

Table of Contents

1. Linux 基础命令课程.....	3
2. Shell.....	3
3. Linux 的登陆.....	4
4. Linux 的登陆续.....	5
5. 关机和注销.....	6
6. 显示当前工作目录中的内容.....	7
7. 显示隐藏的文件.....	8
8. 显示其他目录中的文件.....	9
9. 建立目录.....	10
10. 删除目录.....	11
11. 切换目录.....	12
12. 复制文件.....	13
13. 移动文件.....	14
14. 创建文件.....	15
15. 删除文件.....	16
16. 通配符.....	17
17. pwd.....	19
18. 增加用户.....	19
19. 搜索文件.....	20
20. 显示文件内容.....	21
20.1 cat.....	21
20.2 more.....	22
20.3 less.....	23
20.4 head 和 tail.....	24
21. 基础.....	24
21.1 基础一.....	24
21.2 基础二.....	26
21.3 基础三.....	27
21.4 基础四.....	29
21.5 基础五.....	30
21.6 基础六.....	31
21.7 基础七.....	31
21.8 基础八.....	32
21.9 基础九.....	33
21.10 基础十.....	34
21.11 基础结束.....	36
22. 命令学习.....	36
22.1 命令学习一.....	36
22.2 命令学习二.....	37
22.3 命令学习三.....	38
22.4 命令学习四.....	39

22.5 命令学习五.....	40
22.6 命令学习六.....	41
22.7 命令学习七.....	41
22.8 命令学习八.....	42
22.9 先停.....	42
23. vi 编辑器.....	43
23.1 vi 编辑器学习一.....	43
23.2 vi 编辑器学习二.....	44
23.3 vi 编辑器学习三.....	45
23.4 vi 编辑器学习结束.....	46
24. 回到命令学习.....	47
24.1 回到命令学习一.....	47
24.2 回到命令学习二.....	47
24.3 回到命令学习三.....	49
24.4 回到命令学习四.....	49
24.5 回到命令学习五.....	50
24.6 回到命令学习六.....	51
24.7 回到命令学习七.....	52
24.8 回到命令学习八.....	53
24.9 回到命令学习九.....	53
24.10 结束.....	54

1. Linux 基础命令课程

豪华版主张学到更多的 Linux 命令知识。

从这里开始学习 Linux 命令，本课程让你更清楚地了解和掌握它，在 Linux 中命令是讲究大小写的，所以请管好你的 Caps Lock 键。

不用担心，Linux 命令说到底是几个字母的东西，你能学好它的，好，现在你找一个叫系统终端的东西，输入 cal 就行，出来的是什麼，是当前月份的月历，就从这里开始吧。很简单的吧，再输入 cal 2008，看到了吧，找出奥运会举办的日子。这只是非常简单的，你能学会的，shell 是 Linux 四个主要部分之一，其实上 Shell 是一个命令解释器。

红联 Linux 论坛致力于 Linux 技术讨论的站点。

红联 Linux 门户：www.linux110.com

红联 Linux 论坛：www.linux110.com/bbs

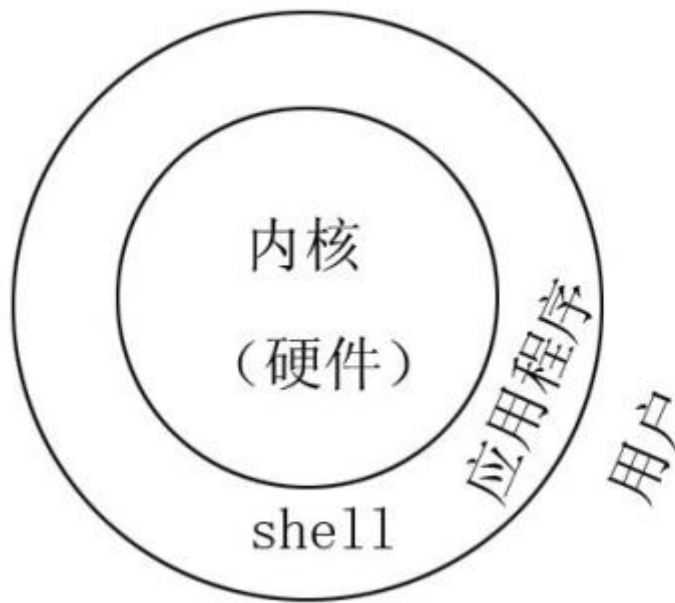
Linux 电子书籍推荐下载：<http://www.linux286.com/linux/linuxdzsj.htm>

制作：红 联
祝您阅读愉快！

2. Shell

在学习 Linux 的过程中，经常会听到 Shell 这个词。可能大家在使用 Windows 的时候有这种感觉，我要听歌，我打开一个播放器，那么音箱就可以出声音，要打印一片文档，我们只要在程序中点击一下打印按钮，打印就会自动的把文档打印出来。但是我们并没有自己手动的控制音箱和打印机等硬件设备，是因为系统已经为我们作好了。其实我们可以把 Windows 叫做 Shell，我们不用知道打印机的原理，也不需要自己控制它，我们只需要轻轻的点击一下鼠标，系统为我们来控制这些硬件设备。

看以下图，就知道 Shell 所起到的作用：



当用户使用系统，其实在和 Shell 在打交道，我们的 Windows 已经把各种各样的功能用图形表示出来了，当用户发出指令，其实先将这些指令发送给 Shell，然后由 Shell 将用户的指令翻译后传送给内核，再由内核来控制硬件的工作。然后内核将硬件的工作情况发送个 Shell，最后有 Shell 返回硬件的工作信息和情况。

这样一解释，大家应该明白其实 Shell，就是一个工作环境，就像 Windows 系统一样，如果没有这个系统，那么 Office 软件、Photoshop 软件，将无法工作，整个电脑就是一台废铁。也可以说我们在使用电脑的过程就是在和 Shell 打交道，是一个用户界面。

3. Linux 的登陆

我们来看看图中的每行所表示的是什么意思：

```
Red Hat Enterprise Linux Server release 5 (Tikanga)
Kernel 2.6.18-8.el5 on an i686

Linux login: _
```

第一行显示的是 Linux 发行商和所发布的版本，我们可以看到是 Red Hat Enterprise Server Linux 5，接着后面是该版本在该公司的内部代号。

第二行显示的是该版本的 Linux 所使用的 Linux 内核，表示内核版本是 2.6.18，其中的 2 是主版本号，6 是次要版本号，如果这个为偶数的话，就是一个稳定版本，所以在以后选择 Linux 的时候，一定要注意这个参数。最后的 18 为修订次数。这里要注意的是次要版本号如果为奇数的话，那么就是一个开发的版本号，并不是一个稳定的版本，并不适合作为服务器来使用。

第三行显示的是我们这台电脑的名字，这里的 Linux 就是我们这台电脑的名称。

4. Linux 的登陆续

明白以上的一些参数之后，我们就可以登录了，首先在第三行的“Linux login:”后面输入用户名，这里我们输入 root，然后按 Enter，接着要求输入用户密码，输入密码后，如果密码正确，则会出现如下画面：

```
Red Hat Enterprise Linux Server release 5 (Tikanga)
Kernel 2.6.18-8.el5 on an i686

Linux login: root
Password:
Last login: Wed Mar 19 05:25:21 from 192.168.8.88
[root@Linux ~]# _
```

现在可能大家对图中的所有信息应该都比较了解，可能就是最下面两行不是很明白，我们来分别看看：

Last login:这行表示我们这个用户的上次登录的时间和位置。

[root@linux ~]#这行的 root:表示登录的是 root 用户

@:相当 at

Linux:表示所登录的服务器,一般为服务器名称

~:表示目前位于/root 目录下

#:表示 root 的提示符号,普通用户为\$

这里可能大家对上面的一些信息还不是很了解，这里也不需要大家马上对所有的知识完全了解，这里有部分知识我们在以后的学习中才能彻底的明白。

到这里我们的用户就完成了登录，相信大家对这个登录应该是马上就能学会。

5. 关机 and 注销

先看看注销，我们可以使用 logout 可以退出系统，这个和 Windows 中的注销是同等效果。用于用户保存自己的资料。

例如[root@Linux ~]# logout

当我们输入这个命令按下 Enter 之后，我们就又回到我们的登录界面了。

在来看看关机，关闭系统使用 Shutdown 命令，确保用户和系统的资料完整。只有 root 用户才能使用这个命令。一般的用户是不允许执行这个命令的。

我们先看看 shutdown 语法：

`shutdown [options] when [message]`

options: `-r` 表示重启, `-h` 表示系统服务停滞(halt)后, 立刻关机, `-f` 表示快速重启

when: 为 `shutdown` 指定时间。`hh:mm`: 绝对时间, `hh` 指小时, `mm` 指分钟; 如 08:30, `+m:m` 分钟后执行, `now=+0`, 也就是立刻执行

message: 表示系统的广播信息, 一般提示各个用户系统关机或重启, 要求用户保存资料后退出。

我们来看看几个例子:

