# 安装篇：

主机名的命名必须以域名的形式出现，如jimmy.com，即xx.xx，否则可能导致一些错误。

Linux是实现了Unix内核所有功能并且免费的系统

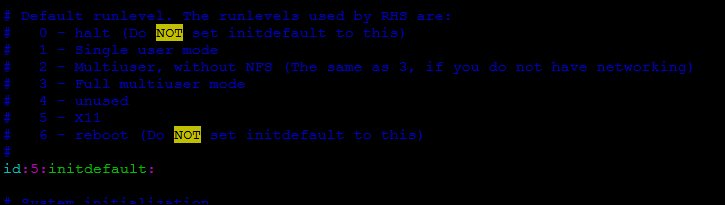
Info一般不推荐使用，过于详细。使用man已经足够

在使用man时如果发生乱码的情况，可以在shell中输入下面的命令解决问题：

|  |
| --- |
| LANG=C |

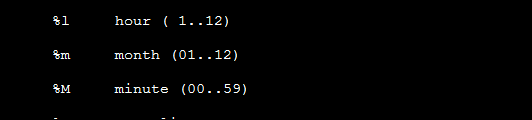
Ctrl+alt+F1..F6可以切换到不同的终端，之后可以ctrl+alt+F7返回原来的桌面。

Linux开机后默认进入的是图形界面，其实这个是在/etc/inittab这个文件里配置的。我们可以通过修改这个文件来配置它的默认开机后进入的界面：



因为默认设置为5，所以进入的是X11（即是图形界面）

支持多个终端有一个好处：当你在图形界面或者其中一个终端的某个进程出现问题并导致整个图面卡死时，我们就可以切换到其它的终端来把出问题的进程来杀掉



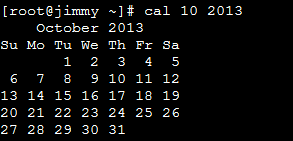
看MAN手册的时候，如看到上面的这个的“%M”它是表示你输入的命令中如果带”%M”就按提示输出，一定要记得要加入%而不是什么都不加，直接上M，或者是-M。如下面的输入是这样的：



