它是 vi 的增强版 (vi Improved), 与 vi 编辑器完全兼容, 而且实现了很多增强功能。

vi 编辑器支持编辑模式和命令模式, 编辑模式下可以完成文本的编辑功能, 命令模式下可以完成对文件的操作命令, 要正确使用 vi 编辑器就必须熟练掌握着两种模式的切换。

默认情况下, 打开 vi 编辑器后自动进入命令模式。从编辑模式切换到命令模式使用 “esc” 键, 从命令模式切换到编辑模式使用 “A”、“a”、“O”、“o”、“I”、“i” 键。

vi 编辑器提供了丰富的内置命令, 有些内置命令使用键盘组合键即可完成, 有些内置命令则需要以冒号 “:” 开头输入。常用内置命令如下:

## 1 Ctrl+u: 向文件首翻半屏;

```
2 Ctrl+d: 向文件尾翻半屏;
3 Ctrl+f: 向文件尾翻一屏;
4 Ctrl+b: 向文件首翻一屏;
5 Esc: 从编辑模式切换到命令模式;
6 ZZ: 命令模式下保存当前文件所做的修改后退出 vi;
7 :行号: 光标跳转到指定行的行首;
8 :$: 光标跳转到最后一行的行首;
9 x 或 X: 删除一个字符, x 删除光标后的, 而 X 删除光标前的;
10 D: 删除从当前光标到光标所在行尾的全部字符;
11 dd: 删除光标行正行内容;
12 ndd: 删除当前行及其后 n-1 行;
13 nyy: 将当前行及其下 n 行的内容保存到寄存器?中, 其中? 为一个字母, n 为一个数字;
14 p: 粘贴文本操作, 用于将缓存区的内容粘贴到当前光标所在位置的下方;
15 P: 粘贴文本操作, 用于将缓存区的内容粘贴到当前光标所在位置的上方;
16 /字符串: 文本查找操作, 用于从当前光标所在位置开始向文件尾部查找指定字符串的内容, 查找的字符串会被加亮显示;
17 ? name: 文本查找操作, 用于从当前光标所在位置开始向文件头部查找指定字符串的内容, 查找的字符串会被加亮显示;
18 a, bs/F/T: 替换文本操作, 用于在第 a 行到第 b 行之间, 将 F 字符串换成 T 字符串。其中, "s/"表示进行替换操作;
19 a: 在当前字符后添加文本;
20 A: 在行末添加文本;
21 i: 在当前字符前插入文本;
22 I: 在行首插入文本;
23 o: 在当前行后面插入一空行;
24 O: 在当前行前面插入一空行;
25 :wq: 在命令模式下, 执行存盘退出操作;
26 :w: 在命令模式下, 执行存盘操作;
27 :w!: 在命令模式下, 执行强制存盘操作;
28 :q: 在命令模式下, 执行退出 vi 操作;
29 :q!: 在命令模式下, 执行强制退出 vi 操作;
30 :e 文件名: 在命令模式下, 打开并编辑指定名称的文件;
31 :n: 在命令模式下, 如果同时打开多个文件, 则继续编辑下一个文件;
32 :f: 在命令模式下, 用于显示当前的文件名、光标所在行的行号以及显示比例;
33 :set nu: 在命令模式下, 用于在最左端显示行号;
34 :set nonu: 在命令模式下, 用于在最左端不显示行号;
35 :1,3y 复制第一行到第三行
36 :1,3d 删除第一行到第三行
37 :1,3s/str/str_new/g 替换第一行到第三行中的字符串
38 :1,3s/str/str_new 替换第一行到第三行中的字符串第一个字符
39 :1,3 g/str /d 删除第一行到第三行中含有这个字符串的行
```



---

## 6. 重定向 >

### 输出重定向

```
who > bbb.txt 将 aaa 的内容写入 bbb 中，覆盖写入
cat bbb.txt >> ccc.txt 讲 aaa 的内容追加写入 bbb 中，不覆盖原来内容
> bbb.txt 将 bbb 清空
```

### 输入重定向

```
grep 05:37:43.730 < web.2019-07-22.0.log
```

大多数 UNIX 系统命令从你的终端接受输入并将所产生的输出发送回 到您的终端。

一个命令通常从一个叫标准输入的地方读取输入，默认情况下，这恰好是你的终端。

同样，一个命令通常将其输出写入到标准输出，默认情况下，这也是你的终端。

### 重定向深入讲解

一般情况下，每个 Unix/Linux 命令运行时都会打开三个文件：

- 标准输入文件(stdin)：stdin 的文件描述符为 0，Unix 程序默认从 stdin 读取数据。
- 标准输出文件(stdout)：stdout 的文件描述符为 1，Unix 程序默认向 stdout 输出数据。
- 标准错误文件(stderr)：stderr 的文件描述符为 2，Unix 程序会向 stderr 流中写入错误信息。