`shutdown -h now` 立刻关机

`shutdown -h 21:30` 今天 21: 30 关机

`shutdown -h +10` 十分钟后关机

`shutdown -r now` 立刻重启

`shutdown -r +10 'the system will reboot'` 10 分钟后重启, 管理员提示用户系统要重启了, 便于用户保存工作中的资料。只有 `root` 用户才能使用这个命令。

我们 `shutdown -h now`, 看看是不是立刻关机。

6. 显示当前工作目录中的内容

要想知道当前的工作目录中有些什么文件或目录, 在 Windows 的“命令提示符”中要使用“`dir`”命令, 那么在 Linux 中就要使用“`ls`”命令, 如下所示:

```
[root@Linux ~]# ls
```

```
anaconda-ks.cfg Desktop install.log install.log.syslog
```

这样屏幕上就立刻显示当前目录下的所有文件及自目录。注意你可能发现“Desktop”显示的颜色是蓝颜色, 这里提示大家一下:

白色的文件: 一般的文件

蓝色的文件: 目录

绿色的文件: 可执行文件

这种方法可以最直观的方法来识别文件类型。

直接使用“`ls`”可能看到的信息有限, 为了知道关于文档的更多资料, 我们可以增加一个参数“`-l`”, 这样可以看到更多关于文件的信息:

```
[root@Linux ~]# ls -l
总计 56
-rw----- 1 root root 865 03-15 04:12 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 03-14 20:34 Desktop
-rw-r--r-- 1 root root 27964 03-15 04:12 install.log
-rw-r--r-- 1 root root 5069 03-15 04:11 install.log.syslog
```

这样就可以知道文档的权限、拥有者、大小、日期等相关的信息，关于这些信息可能大家一时看不懂，在以后的学习中会讲这些内容的。

“ls -l”有一个简写是“ll”，如下：

```
[root@Linux ~]# ll
总计 56
-rw----- 1 root root 865 03-15 04:12 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 03-14 20:34 Desktop
-rw-r--r-- 1 root root 27964 03-15 04:12 install.log
-rw-r--r-- 1 root root 5069 03-15 04:11 install.log.syslog
```

所显示的和使用“ls -l”一模一样。

7. 显示隐藏的文件

系统中一些文件，和 Windows 中的系统文件一样，这些文件一般是不显示出来的，那么我们可以使用“-a”参数来显示隐藏的文件。如下所示：

```
[root@Linux ~]# ls -a
. .chewing .gconfd install.log .redhat
.. .cshrc .gnome install.log.syslog .scim
anaconda-ks.cfg Desktop .gnome2 .lessht .tcshrc
.bash_history .dmrc .gnome2_private .metacity .thumbnails
.bash_logout .egg cups .gstreamer-0.10 .mozilla .Trash
.bash_profile .esd_auth .gtkrc-1.2-gnome2 .nautilus
.bashrc .gconf .ICEauthority .recently-used.xbel
```

看看这样显示的文件就比较多了。可以发现，隐藏文件是以“.”开头的文件。这些文件一般都有特殊的作用，如果显示出来，有可能被用户不小心删除。

我们可以结合上面的两个参数来显示目录下的文件，如下所示：

```
[root@Linux ~]# ls -al
总计 212
```



```

drwxr-x--- 18 root root 4096 03-19 05:30 .
drwxr-xr-x 24 root root 4096 03-19 12:21 ..
-rw----- 1 root root 865 03-15 04:12 anaconda-ks.cfg
-rw----- 1 root root 2372 03-19 05:52 .bash_history
-rw-r--r-- 1 root root 24 2006-07-13 .bash_logout
-rw-r--r-- 1 root root 191 2006-07-13 .bash_profile
-rw-r--r-- 1 root root 176 2006-07-13 .bashrc
drwx----- 2 root root 4096 03-14 20:32 .chewing
-rw-r--r-- 1 root root 100 2006-07-13 .cshrc
drwxr-xr-x 2 root root 4096 03-14 20:34 Desktop
-rw----- 1 root root 26 03-14 20:31 .dmrc
drwxr-x--- 2 root root 4096 03-14 20:32 .eggcups
-rw----- 1 root root 16 03-19 05:30 .esd_auth
drwx----- 4 root root 4096 03-19 05:06 .gconf
drwx----- 2 root root 4096 03-19 05:30 .gconfd
drwxr-xr-x 3 root root 4096 03-14 20:32 .gnome
drwx----- 6 root root 4096 03-19 05:30 .gnome2
drwx----- 2 root root 4096 03-14 20:32 .gnome2_private
drwxr-xr-x 2 root root 4096 03-14 20:32 .gstreamer-0.10
-rw-r--r-- 1 root root 81 03-14 20:32 .gtkrc-1.2-gnome2
-rw----- 1 root root 358 03-19 05:30 .ICEauthority
-rw-r--r-- 1 root root 27964 03-15 04:12 install.log
-rw-r--r-- 1 root root 5069 03-15 04:11 install.log.syslog
-rw----- 1 root root 35 03-19 05:26 .lessht
drwx----- 3 root root 4096 03-14 20:32 .metacity
drwx----- 3 root root 4096 03-14 20:36 .mozilla
drwxr-xr-x 3 root root 4096 03-19 05:30 .nautilus
-rw-r--r-- 1 root root 1528 03-19 05:30 .recently-used.xbel
drwxr-xr-x 3 root root 4096 03-14 20:32 .redhat
drwx----- 4 root root 4096 03-14 21:06 .scim
-rw-r--r-- 1 root root 129 2006-07-13 .tcshrc
drwx----- 3 root root 4096 03-14 20:33 .thumbnails
drwx----- 2 root root 4096 03-14 20:32 .Trash

```

这样显示是不是觉得更好呢？

8. 显示其他目录中的文件

上面都是显示当前工作目录中的文件，如果要显示其他目录中的文件，那么我们只要在“ls”后面加入所要显示的目录的路径。如下所示：

```
[root@Linux ~]# ls /
```

```
bin dev home lost+found misc net proc sbin srv tftpboot usr
boot etc lib media mnt opt root selinux sys tmp var
```

这样显示的就是“/”目录下的文件。当然我们也可以加入一下参数，如下所示：

```
[root@Linux ~]# ls -l /
总计 148
drwxr-xr-x 2 root root 4096 03-16 17:05 bin
drwxr-xr-x 4 root root 1024 03-15 03:58 boot
drwxr-xr-x 11 root root 3920 03-19 12:21 dev
drwxr-xr-x 96 root root 12288 03-19 12:21 etc
drwxr-xr-x 4 root root 1024 03-14 20:29 home
drwxr-xr-x 14 root root 4096 03-16 17:04 lib
drwx----- 2 root root 16384 03-15 03:45 lost+found
drwxr-xr-x 2 root root 4096 03-15 04:26 media
drwxr-xr-x 2 root root 0 03-19 12:21 misc
drwxr-xr-x 2 root root 4096 2006-10-11 mnt
drwxr-xr-x 2 root root 0 03-19 12:21 net
drwxr-xr-x 2 root root 4096 2006-10-11 opt
dr-xr-xr-x 83 root root 0 03-19 12:20 proc
drwxr-x--- 18 root root 4096 03-19 05:30 root
drwxr-xr-x 2 root root 12288 03-16 17:05 sbin
drwxr-xr-x 2 root root 4096 03-15 03:50 selinux
drwxr-xr-x 2 root root 4096 2006-10-11 srv
drwxr-xr-x 11 root root 0 03-19 12:20 sys
drwxr-xr-x 2 root root 4096 2006-07-13 tftpboot
drwxrwxrwt 6 root root 4096 03-19 12:21 tmp
drwxr-xr-x 14 root root 4096 03-15 03:54 usr
drwxr-xr-x 25 root root 4096 03-15 04:04 var
```

加入“-l”参数后，显示的信息还是多多了。

9. 建立目录

当我们工作的需要，建立一个目录的时候，我们可以使用“mkdir”命令来建立一个目录，如下所示：

```
[root@Linux ~]# mkdir myfile
[root@Linux ~]# ll
总计 60
-rw----- 1 root root 865 03-15 04:12 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 03-14 20:34 Desktop
```

```
-rw-r--r-- 1 root root 27964 03-15 04:12 install.log
-rw-r--r-- 1 root root 5069 03-15 04:11 install.log.syslog
drwxr-xr-x 2 root root 4096 03-19 12:36 myfile
```

我们先使用“`mkdir myfile`”建立了一个名字叫“myfile”的目录，然后我们使用命令验证一下，发现我们已经建立了这个文件夹，可以根据颜色来判断是一个文件夹。

10. 删除目录

如果这个目录不需要了，我们可以使用“`rmdir`”命令来删除一个目录，用法如下：

```
[root@Linux ~]# ll
总计 60
-rw----- 1 root root 865 03-15 04:12 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 03-14 20:34 Desktop
-rw-r--r-- 1 root root 27964 03-15 04:12 install.log
-rw-r--r-- 1 root root 5069 03-15 04:11 install.log.syslog
drwxr-xr-x 2 root root 4096 03-19 12:36 myfile
[root@Linux ~]# rmdir myfile
[root@Linux ~]# ll
总计 56
-rw----- 1 root root 865 03-15 04:12 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 03-14 20:34 Desktop
-rw-r--r-- 1 root root 27964 03-15 04:12 install.log
-rw-r--r-- 1 root root 5069 03-15 04:11 install.log.syslog
```