对于空格在终端中的输入，可以使用上面的方式来转义，也可以使用字符串的方式：



关于时间的命令还有cal：

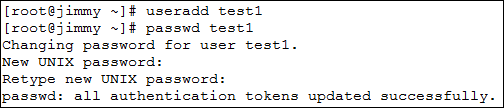


# 常用命令：

ls 一般与 -l 一起使用

touch 用于创建文件

创建用户：



创建用户和设置密码在这里是分开的两步。

因为当前用户是root，所以如果su test1，这个时候是不需要输入test1的密码的。但如果是从test1 su到root，这个时候就需要了

sync:将内存中的数据写到硬盘中

# 网络初步操作

查看SSH服务的状态可以使用下面两种方式：

/etc/init.d/sshd status

或者：

service sshd status

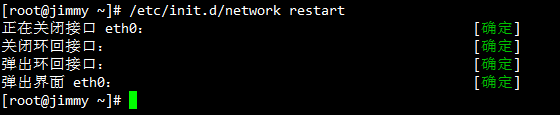
修改这个文件可以使用命令来配置网络的信息：

|  |
| --- |
| [root@jimmy ~]# vim /etc/sysconfig/network-scripts/ifcfg-eth0 |

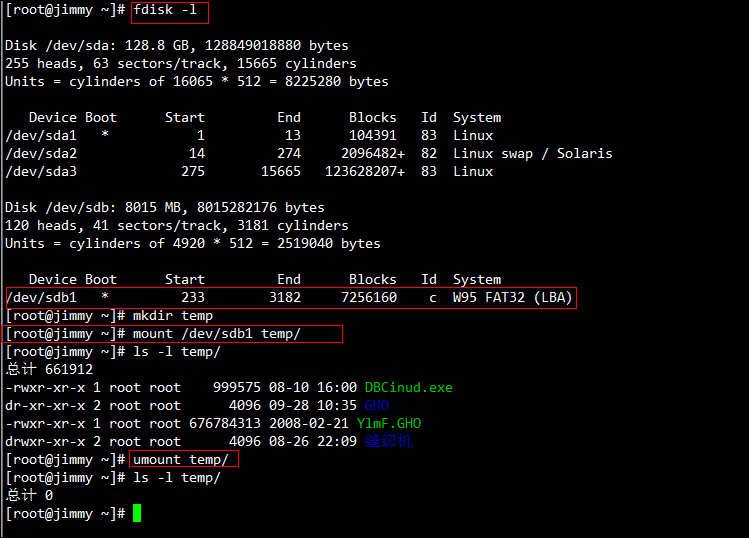
但DNS服务器的配置并没有在这个文件中，而是在：

|  |
| --- |
| [root@jimmy ~]# vim /etc/resolv.conf |

修改完后记得要重启服务：



# 挂载U盘的方法：



首先使用fdisk -l命令来查看硬盘的状态并找到插入的U盘，然后创建一个目录，再使用mount命令将U盘挂载到刚才创建的目录里。

卸载的时候则使用umount命令（注意不是u**n**mount）

cd - : 表示返回上一次进入的目录（来回切换，类似于遥控器的上一节目的功能）

# 目录文件和权限操作

文件信息详解：

drwxr-xr-x 2 root root 4096 09-28 09:11 bin

rmdir:删除文件夹

unalias:可以删除已经起的别名

tac: 与cat命令的区别是将文件内容倒序输出（tac本身就是cat的倒序）

/var/log/secure : 此文件可以记录用户登录的一些信息

tail -f：可以用来追踪日志，如：

tail -f /var/log/secure

RWX的意义：

* 作用到文件：
* r 可以读
* w 可以修改，但是不代码能删除文件
* x 可以被系统执行
* 作用到目录
* r 可以读 ls查看目录内的文件
* w 可以在目录内创建文件 删除文件 对目录重命名
* x **可以进入这个目录** cd

要想删除文件，用户必须对当前**目录**有w权限

对于Linux的权限有一个地方需要注意，无论你的权限怎么设置，对于root用户都是透明的，即是root什么都能做。

rm -rf folder

要删除这个folder目录，首先必须要有cd的权限，即权限x，因为rm -rf的意思是先进入这个目录删除里面的文件之后再删除当前目录。

# 硬盘的一些概念：

* block: 记录文件的具体内容，每一个block只能存储一个文件的信息。比如一个block的大小为4K，而a.txt大小为1K，那么剩余的3K就浪费了，不可以被其它文件来使用。
* inode：记录属性和权限，还记录了该文件对应的block的编号(索引)。每一个文件单独占用一个inode，每一个inode也只能记录一个文件的信息。每一个inode的大小为128byte
* super block: 记录文件系统的信息，如使用的是什么文件系统。和整个文件系统中inode和block的使用情况：如使用和剩余的数量，大小等

比如执行命令cat a.txt，OS会先去找到对应的inode来查询是操作者对文件的权限，权限匹配的话再根据inode里面的信息定位到对应的block，从而找到相对应的文件内容

可以使用下面的命令来查看分区的inode， block，卷标名等的信息：

|  |
| --- |
| [root@jimmy ~]# **dumpe2fs** /dev/sda1 |

分析下面的命令OS的查找过程：

|  |
| --- |
| [root@jimmy ~]# cat /etc/hosts |

首先要获取/的inode号，然后再找到/所在的block，再找到根下面的etc文件的inode号，再找到etc的block，然后再查找到etc下面的hosts的inode号，最后根据这个号查找到hosts的block，并把文件内容读出来

由于分区表的大小是固定为64byte，所以只能分4个分区（主分区）



sda描述的是这一块硬盘，sda1~sda3描述的是这个硬盘的三个分区

将每一个分区隔开的叫做柱面

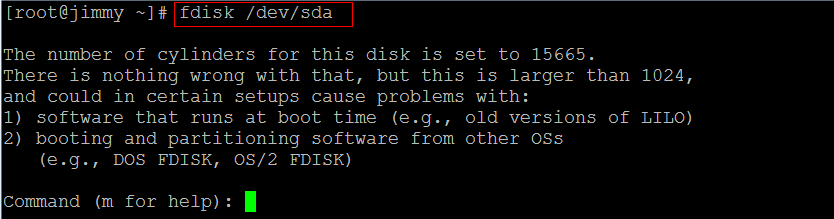
扩展(Extend)分区下面可以有多个逻辑分区。扩展分区不可以直接格式化或使用，只可以对