默认情况下，command > file 将 stdout 重定向到 file，command < file 将 stdin 重定向到 file。

## 7. 查看服务

netstat -nlpt|grep 80 查看该端口号是否被占用

free -m //查看 LINUX 内存剩余容量

ps 可以查看具体的进程信息，一般与管道符连接其他命令使用，如：grep

ps 常用参数-ef/-aux，一般最常用还是-ef，例：ps -ef|grep mysql 查询 mysql 进程

top 也可查看进程信息，而且是动态显示

whoami 查看当前登陆用户

who 查看多少用户在使用系统

date 查看系统时间，可跟时间格式使用

cal 查看日历，可跟年份，查看指定的年份

```
chkconfig --list #查看系统服务启动
```

```
chkconfig iptables on #开机启动该服务
```

```
chkconfig iptables off #开机不启动该服务
```

```
service iptables start #启动该服务
```

```
service iptables restart #重启该服务
```

```
ps -ef|grep mysql|grep -v grep|awk '{print $2}'ps -ef|grep mysql 是查询 mysql 服务的进程
```

|后的 grep -v grep 是匹配不包含 grep 的行

awk 是取查询结果的第几列, awk '{print \$2}'则是取第二列的值

grep 无参数则显示匹配的行

-c 显示匹配的行数

-v 显示不匹配的行

## 8. 杀掉进程 kill 命令

kill 最常用的参数是-9, 用法: kill -9 进程号 即可强制杀掉该进程

### 统计命令 wc

常用的参数是 -l 用法: wc -l , 例:

```
ps -ef|grep mysql|wc -l 统计查询出的 mysql 进程的行数
```

查找命令 find/locate/whereis/which

find -name 后跟文件名, 可查看文件所在目录, 可跟 user, 查看属于 user 的文件

find -name filename 查找 filename 所在目录

find -name name\* 查找开头为 name 的文件所在目录

find -name \*name 查找结尾为 name 的文件所在目录

find -name \*name\* 查找包含 name 字符串的文件所在目录

find -user faith 查看用户 faith 的文件

locate 用法与 find 基本相似, 只是 locate 搜索速度较快些, locate 一般系统不会自带, 需要安装, 可用 yum 安装

whereis 只能搜索程序名

which 则是只查询 path 中的环境变量

## 9. 压缩和解压命令 gzip/gunzip zip/unzip tar

gzip 和 gunzip 一般可用参数是-r, 例:

```
gzip test.txt 压缩文件
```

```
gzip -r test 压缩所有 test 下的子文件
```

```
gunzip test.gz 解压文件
```

zip 和 unzip 可用参数较多, 例:

```
zip test 不跟参数直接使用
```

```
zip -r test 递归压缩 test 下所有文件
```

```
unzip test 不跟参数直接使用
```

```
unzip -n 解压时不覆盖已存在的文件
```

```
unzip -o 解压时覆盖已存在的文件
```

```
unzip -d 将文件解压到目录中去
```

tar 使用的较为多些, 用法也多, 最常用的是 zxcvf 几个参数, 例:

```
-c 创建新文档, 就是代表压缩的意思
```

```
-x 解压文档
```

```
-f 使用归档文件
```

```
-z 使用 gzip 解压
```

```
-v 详细输出模式
```

最为常用的使用方法:

```
tar -zcvf test.tar test 将 test 压缩为 test.tar 并输出详细信息
tar -zxvf test.tar 将 test.tar 解压缩，并输出详细信息
```

## 10. 定时任务 crontab

```
crontab [-u user] 文件
```

```
crontab [-u user] {-r -e -l}
```

不加-u 的话默认当前用户

-e: 执行文字编辑器来设定日程表，内定的文字编辑器是 vi。

-r: 删除目前的日程表。

-l: 列出目前的日程表。

crontab -e 就可以打开一个文件进行编辑

crontab 文件的格式为“M H D m d cmd”，M 为分钟 1-59，H 为小时 1-24，D 为天 1-31，m 为月 1-12，d 为周 0-6（0 为周日）。cmd 代表要执行的程序，\*代表每分钟都执行

\* \* \* \* \* sh /opt/lampp/test.sh 表示每分钟执行一次 test.sh 这个脚本

\* /5 \* \* \* \* sh /opt/lampp/test.sh 表示每 5 分钟执行一次 test.sh 这个脚本

30 21 \* \* \* /usr/local/apache/bin/apachectl restart 表示每晚的 21:30 重启 apache