我们先看看当前目录下有一个myfile的目录，我们不需要这个目录了，我们可以使用“`rmdir myfile`”命令来删除这个目录，然后使用`ll`命令验证一下。

注意的是，使用这个命令来删除一个目录的时候，那么这个目录下必须是没有任何文件和文件夹的，我们看看如下的操作：

```
[root@Linux ~]# mkdir myfile
[root@Linux ~]# cd myfile cd命令是切换目录，这里就是进入myfile
[root@Linux myfile]# mkdir one
[root@Linux myfile]# cd .. ..表示当前目录的上一层目录
[root@Linux ~]# rmdir myfile
rmdir: myfile: 目录非空
```

我们在myfile目录下建立了一个one的目录，当我们删除myfile目录时，提示

目录为空，所以必须将这个目录下的所有文件和目录全部删除之后，才能删除这个目录。

11. 切换目录

上面的例子中提到了一个“cd”命令，一般为了工作的需要，我们可能需要到不同的目录中编辑不同的文件，这个时候我们就要在不同的目录中切换了。我们使用“cd”命令可以在不同的目录中切换，如下的操作：

```
[root@Linux ~]# mkdir one
[root@Linux ~]# mkdir two
[root@Linux ~]# cd one
[root@Linux one]#
```

我们建立了两个目录“one”和“two”，我们可以使用“cd one”，进入 one 目录，进入这个目录之后，命令提示符有原来的“[root@Linux ~]”变成了“[root@Linux one]”，提示符中的“one”表示我们在 one 这个目录中工作了。

我们看看这个目录下有什么文件，为了看看有没有隐藏文件，我们使用“ls -al”命令来看看这个目录。如下操作：

```
[root@Linux one]# ls -al
总计 12
drwxr-xr-x 2 root root 4096 03-19 12:43 .
drwxr-x--- 20 root root 4096 03-19 12:43 ..
```

我们发现这个目录下面有两个奇怪的家伙，一个是“.”，一个是“..”。其实这又要讲到路径的知识，路径的表示方法有两种：

A：绝对路径：就是表示出目标文件或子目录的目录树结构。我们这个例子中的目录“one”，的绝对路径表示方法为“/root/one”，如果 one 目录下还有一个目录 xu，那么目录 xu 的绝对路径就是：“/root/one/xu”。

B：相对路径：就是相对于当前工作目录的路径来表示位置。相对路径中有两个符号，就是“..”，“.”。“..”表示当前当前工作目录的上一层目录。如当前我们处在 one 目录中，我们作如下操作：

```
[root@Linux one]# cd ..
[root@Linux ~]#
```

我们就或发现提示“[root@Linux one]”变为“[root@Linux ~]”。发现现在已经返回到目录 one 的上一层目录。

“.”表示当前的工作目录。如果当前我们处在 one 目录中，那么“.”就表示当前的工作目录 one。

12. 复制文件

工作的时候，有是需要保存文件的一个副本，便于以后还原。我们可以使用“cp”命令，操作如下：

```
[root@Linux ~]# ll
总计 64
-rw----- 1 root root 865 03-15 04:12 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 03-14 20:34 Desktop
-rw-r--r-- 1 root root 27964 03-15 04:12 install.log
-rw-r--r-- 1 root root 5069 03-15 04:11 install.log.syslog
drwxr-xr-x 2 root root 4096 03-19 12:43 one
drwxr-xr-x 2 root root 4096 03-19 12:43 two
[root@Linux ~]# cp install.log ./one
[root@Linux ~]# cd one
[root@Linux one]# ll
总计 28
-rw-r--r-- 1 root root 27964 03-19 12:55 install.log
```

先看看当前目录下有哪些文件，然后我们需要将 install.log 文件复制一份到 one 目录中，我们可以使用“cp install.log ./one”就可以了，注意了，这里使用了上面的相对路径中的“.”，表示当前工作目录下的 one 目录。

然后进 one 目录，查看一下是否有 install.log 这个文件，可以看到我们已经复制了一份 install.log 文件。

我们在来试一试，我们将目录 one 复制到 two 中，我们看看如下的操作：

```
[root@Linux ~]# ll
总计 64
-rw----- 1 root root 865 03-15 04:12 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 03-14 20:34 Desktop
-rw-r--r-- 1 root root 27964 03-15 04:12 install.log
-rw-r--r-- 1 root root 5069 03-15 04:11 install.log.syslog
drwxr-xr-x 2 root root 4096 03-19 12:55 one
drwxr-xr-x 2 root root 4096 03-19 12:43 two
[root@Linux ~]# cp one two
cp: 略过目录 “one”
```

```
[root@Linux ~]# cd two
[root@Linux two]# ll
总计 0
```

发现没有 two 目录下没有 one 文件夹，这个时候我们需要使用参数 “-r”，看如下操作：

```
[root@Linux ~]# ll
总计 64
-rw----- 1 root root 865 03-15 04:12 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 03-14 20:34 Desktop
-rw-r--r-- 1 root root 27964 03-15 04:12 install.log
-rw-r--r-- 1 root root 5069 03-15 04:11 install.log.syslog
drwxr-xr-x 2 root root 4096 03-19 12:55 one
drwxr-xr-x 2 root root 4096 03-19 12:43 two
[root@Linux ~]# cp -r one two
[root@Linux ~]# cd two
[root@Linux two]# ll
总计 4
drwxr-xr-x 2 root root 4096 03-19 13:00 one
```

看看，在 cp 命令后面加入参数 “-r” 之后，就可以用来复制目录了，我们可以进入 two 目录下的子目录 one，会发现原来 one 目录中的文件也一起被复制了。

最后来看看复制文件并重新命名的新文件，实例如下：

```
[root@Linux one]# cp install.log install.log.bak
[root@Linux one]# ll
总计 56
-rw-r--r-- 1 root root 27964 03-19 12:55 install.log
-rw-r--r-- 1 root root 27964 03-20 13:10 install.log.bak
```

这里就是复制了一个文件，并将复制的文件更名，我们可以看到目录下有两个文件了。

13. 移动文件

我们可以使用 mv 命令将一个文件移动到另外一个目录中，如下操作：

```
[root@Linux ~]# ll
总计 64
-rw----- 1 root root 865 03-15 04:12 anaconda-ks.cfg
```

```

drwxr-xr-x 2 root root 4096 03-14 20:34 Desktop
-rw-r--r-- 1 root root 27964 03-15 04:12 install.log
-rw-r--r-- 1 root root 5069 03-15 04:11 install.log.syslog
drwxr-xr-x 2 root root 4096 03-20 13:10 one
drwxr-xr-x 3 root root 4096 03-19 13:00 two
[root@Linux ~]# mv install.log.syslog ./one
[root@Linux ~]# cd one
[root@Linux one]# ll
总计 68
-rw-r--r-- 1 root root 27964 03-19 12:55 install.log
-rw-r--r-- 1 root root 27964 03-20 13:10 install.log.bak
-rw-r--r-- 1 root root 5069 03-15 04:11 install.log.syslog

```

我们将 root 用户目录下的 install.log.syslog，移动到目录 one 下。除此之外，我们还可以为文件更名，如下操作：

```

[root@Linux one]# ll
总计 68
-rw-r--r-- 1 root root 27964 03-19 12:55 install.log
-rw-r--r-- 1 root root 27964 03-20 13:10 install.log.bak
-rw-r--r-- 1 root root 5069 03-15 04:11 install.log.syslog
[root@Linux one]# mv install.log install.sys
[root@Linux one]# ll
总计 68
-rw-r--r-- 1 root root 27964 03-20 13:10 install.log.bak
-rw-r--r-- 1 root root 5069 03-15 04:11 install.log.syslog
-rw-r--r-- 1 root root 27964 03-19 12:55 install.sys

```

我们将 install.log 这个文件名称变为了 install.sys。

14. 创建文件

这个和前面的创建目录是有区别的，创建文件是指创建一个一般的普通文件，并且这个文件为空，我们可以使用 touch 命令来建立一般文件，如下操作：

```

[root@Linux two]# touch 111.txt
[root@Linux two]# touch 112.txt
[root@Linux two]# touch 113.txt
[root@Linux two]# ll
总计 0
-rw-r--r-- 1 root root 0 03-20 13:26 111.txt
-rw-r--r-- 1 root root 0 03-20 13:26 112.txt

```

```
-rw-r--r-- 1 root root 0 03-20 13:26 113.txt
```

这样我们就创建了三个文件。

15. 删除文件

使用 `rm` 命令来删除文件，和前面的 `rmdir` 有点区别，不过 `rm` 也可以实现删除目录，我们来看看实际的操作：

```
[root@Linux two]# ll
总计 0
-rw-r--r-- 1 root root 0 03-20 13:26 111.txt
-rw-r--r-- 1 root root 0 03-20 13:26 112.txt
-rw-r--r-- 1 root root 0 03-20 13:26 113.txt
[root@Linux two]# mkdir one
[root@Linux two]# ll
总计 4
-rw-r--r-- 1 root root 0 03-20 13:26 111.txt
-rw-r--r-- 1 root root 0 03-20 13:26 112.txt
-rw-r--r-- 1 root root 0 03-20 13:26 113.txt
drwxr-xr-x 2 root root 4096 03-20 13:27 one
[root@Linux two]# rm 111.txt
rm: 是否删除 一般空文件 “111.txt” ? y
[root@Linux two]# ll
总计 4
-rw-r--r-- 1 root root 0 03-20 13:26 112.txt
-rw-r--r-- 1 root root 0 03-20 13:26 113.txt
drwxr-xr-x 2 root root 4096 03-20 13:27 one
```