逻辑分区进行这些操作

# 分区

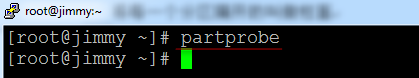
主分区+扩展分区最多在一块硬盘上只允许有4个，每一个硬盘最多有一个扩展分区

使用fdisk并将磁盘作为参数可以对磁盘进行新建分区，删除分区等操作



接着再按提示操作就可以了。

最后一定要记得使用以下命令来刷新分区表：



分区建好后就可以格式化了：



这个命令等同于下面的命令：



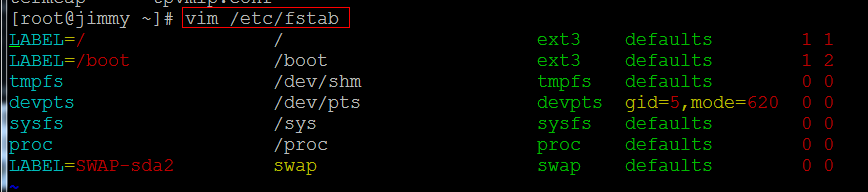
然后挂载：



卸载



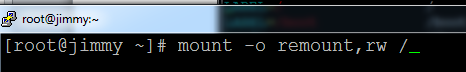
通过修改下面的文件可以使得系统在启动的时候自动的挂载分区



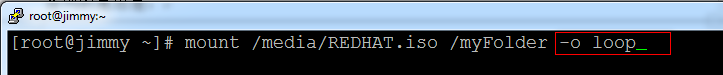
各列的意思是：

1. 卷标
2. 挂载点
3. 文件系统的格式
4. 一些系统参数，相当于mount中的-o，一般default就可以了
5. 表示能否被dump命令所使用，dump时，是否对当前的分区起作用
6. 表示是否使用fsck来检查当前的分区，如系统不正常关闭，下次启动时有可能会要求进行fsck的操作。（0表示不检查，1表示由用户来选择是否检查，2必须检查）根目录一般设置为1

根分区是不能被卸载的，但有时分区配置文件存在错误，如/etc/fstab文件里面有配置错误，那么我们就可能需要重新挂载根分区



挂载ISO文件时通常需要加一些参数



# 内存交换空间的问题

1. 安装Linux时，忘记配置swap

新建一个swap分区就可以了，fdisk /dev/sda

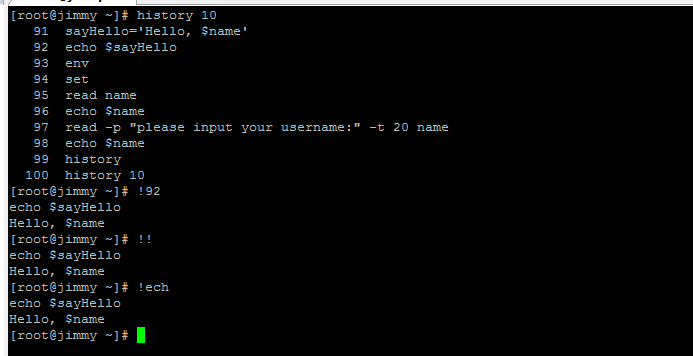
1. Swap不足，需要增加空间

# 磁盘配额

# Shell

## Shell命令历史记录

记录于~/.bash\_history，默认为1000条记录，可以在环境变量中设置



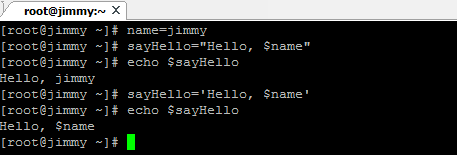
History后面跟的数字n表示显示history中记录的最后n条记录

!92： 表示执行第92条命令

!! ： 表示执行上一次的命令（相当于向上的箭头，在scripts中会很有用）

!ech ： 表示执行上一次以ech开头的命令

## 双引号和单引号的区别



单引号保留原始字符

双引号则继续下探

## 定义全局变量

其实就是定义环境变量

在Redhat中启动顺序是：

ect/profile

ect/bashrc

~/.bashrc

~/.bashr\_profile

如果使用切换用户命令，如: su jimmy

则只有ect/bashrc文件会被读取。

但如果使用: su **–** jimmy

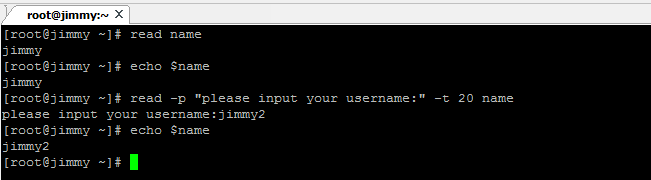
则**etc/profile**和ect/bashrc会被按顺序读取

所以如果想定义全局的系统变量，应当把变量定义在ect/bashrc中，如果只想定义针对某些用户才用的，可将其定义在profile中

## 打印环境变量

* 使用env可以打印当前系统的环境变量（使用export分布过的）
* 使用set可以打印当前系统的环境变量和在shell中定义过的变量(没有用export发布过的，如a=1，这样的变量也会输出)