我们先查看目录下的文件，然后删除这个目录下的文件 `111.txt`，删除的时候，提示我们是否要删除文件，我们输入“y”，然后删除文件，删除之后，我们查看这个文件夹下的文件，发现少了 `111.txt` 文件。

如果我们不需要系统提示我们就要增加一个参数“`-f`”，这样只要输入命令之后，系统直接删除文件。看如下操作：

```
[root@Linux two]# rm -f 112.txt
[root@Linux two]# ll
总计 4
-rw-r--r-- 1 root root 0 03-20 13:26 113.txt
drwxr-xr-x 2 root root 4096 03-20 13:27 one
```


这样增加参数后，系统没有提示用户，直接删除了文件。

我们在来看看如何删除这个文件下的目录，增加参数“-r”就可以了，看如下操作：

```
[root@Linux two]# ll
总计 4
-rw-r--r-- 1 root root 0 03-20 13:26 113.txt
drwxr-xr-x 2 root root 4096 03-20 13:27 one
[root@Linux two]# rm -rf one
[root@Linux two]# ll
总计 0
-rw-r--r-- 1 root root 0 03-20 13:26 113.txt
```

这样，这个目录下的 one 目录就被删除了。

16. 通配符

以上所有的操作我们好像都只能操作一个文件，当我们要同时对多个文件进行操作的时候，我们就可以使用通配符，这些符号不是作为普通的文字来使用，而是来代替文键名称的字符串。我们来看看 Linux 下的几种通配符，和他们的含义：

A: * : 任何字符和字符串。

B: ? : 一个任意字符

C: [abc...] : []内的任意一个字符。[abc]表示 a、b、c 任一字符；有时候也表示范围，如[a-x],表示 a 到 x 的任一字符；[1-9]表示 1 到 9 的任一数字。

D: [!abc...] : 和上面的相反，表示除[]内的字符外的任意一个字符。

我们来分别看看这些符号的实际操作中的实例：

```
[root@Linux two]# ll
总计 0
-rw-r--r-- 1 root root 0 03-20 20:08 111.txt
-rw-r--r-- 1 root root 0 03-20 20:08 112.txt
-rw-r--r-- 1 root root 0 03-20 20:08 113.txt
[root@Linux two]# rm 1*.txt
rm: 是否删除 一般空文件 “111.txt” ? y
rm: 是否删除 一般空文件 “112.txt” ? y
rm: 是否删除 一般空文件 “113.txt” ? y
[root@Linux two]# ll
总计 0
```

这样只要是以“1”开头的文件全部删除。我们在来看下面的操作：

```
[root@Linux two]# ll
总计 0
-rw-r--r-- 1 root root 0 03-20 20:09 111.txt
-rw-r--r-- 1 root root 0 03-20 20:09 112.txt
-rw-r--r-- 1 root root 0 03-20 20:09 113.txt
-rw-r--r-- 1 root root 0 03-20 20:10 121.txt
[root@Linux two]# rm -f 11?.txt
[root@Linux two]# ll
总计 0
-rw-r--r-- 1 root root 0 03-20 20:10 121.txt
```

这样就删除了以“11”开头的文件，我们还是以上面的例子在看：

```
[root@Linux two]# ll
总计 0
-rw-r--r-- 1 root root 0 03-20 20:11 111.txt
-rw-r--r-- 1 root root 0 03-20 20:11 112.txt
-rw-r--r-- 1 root root 0 03-20 20:11 113.txt
-rw-r--r-- 1 root root 0 03-20 20:10 121.txt
[root@Linux two]# rm -f 11[1-3].txt
[root@Linux two]# ll
总计 0
-rw-r--r-- 1 root root 0 03-20 20:10 121.txt
```

这样就删除了以“11”开头，后面是以1、2、3结尾的文件。最后我们来看：

```
[root@Linux two]# ll
总计 0
-rw-r--r-- 1 root root 0 03-20 20:13 111.txt
-rw-r--r-- 1 root root 0 03-20 20:13 112.txt
-rw-r--r-- 1 root root 0 03-20 20:13 113.txt
-rw-r--r-- 1 root root 0 03-20 20:10 121.txt
[root@Linux two]# rm -f 1[!1]*.txt
[root@Linux two]# ll
总计 0
-rw-r--r-- 1 root root 0 03-20 20:13 111.txt
-rw-r--r-- 1 root root 0 03-20 20:13 112.txt
-rw-r--r-- 1 root root 0 03-20 20:13 113.txt
```

这样就删除了第二个字符不是为“1”的任何文件。

有了通配符，我们就可以针对多个文件一起进行操作。

17. pwd

如果切换目录次数太多，可能不知道自己在哪个目录，那么就可以使用 pwd 命令，显示当前所在的目录。是 print working directory 的缩写。看如下操作：

```
[root@Linux ~]# pwd
/root
[root@Linux ~]# cd one
[root@Linux one]# pwd
/root/one
[root@Linux one]# cd ../two
[root@Linux two]# pwd
/root/two
```

这样就很清楚自己在那个目录下工作了。

18. 增加用户

我们来看看增加用户的操作。到目前为止，我们都是使用的 root 用户登录系统，由于 root 这个用户的权限非常的高，可以进行任何的操作，所以有的时候，可能破坏系统。如前面的操作，如果执行了一个“rm -rf /*”这个命令，这样就会把“/”下面的文件全部删除。我们如果使用一般的用户，我们还是可以执行大部分的程序，除了少数的一些管理员的命令不能执行外，一些目录不能进行操作外，其他的操作我们都可以。所以建议：在使用一般操作的时候，我们使用一般用户，如果需要对系统进行维护，我们就使用 root 用户。

现在来看看如何增加一般用户。看如下操作：

```
[root@Linux one]# useradd user1
```

我们就为系统增加了一个用户“user1”，z 增加完用户后，我们需要对用户设置密码，看如下操作：

```
[root@Linux one]# passwd user1
Changing password for user user1.
New UNIX password: <--这里输入用户的密码
BAD PASSWORD: it is too simplistic/systematic
Retype new UNIX password: <--再次输入上面的密码
passwd: all authentication tokens updated successfully. <--提示用户设置密码成功
```

注意设置密码的时候，为了安全，一般不显示任何的字符。

这里我们知道了使用“useradd”命令来增加用户，使用“passwd”命令来为用户设置密码。到此，我们就为系统增加了一个用户“user1”，下次登录的时候，我们就可以使用这个用户了。

注意，这里只有 root 用户可以设置其他用户的密码，一般用户登录后可以更改自己的密码，但不能更改别人的密码。看如下操作：

```
[user1@Linux ~]$ passwd root
passwd: Only root can specify a user name.
```

提示只有 root 用户才能更改自己的密码。我们在来看看用户更改自己的密码：

```
[user1@Linux ~]$ passwd
Changing password for user user1.
Changing password for user1
(current) UNIX password: <--输入当前的密码
New UNIX password: <--输入新密码
BAD PASSWORD: is too simple <--提示密码太简单
New UNIX password: <--输入复杂的新密码
Retype new UNIX password: <--再次输入上面输入的密码
passwd: all authentication tokens updated successfully. <--提示密码
更改成功
```

这个时候 passwd 后面不需要跟用户名称。

19. 搜索文件

我们先来学习一下如何搜索文件，特别是刚开始学习 Linux 的时候，自己建立的文件不知道放在哪里了，常有发生。如果知道文件名，却不知道文件在那个目录下面了，我们就可以使用 locate 命令来搜索文件。看如下操作：

```
[root@Linux one]# locate install.log
/root/install.log
/root/install.log.syslog
```

看一下，我们一下就搜索了两个与 install.log 相关的文件，他们都在 /root 目录下，同时我们感觉到，使用这个命令搜索文件的速度比较快，其实要使用这个命令，必须配合数据库来使用，因为这个命令是从数据库中来搜索文件，这个数据库的更新速度是 7 天更新一次。如下操作：

```
[root@Linux one]# touch 001.txt
[root@Linux one]# locate 001.txt
```

发现这个命令找不到新建立的文件，所以我们要使用这个命令搜索文件之前，必须自己更新一下数据库，如下操作：

```
[root@Linux one]# updatedb
[root@Linux one]# locate 001.txt
/root/one/001.txt
```

看看，如果执行 updatedb 这个命令更新数据库之后，我们就可以找到我们所需要的数据。不过更新数据库的时间需要一段时间。

20. 显示文件内容

20.1 cat

是直接显示文件的内容到屏幕上，如下操作：

```
[root@Linux ~]# cat install.log
...
...
...
安装 vnc - 4.1.2-9.el5.i386
安装 screen - 4.0.3-1.el5.i386
安装 scim-bridge-gtk - 0.4.5-7.el5.i386
安装 scim-tables-chinese - 0.5.6-7.i386
安装 minicom - 2.1-3.i386
安装 Deployment_Guide-zh-CN - 5.0.0-19.noarch
安装 Deployment_Guide-zh-TW - 5.0.0-19.noarch
安装 scim-pinyin - 0.5.91-15.el5.i386
安装 scim-chewing - 0.3.1-10.el5.i386
安装 scim-chinese-standard - 0.0.2-1.el5.i386
安装 vim-X11 - 2:7.0.109-3.i386
安装 gftp - 1:2.0.18-3.2.2.i386
安装 gtkspell - 2.0.11-2.1.i386
安装 openldap-clients - 2.3.27-5.i386
安装 nmap - 2:4.11-1.1.i386
安装 OpenIPMI-tools - 2.0.6-5.el5.3.i386
安装 dhcp - 12:3.0.5-3.el5.i386
```

```
安装 zsh - 4.2.6-1.i386
安装 xchat - 1:2.6.6-8.el5.i386
安装 fonts-chinese - 3.02-9.6.el5.noarch
安装 stardict - 2.4.5-5.i386
安装 system-config-bind - 4.0.3-2.el5.noarch
```

前面的内容比较多，用“...”省略，我们只看到了这个文件的后面的部分，由此可以知道，这个命令适合于显示内容少于屏幕所能显示的最大行数。

这个命令还有一个 `tac` 的相反的写法，大家想想应该显示什么？

20.2 more

`cat` 命令我们只能看到文件的部分或者只能看比较小的文件，那么 `more` 解决这个问题，操作如下：

```
[root@Linux ~]# more install.log
安装 libgcc - 4.1.1-52.el5.i386
warning: libgcc-4.1.1-52.el5: Header V3 DSA signature: NOKEY, key ID
37017186
安装 setup - 2.5.58-1.el5.noarch
安装 filesystem - 2.4.0-1.i386
安装 basesystem - 8.0-5.1.1.noarch
安装 tzdata - 2006m-2.fc6.noarch
安装 glibc-common - 2.5-12.i386
安装 glibc - 2.5-12.i686
安装 glib2 - 2.12.3-2.fc6.i386
安装 zlib - 1.2.3-3.i386
安装 chkconfig - 1.3.30.1-1.i386
安装 atk - 1.12.2-1.fc6.i386
安装 popt - 1.10.2-37.el5.i386
安装 libICE - 1.0.1-2.1.i386
安装 libSM - 1.0.1-3.1.i386
安装 libstdc++ - 4.1.1-52.el5.i386
安装 libart_lgpl - 2.3.17-4.i386
安装 mktmp - 3:1.5-23.2.2.i386
安装 audit-libs - 1.3.1-1.el5.i386
安装 libusb - 0.1.12-5.1.i386
安装 bzip2-libs - 1.0.3-3.i386
安装 libpng - 2:1.2.10-7.i386
安装 libattr - 2.4.32-1.1.i386
--More--(2%)
```

注意最下面显示了一个提示，表示只显示这个文件的 2%，按下空格键就翻下一页，当内容显示完后，就自动的退出，还可以按下“Q”键也可以退出。

20.3 less

more 虽然可以显示比较大的文件，但是也有一个缺点，就是不能向上翻页，看过的内容要重新看，必须在重新运行命令。less 就解决了 more 命令这个命令的不足。看如下操作：

```
[root@Linux ~]# less install.log
安装 libgcc - 4.1.1-52.el5.i386
warning: libgcc-4.1.1-52.el5: Header V3 DSA signature: NOKEY, key ID
37017186
安装 setup - 2.5.58-1.el5.noarch
安装 filesystem - 2.4.0-1.i386
安装 basesystem - 8.0-5.1.1.noarch
安装 tzdata - 2006m-2.fc6.noarch
安装 glibc-common - 2.5-12.i386
安装 glibc - 2.5-12.i686
安装 glib2 - 2.12.3-2.fc6.i386
安装 zlib - 1.2.3-3.i386
安装 chkconfig - 1.3.30.1-1.i386
安装 atk - 1.12.2-1.fc6.i386
安装 popt - 1.10.2-37.el5.i386
安装 libICE - 1.0.1-2.1.i386
安装 libSM - 1.0.1-3.1.i386
安装 libstdc++ - 4.1.1-52.el5.i386
安装 libart_lgpl - 2.3.17-4.i386
安装 mktemp - 3:1.5-23.2.2.i386
安装 audit-libs - 1.3.1-1.el5.i386
安装 libusb - 0.1.12-5.1.i386
安装 bzip2-libs - 1.0.3-3.i386
安装 libpng - 2:1.2.10-7.i386
安装 libattr - 2.4.32-1.1.i386
install.log
```

最下面显示的是这个文件的名称，我们可以使用“PageUp”和“PageDown”可以进行上一页和下一页的翻页。如果要知道具体的控制键，我们可以按下“H”键，可以显示 less 命令的所有控制键，如果想结束，可以按“Q”键。

20.4 head 和 tail

上面是最常用的几个显示文件内容的命令，这里还介绍两个显示文件内容的命令，分别是 head 和 tail 命令，head 是显示文件的头几行，tail 是显示文件的末尾几行，默认情况下，都是 10 行。看如下操作：

```
[root@Linux ~]# head install.log
安装 libgcc - 4.1.1-52.el5.i386
warning: libgcc-4.1.1-52.el5: Header V3 DSA signature: NOKEY, key ID
37017186
安装 setup - 2.5.58-1.el5.noarch
安装 filesystem - 2.4.0-1.i386
安装 basesystem - 8.0-5.1.1.noarch
安装 tzdata - 2006m-2.fc6.noarch
安装 glibc-common - 2.5-12.i386
安装 glibc - 2.5-12.i686
安装 glibc2 - 2.12.3-2.fc6.i386
安装 zlib - 1.2.3-3.i386
```

```
[root@Linux ~]# tail install.log
安装 gtkspell - 2.0.11-2.1.i386
安装 openldap-clients - 2.3.27-5.i386
安装 nmap - 2:4.11-1.1.i386
安装 OpenIPMI-tools - 2.0.6-5.el5.3.i386
安装 dhcp - 12:3.0.5-3.el5.i386
安装 zsh - 4.2.6-1.i386
安装 xchat - 1:2.6.6-8.el5.i386
安装 fonts-chinese - 3.02-9.6.el5.noarch
安装 stardict - 2.4.5-5.i386
安装 system-config-bind - 4.0.3-2.el5.noarch
```

分别显示的是这个文件的前 10 行与后 10 行。

21. 基础

21.1 基础一

使用 **root** 登录 进入系统后默认进入的用户的 **home** 目录，可以使用 **pwd** 打印当前工作目录的位置 在根目录下输入 **ls /** 查看。

由于 **root** 用户权限比较高，不要轻易使用，先建立一个普通用户，使用 **useradd rathat** 命令建立 **rathat** 命令 **passwd redhat** 命令创建密码。

普通用户的目录放在 **home** 目录下 **ls /home** 查看，**ls -ls /home/redhat** 查看其权限等。推出系统 **exit**，用刚才建立的用户登录系统。 **ls -a** 查看隐藏文件。使用 **su -**以超级用户身份登录，**whoami** 查看当前登录用户。看一下根目录中的子目录 **ls -l /**

第一个目录 **bin** 放置的是二进制文件程序工具，是系统不可缺少的工具机，我们运行的命令一般都放在这里 **ls dir**

第二个目录 **boot** 方的是系统启动的必须文件

Dev 目录方的是系统的设备 **linux** 将系统所需的外设都堪称文件

Etc 系统的配置文件所存放的位置，系统所有的配置文件都放在这下面

Home 普通用户的防止位置，刚装好后是空的 随着用户增多而增加

Initrd 临时目录，初始化引导时候用的

Lib 动态链接库存放的位置

Lost +found 自检的时候发现的碎片文件，某个文件丢失了可能在他里找回了

Misc 基本不用的，杂七杂八的

Mnt 挂在外部设备的 ，光驱

Opt 安装第三方软件的

Proc 虚拟的文件系统 反应的是进程信息，不占用硬盘空间的

root 管理的目录

Sbin 超级用户的管理工具 系统不可去少的

Tmp 临时文件 全局可些的

User 所有用户公共的资料

Var 动态变化的数据，数据库，日志等

Var 和 home 是变化的经常需要维护更新

```
lredhat@server redhat1$ ls -l /
total 196
drwxr-xr-x  2 root    root      4096 Jun  9 14:42 bin
drwxr-xr-x  3 root    root      4096 Jun  7 11:32 boot
drwxr-xr-x 21 root    root     118784 Jun  9 15:50 dev
drwxr-xr-x 42 root    root      4096 Jun  7 16:00 etc
drwxr-xr-x  3 root    root      4096 Jun  9 16:08 home
drwxr-xr-x  2 root    root      4096 Oct  7 2003 initrd
drwxr-xr-x  9 root    root      4096 Jun  9 14:40 lib
drwx      2 root    root     16384 Jun  7 19:25 lost+found
drwxr-xr-x  2 root    root      4096 Sep  9 2003 misc
drwxr-xr-x  4 root    root      4096 Jun  7 11:16 opt
drwxr-xr-x  3 root    root      4096 Oct  7 2003 opt
dr-xr-xr-x 13 root    root         0 Jun  9 2004 proc
drwxr-xr-x  2 root    root      4096 Jun  7 11:51 root
drwxr-xr-x  2 root    root      8192 Jun  9 14:42 sbin
drwxrwxrwt  3 root    root      4096 Jun  9 15:50 tmp
drwxr-xr-x 15 root    root      4096 Jun  7 11:30 usr
drwxr-xr-x 16 root    root      4096 Jun  7 11:35 var
lredhat@server redhat1$ pwd
/home/redhat
lredhat@server redhat1$ _
```