## 键盘IO

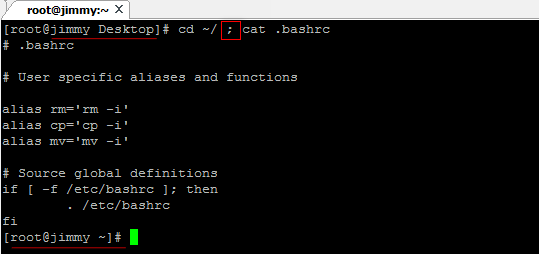


-p :可以在等待输入的位置前附加一些字符提示信息  
-t : 表示对当前输入的等待时间，如果超过20s，就会停止

## 特殊通配符

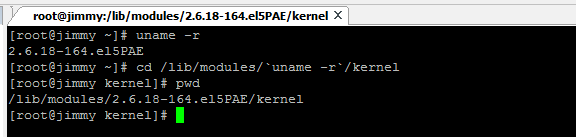
### ;(分号)：

先执行分号前面的命令，紧接着执行后面的命令



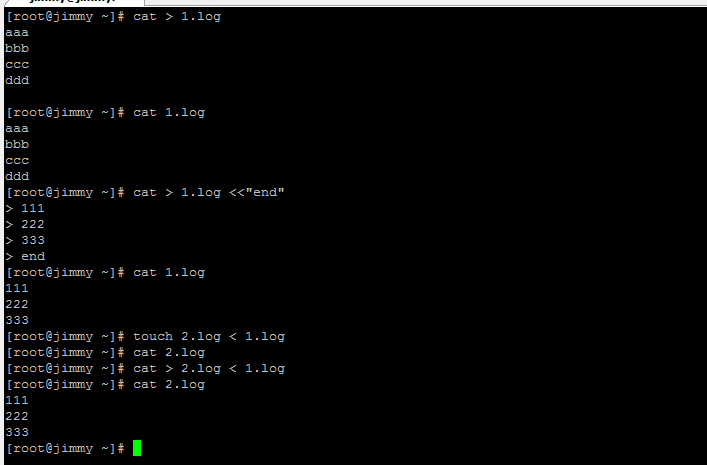
### ·（点号数字1前面的键，不是句号）

先执行点号内的命令然后将命令返回的值附加到外部命令中再执行



### >, >>, <, <<（数据流重定向）

>>表示在原有文件中添加，>表示覆盖原有文件，<和<<相似



* cat > 1.log : 等待从键盘中输入字符到1.log文件中（输入ctrl+c来结束输入）
* cat > 1.log <<”end” ：等待从键盘中输入字符到1.log中，并且当输入end时表示输入结束
* 注意后面从一个文件的内容输入新建文件的输入的例子

### 标准输入/输出，标准错误输出

标准输入 stdin 0

标准输出 stdout 1

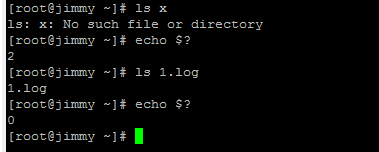
标准错误输出 stderr 2



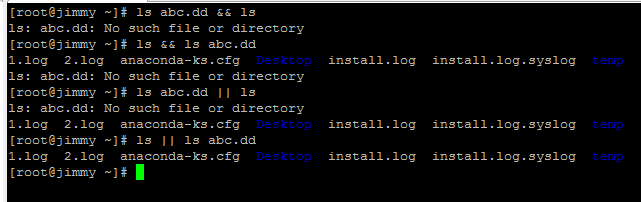
如果不想保留错误信息，可以将它输出到/dev/null中

## 命令回传码

每一条命令执行后都会返回一个命令回传码，如果为非0，则表示命令没有正确结束



* &&：cmd1 && cmd2，如果cmd1命令回传码为0（成功）才执行cmd2，如果为非0（不成功），则不再执行cmd2
* ||：cmd1 || cmd2，如果cmd1命令回传码为0（成功），则cmd2不执行，如果为非0（不成功），cmd2才会被执行