21.2 基础二

如何了解系统里面分区的情况呢？

使用 df 查看当前系统中所有分区占用的大小

```
drwxr-xr-x 16 root    root      4096 Jun  7 11:35 var
lredhat@server redhat1$ pwd
/home/redhat
lredhat@server redhat1$ df
Filesystem            1K-blocks      Used Available  Use% Mounted on
/dev/sda1             3020140    787412    2079312   28% /
/dev/sda3              845948     16128     786548    3% /opt
none                  629000         0      629000    0% /dev/shm
lredhat@server redhat1$ _
```

df 的结果不是很好认识，加后缀-h 比较好读 每一个结果都会有一个后缀

```

drwxr-xr-x 16 root root 4096 Jun 7 11:35 .
[redhat@server redhat]$ pwd
/home/redhat
[redhat@server redhat]$ df
Filesystem            1k-blocks      Used Available Use% Mounted on
/dev/sda1             3020148    787412    2079312   28% /
/dev/sda3              845948     16428     786548    3% /opt
none                   62988      0         62988    0% /dev/shm
[redhat@server redhat]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       2.9G  769M  2.0G   28% /
/dev/sda3       827M   17M  769M    3% /opt
none            62M    0    62M    0% /dev/shm
[redhat@server redhat]$ _

```

Df 只能查看每个分区的使用情况不能使用每个目录的使用情况，如果向查看每个目录的大小使用 du du /* 根目录下的所有文件夹

```

du: cannot read directory '/usr/share/ssl/CAs': Permission denied
[redhat@server redhat]$ su -
Password:
su: incorrect password
[redhat@server redhat]$ su -
Password:
[root@server root]# du -sh /*
4.9M    /bin
2.4M    /boot
424K    /dev
7.4M    /etc
20K     /home
4.0K    /initrd
49M     /lib
16K     /lost+found
4.0K    /misc
12K     /mnt
20K     /opt
2.0K    /proc
80K     /root
13M     /sbin
0.0K    /tmp

```

21.3 基础三

这个结果中看出最占用的是 usr 。fdisk -l /dev/sda 反应的是分区情况

```

[root@server root]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       2.9G  769M  2.0G  28% /
/dev/sda3       827M   17M  769M   3% /opt
none            62M    0   62M   0% /dev/shm
[root@server root]# fdisk -l /dev/hda
[root@server root]# fdisk -l /dev/sda

Disk /dev/sda: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1         382       3068383+   83   Linux
/dev/sda2             383         415        265072+   82   Linux swap
/dev/sda3             416         522       859477+   83   Linux
[root@server root]# _

```

检查网络环境:

首先使用 ping 检查是否能与外网通讯 结果不能解析主机名称。然后查看 dns 的客户端配置文件使用 more /etc/resolv.conf (保存的默认的域名及 dns 服务器)

```

[root@server root]# hdparm -d1 /dev/hda
[root@server root]# more /etc/rc.local
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
[root@server root]# ping www.redhat.org.cn
ping: unknown host www.redhat.org.cn
[root@server root]# more /etc/resolv.conf
search redhat.org.cn
nameserver 202.96.209.5
nameserver 202.96.209.133
[root@server root]#

```

首先限于 dns 的 ip 地址通讯不通

检查 ip 地址的配置 ifconfig

```

[root@server root]# ping 202.96.209.5
connect: Network is unreachable
[root@server root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:52:A7:3D
          inet addr:192.168.0.254  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:472 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:30341 (29.6 Kb)  TX bytes:476 (476.0 b)
          Interrupt:10 Base address:0x1080

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:700 (700.0 b)  TX bytes:700 (700.0 b)

[root@server root]# ifconfig

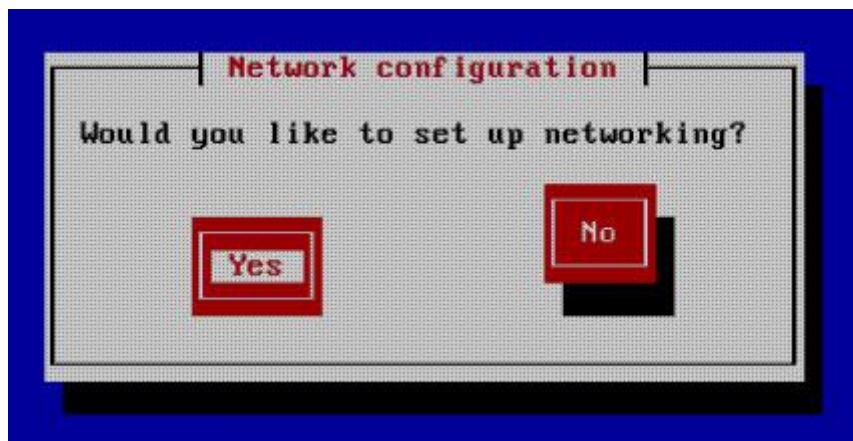
```

21.4 基础四

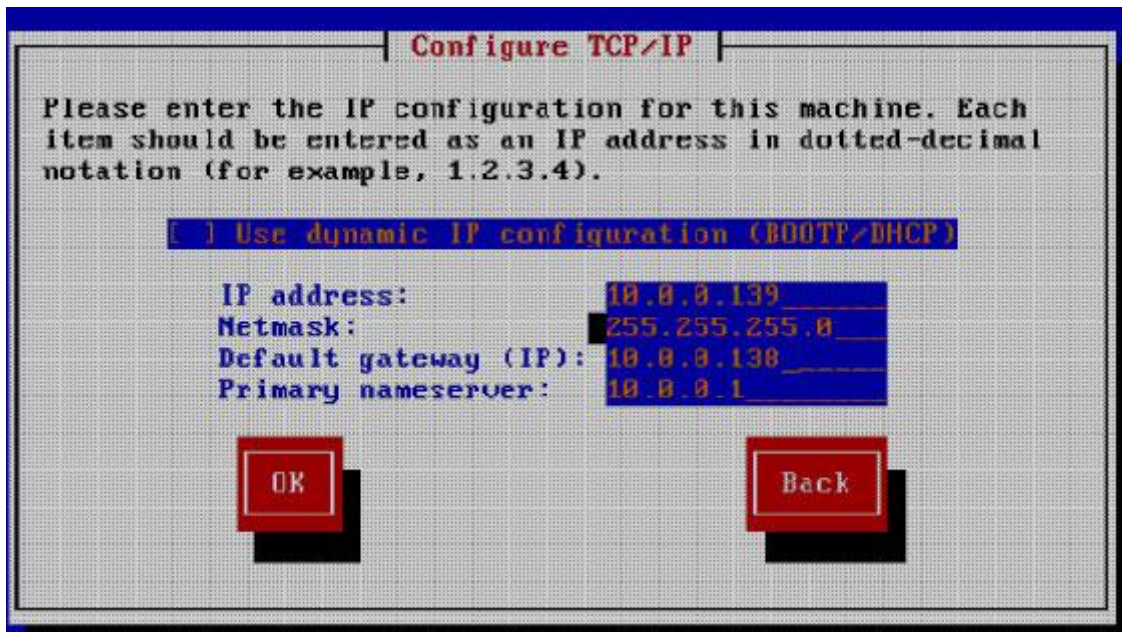
Eth0 网卡接口，默认可打印所有网络接口的信息，ip 地址是 192.168.0.254 不符合

Lo 是回环设备

更改 ip 地址和默认网关 使用工具 netconfig



Ok 回车配置 ip 及 dns 网关等



Ok 继续

此时这个配置信息没有马上保存，需运行 `service network restart` 重新启动网络服务，如果全都看到 ok 的话说明修改成功了，使用 `route -n` 检查一下默认路由看到网关是 10.0.0.138 说明生效了

在网络配置过程中如果不肯定整个网络环境可以使用基本环境进行测试，可以使用 `ping -b ip 地址` 可以扫描整个网段，ip 地址使用广播地址，如此时使用 10.0.0.255。

21.5 基础五

`ifconfig` 命令还可以修改网络接口 ip 地址 使用 `ifconfig eth0 10.0.0.254`，默认情况现会默认分配掩码地址，我们可以使用 `netmask` 来改变掩码

```

RX bytes:1288 (1.2 Kb)  TX bytes:1288 (1.2 Kb)

root@server root# ifconfig eth0 10.0.0.254 netmask 255.255.255.0
root@server root# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:52:A7:3D
          inet addr:10.0.0.254  Bcast:10.255.255.255  Mask:255.255.255.255
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:585 errors:0 dropped:0 overruns:0 frame:0
          TX packets:46 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:38099 (37.2 Kb)  TX bytes:3634 (3.5 Kb)
          Interrupt:10 Base address:0x1080

```

使用 `route add default gw 10.0.0.138` 要到达任何网络默认要经过 10.0.138 ，修改网关。

Route 和 ifconfig 修改之后不会马上保存 重启后丢失。建议使用 `netconfig` 命

令。这个命令把修改信息写进了一个配置文件，配置文件的位置是 `/etc/sysconfig/network-scripts/`

```
Interrupt:10 Base address:0x1030
[root@server root]# ifconfig eth0
[root@server root]# ls /etc/sysconfig/network-scripts/
ifcfg-eth0      ifdown-ipv6    ifup           ifup-isdn      ifup-sit
ifdown-ipv6     ifdown-ipv6    ifup-aliases  ifup-plip      ifup-sl
ifdown          ifdown-post    ifup-ppp       ifup-plusb     ifup-wireless
ifdown-aliases  ifdown-ppp     ifup-ipsec     ifup-post      init.ipv6-global
ifdown-ippv     ifdown-sit     ifup-ipv6      ifup-ppp       network-functions
ifdown ipsec    ifdown sl      ifup ipx       ifup routes    network-functions ipv6
[root@server root]#
```

21.6 基础六

Netconfig 写进了 `ifcfg-eth0` 这个文件，看一下这个文件是不是和 netconfig 修改的一致，使用 `cat` 查看里面的信息 `cat /etc/sysconfig/network-scripts/ifcfg-eth0`

```
ifdown-ippv     ifdown-sit     ifup-ipv6      ifup-ppp       network-functions
ifdown-ipsec    ifdown-sl      ifup-ipx       ifup-routes    network-functions-ipv6
[root@server root]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=10.0.0.139
NETMASK=255.255.255.0
GATEWAY=10.0.0.138
[root@server root]# service network restart
```

使用 `service network restart` 就是加载这个文件的配置信息当重新运行这个命令是 会加载这个文件中的信息，而冲掉刚才使用 `ifconfig` 修改的 ip 地址。

如果修改 ip 地址只需修改刚才的文件就可以了 使用 `vi` 编辑器打开我们的配置文件 `vi /etc/sysconfig/network-scripts/ifcfg-eth0` 在里面修改 ip 地址就可以了，保存配置，重启服务 `service network restart` 这部工作是系统启动激活网卡的工作，激活网络就是靠的这个脚本。

在配置网卡的时候有一个问题要注意 在 `sysconfig` 目录下面有两个相关的目录一个是 `network-scripts` 另一个是 `networking`，`net` 命令会默认把配置信息保存到 `profiles` 而且这个是优先的。

接下来看

用户的语言环境

21.7 基础七

运行 `locale` 能看语言环境

```
[root@server root]# locale
LANG=en_US.UTF-8
LC_CTYPE="en_US.UTF-8"
LC_NUMERIC="en_US.UTF-8"
LC_TIME="en_US.UTF-8"
LC_COLLATE="en_US.UTF-8"
LC_MONETARY="en_US.UTF-8"
LC_MESSAGES="en_US.UTF-8"
LC_PAPER="en_US.UTF-8"
LC_NAME="en_US.UTF-8"
LC_ADDRESS="en_US.UTF-8"
LC_TELEPHONE="en_US.UTF-8"
LC_MEASUREMENT="en_US.UTF-8"
LC_IDENTIFICATION="en_US.UTF-8"
LC_ALL=
[root@server root]# _
```

在显示中有两个其主要作用 lang 和 lc_all

Locale -a 查看系统所支持的语言信息

设置 lang 的变量 lang=zh_cn.gb2312 在运行 date 看到显示的是中文的格式

在改回到英语 lang=en_us.utf-8 运行 date 区别如图下

```
zu_ZA
zu_ZA.iso88591
zu_ZA.utf8
[root@server root]# LANG=zh_CN.GB2312
[root@server root]# date
  6  9 16:48:25 CST 2004
[root@server root]# LANG=en_US.UTF-8
[root@server root]# date
Wed Jun  9 16:49:00 CST 2004
[root@server root]# _
```

保存语言信息的文件在/etc/sysconfig/i18n 中。

21.8 基础八

Sysconfig 目录是大多数服务器程序系统启动脚本的主要配置文件，这是我们来查看一个 hwconf 文件，这个文件是保存的系统硬件信息，输入 more /etc/sysconfig/hwconf 可以看到如下


```

[root@server root]# more /etc/sysconfig/hwconf
-
class: OTHER
bus: PCI
detached: 0
driver: agpgart
desc: "Intel Corp. i440BX/ZX/DX - 82443BX/ZX/DX Host bridge"
vendorId: 8086
deviceId: 7198
subVendorId: 15ad
subDeviceId: 1976
pciType: 1
pciBus: 0
pciDev: 0
pciFn: 0
-
class: OTHER
bus: PCI
detached: 0
driver: ignore
desc: "Intel Corp. i440BX/ZX/DX - 82443BX/ZX/DX AGP bridge"
vendorId: 8086
deviceId: 7191
subVendorId: 8008
--More--(13%)

```

这里系统中所有硬件信息，硬件资料。安装的时候有一个工具叫 kudzu 是检查有无新增硬件信息，当我们服务器新装硬件后运行 `service kudzu start` 可以让计算机自动识别，检查过程，kudzu 是自动运行的脚本，如图所示：

```

pciBus: 0
pciDev: 7
pciFn: 1
[root@server root]# service kudzu start
Checking for new hardware [ OK ]
Updating /etc/fstab [ OK ]
[root@server root]# chkconfig _

```

21.9 基础九

如果不详让他开机自动运行可以将其关闭，修改系统启动服务脚本 可以使用 `chkconfig kudzu --list` 检查状态，默认 345 是自动运行的

```

[root@server root]# service kudzu start
Checking for new hardware [ OK ]
Updating /etc/fstab [ OK ]
[root@server root]# chkconfig deviceId: 7112
chkconfig version 1.3.9 - Copyright (C) 1997-2000 Red Hat, Inc.
This may be freely redistributed under the terms of the GNU Public License.

usage:  chkconfig --list [name]
        chkconfig --add <name>
        chkconfig --del <name>
        chkconfig [--level <levels>] <name> <on|off|reset>
[root@server root]#
[root@server root]# chkconfig kudzu --list
kudzu      0:off  1:off  2:off  3:on   4:on   5:on   6:off
[root@server root]# setup_

```

如果向调整其运行状态可以运行 `setup` 这个一个综合工具。包括如图所示的五项：认证方式、防火墙配置、鼠标配置、网络配置、系统服务等。这里包含了各种系统服务。



运行 如图



21.10 基础十

在这里我们选择不需要的服务来关闭。系统默认运行的服务 可以使用 `chkconfig -list | more` 来显示出来 如图：

```

[root@server root]# chkconfig --list |more
gpm            0:off  1:off  2:on   3:on   4:on   5:on   6:off
kudzu          0:off  1:off  2:off  3:off  4:on   5:on   6:off
syslog         0:off  1:off  2:on   3:on   4:on   5:on   6:off
rawdevices     0:off  1:off  2:off  3:on   4:on   5:on   6:off
netfs          0:off  1:off  2:off  3:on   4:on   5:on   6:off
network        0:off  1:off  2:on   3:on   4:on   5:on   6:off
random         0:off  1:off  2:on   3:on   4:on   5:on   6:off
ssslauthd      0:off  1:off  2:off  3:off  4:off  5:off  6:off
iptables       0:off  1:off  2:on   3:on   4:on   5:on   6:off
anacron        0:off  1:off  2:on   3:on   4:on   5:on   6:off
atd            0:off  1:off  2:off  3:on   4:on   5:on   6:off
irda           0:off  1:off  2:off  3:off  4:off  5:off  6:off
acpid          0:off  1:off  2:off  3:on   4:on   5:on   6:off
apmd           0:off  1:off  2:on   3:on   4:on   5:on   6:off
irqbalance     0:off  1:off  2:off  3:on   4:on   5:on   6:off
pcmcia         0:off  1:off  2:on   3:off  4:on   5:on   6:off
nfslock        0:off  1:off  2:off  3:on   4:on   5:on   6:off
nfs            0:off  1:off  2:off  3:off  4:off  5:off  6:off
microcode_ctl  0:off  1:off  2:on   3:on   4:on   5:on   6:off
smartd         0:off  1:off  2:on   3:on   4:on   5:on   6:off
lshn           0:off  1:off  2:on   3:off  4:on   5:on   6:off
autofs         0:off  1:off  2:off  3:on   4:on   5:on   6:off
sshd           0:off  1:off  2:on   3:on   4:on   5:on   6:off
--More--