## 管道(pipe)

### find命令

find dir

-name 指定文件名

-size 指定大小

-type 类型 f文件 d目录

-o 或

-a 与

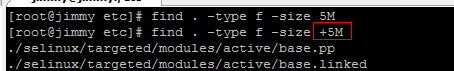
-not 非

-user root 搜索属于root用户的文件

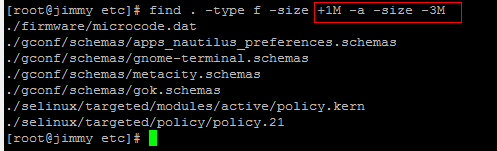
-group groupA指定用户组

-perm 777 指定权限

查询大小为5M和大于5M的文件

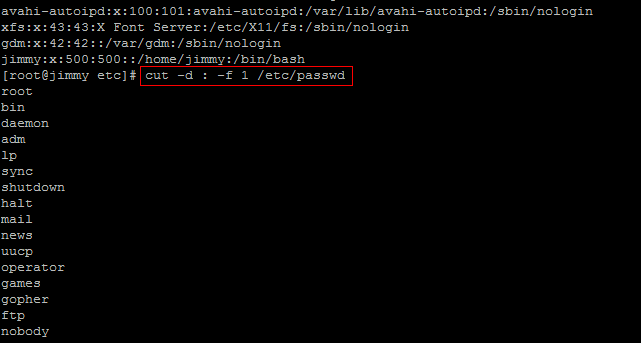


查询大小大于1M并且小于3M的文件



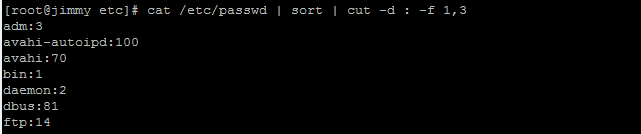
### cut命令（相当于java中的StringTockenizer）

cut –d’分割符’ –f 1 文件名



cut –d’分割符’ –f 1,3 文件名（表示取第1个和第3个，第2个不会输出，不是从第1个到第3个）

### sort命令



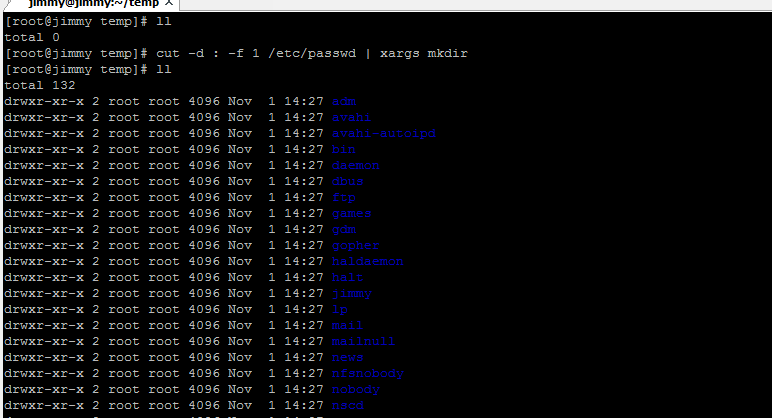
### wc命令

print the number of newlines, words, and bytes in files

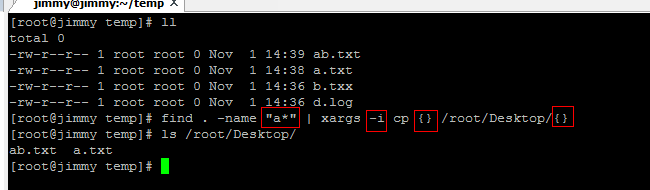
### uniq命令（重复数据只显示一个）

### xargs

使用管道前的输出结果作为管道后的命令的输入参数



注意其格式是管道后端是先到xargs才到待接收参数的命令mkdir



按名字匹配时，匹配字符串需要以引号引住

由于需要告诉cp命令复制什么东西以什么命令复制到什么目录下，所以使用{}来占位，表示以管道前的输入来作为占位符的内容，且这个情况下需要加入参数-i

### grep

查找出每一行符合查询条件的内容

-i ： 忽略大小写

-n :　输出行号

-v ：反向选择（即输出不包含查询字符串的内容）

-r : 同时也递归搜索每一个文件夹内的文件内容

# FAQ:

1. 安装后无法ping通主机。

默认是NAT的方式，选择桥接的方式就可以了。