```

显示了运行级别 0 到运行级别六的情况、

这些服务都是靠系统脚本 init 启动的还有一些不是靠系统脚本启动的而下面会看到一些特殊服务，他们不是靠 init 启动的是靠 xinetd 启动的，是一个独立的互联网服务器的服务器是一个超级服务其，可以启动很多的子服务器。

如图

```

lshn           0:off  1:off  2:on   3:off  4:on   5:on   6:off
autofs         0:off  1:off  2:off  3:on   4:on   5:on   6:off
sshd           0:off  1:off  2:on   3:on   4:on   5:on   6:off
portmap        0:off  1:off  2:off  3:on   4:on   5:on   6:off
sendmail       0:off  1:off  2:on   3:on   4:on   5:on   6:off
rhnssd         0:off  1:off  2:off  3:on   4:on   5:on   6:off
crond          0:off  1:off  2:on   3:on   4:on   5:on   6:off
yum            0:off  1:off  2:off  3:off  4:off  5:off  6:off
xinetd         0:off  1:off  2:off  3:on   4:on   5:on   6:off
xinetd based services:
  chargen-udp:  off
  rsync:        off
  chargen:      off
  daytime-udp:  off
  daytime:      off
  echo-udp:     off
  echo:         off
  services:     off
  time:         off
  time-udp:     off
  sgi_fam:      on
[root@server root]# xinetd_

```

大家看到 xinetd 这个服务 只要他是开启的，就可以运行他下面的服务器，它下面的大部分都是关闭的，只有一个开启的，如果我们想开启一个服务可以使用 chkconfig 命令，例如我们想开启 rsync 服务，我们可以使用 chkconfig rsync on 命令，然后可以使用 -list 查看。

21.11 基础结束

重新启动系统 xinetd，此时 xinetd 会发现需要启动 rsync 这个服务，如果我们要临时启动这个服务，我们可以使用 `service xinetd restart`。Service 启动脚本的名称一般是放在固定目录下面的 `/etc/init.d` 这个目录下面都有七个状态，运行上面的命令本质上就是运行 `/etc/init.d/xinetd` 这个脚本，它会检查自己的配置目录，是 `etc/xinetd.d` 这个目录这里面有如图所示

```
time:      off
time-udp:  off
sgi_fam:   on
root@server rootl# chkconfig rsync on
root@server rootl# chkconfig rsync --list
sync      on
root@server rootl# _
```

```
services:  off
time:      off
time-udp:  off
sgi_fam:   on
root@server rootl# chkconfig rsync on
root@server rootl# chkconfig rsync --list
sync      on
root@server rootl# service xinetd restart
topping xinetd: [ OK ]
tarting xinetd: [ OK ]
root@server rootl# ls /etc/init.d/xinetd
root@server rootl# ls /etc/xinetd.d
hargen      daytime      echo      rsync      sgi_fam    time-udp
hargen-udp  daytime-udp echo-udp   services   time
root@server rootl# _
```

系统的基本配置就讲到这。

22. 命令学习

22.1 命令学习一

安装完重启后，没有像 sery 所说在图形界面崩溃了，由于我没有安装 X-WINDOWS 而是直接进入了文本界面。如果你想做 linux 管理的话，最好在文本界面下工作，这样会适应如下图：

```
Red Hat Enterprise Linux AS release 4 (Nahant)
Kernel 2.6.9-5.ELsmp on an i686

qiuri login: _
```

第一行显示的是我们所安装的 linux 是 Red Hat 企业 4

第二行显示的是我们的内核版本，以及硬件等级是 i686

第三行显示的是我们的计算机名，也就是我们安装时设定的“qiuri”这时我们只需在 login 后输入用户名“root”即可跳转到输入密码选项，在 linux 中 root 的权限和 windows 中的 administrator 一样具有最大的权限，应该保护这个用户的密码以确保系统安全。

```
Red Hat Enterprise Linux AS release 4 (Nahant)
Kernel 2.6.9-5.ELsmp on an i686

qiuri login: root
Password:
Last login: Sun Oct 14 19:42:53 on tty1
[root@qiuri ~]# _
```

我们发现在 Password 后输入密码时密码不会被显示，不要以为是键盘坏掉了，这是 linux 一种安全措施，尤其使用小键盘输入时注意小键盘灯是否亮着。随后看到了最后登录详细时间，我们还看到一个 tty1 这个是 linux 中文字界面终端 1，在 linux 中一共有六个文字界面终端，使用 Ctrl+Alt+F1~F6 功能键相互切换文字界面终端。最后[root@qiuri ~]#其中 root 是当前用户，qiuri 是计算机名，~这个符号是当前用户的宿目录，最后的#号代表是提示符，一般在 linux 中 root 的命令提示符是#，普通用户的命令提示符是\$，同时命令提示符可以判断当前用户身份状态的依据。

对于 linux 新手来说，一般进入 linux 系统不知道自己该干些什么，今天给大家介绍一些 linux 中常用命令。

22.2 命令学习二

1.在 linux 中我们要知道 linux 系统中基本的命令格式如下：

命令字 【命令选项】 【命令参数】
Command 【option】 【arguments】

命令字就是命令的名称，在输入命令时一定要注意命令的真确性。

命令选项就是定义一条命令输出结果的参数，命令的不同命令的选项个数和内容也会不同。

命令参数就是命令要处理的对象，通常命令参数是文件名、目录名或者用户名等。

2. linux 中输入命令时注意事项

在 linux 中命令的输入过程中一般对于新手常犯的错误就是空格的使用，在输入命令时，命令字、命令选项、命令参数之间一定要有空格分开。在命令输入结束时要使用回车结束命令。

3. 基础命令使用案例解析

正如我们在安装时说过 linux 中所有的文件系统都挂接到根目录下相应的目录节点，那我们进入 linux 中第一件事就是验证一下这个结果，这就是我们要学习的第一条命令。

1. 查看文件名命令-ls 那我们就使用这条目录查看“/”下的目录

```
[root@qiuri ~]# ls /
bin  dev  home  lib  media  mnt  proc  sbin  srv  tmp  var
boot  etc  initrd  lost+found  misc  opt  root  selinux  sys  usr
```

这时大家看到了我们前面所说的目录，详细地罗列出来了。但是大家又会问我，你不是说过 linux 的命令格式是由命令字、命令选项、命令参数组成的。其实这个命令还有好多的命令选项和命令参数，由于个人水平有限在这里介绍几种给大家。

22.3 命令学习三

ls -a 列出所有文件，包括隐藏文件，还以查看“/”下目录为例：

```
[root@qiuri ~]# ls -a /
.  .bash_history  dev  initrd  media  opt  sbin  sys  var
.  bin  etc  lib  misc  proc  selinux  tmp
.  autofsck  boot  home  lost+found  mnt  root  srv  usr
```

ls -l 详细列出目录下文件的权限、所有者文件大小等详细信息，可以简单输入“ll”


```
[root@qiuri ~]# ls -l /
total 146
drwxr-xr-x  2 root root  4096 Oct 15 19:29 bin
drwxr-xr-x  4 root root 1024 Oct 12 09:44 boot
drwxr-xr-x  8 root root  5468 Oct 16 19:57 dev
drwxr-xr-x 65 root root  4096 Oct 16 19:57 etc
drwxr-xr-x  2 root root  4096 Aug 12 2004 home
drwxr-xr-x  2 root root  4096 Aug 12 2004 initrd
drwxr-xr-x 11 root root  4096 Oct 15 19:29 lib
drwx----- 2 root root 16384 Oct 12 05:41 lost+found
drwxr-xr-x  4 root root  4096 Oct 16 19:57 media
drwxr-xr-x  2 root root  4096 Dec  7 2004 misc
drwxr-xr-x  2 root root  4096 Aug 12 2004 mnt
drwxr-xr-x  2 root root  4096 Aug 12 2004 opt
dr-xr-xr-x 66 root root    0 Oct 16 15:55 proc
drwxr-xr-x  2 root root  4096 Oct 14 20:47 root
drwxr-xr-x  2 root root 12288 Oct 15 19:29sbin
drwxr-xr-x  1 root root    0 Oct 16 15:55 selinux
drwxr-xr-x  2 root root  4096 Aug 12 2004 srv
drwxr-xr-x  9 root root    0 Oct 16 15:55 sys
drwxrwxrwt  4 root root  4096 Oct 16 19:57 tmp
drwxr-xr-x 15 root root  4096 Oct 12 09:53 usr
drwxr-xr-x 19 root root  4096 Oct 12 09:45 var
```

ls-F 列出目录下文件名及其目录类型，以查看/etc 下文件为例：

```
[root@qiuri ~]# ls -F /etc
acpi/          group-         minicom.users  resolv.conf
adjtime        grub.conf@    modprobe.conf  rmt@
aliases        gshadow        modprobe.conf~  rpc
aliases.db     gssapi_mech.conf modprobe.conf.dist rpm/
alsa/          gtk-2.0/      mtd             samba/
alternatives/  hal/          mtab            screenrc
anacrontab     host.conf     mtools.conf     scsi_id.config
asound.state   hosts         Muttrc          securetty
at.deny        hosts.allow   netplug/        security/
auto.master    hosts.deny    netplug.d/      selinux/
auto.misc      hotplug/      nscd.conf       sensors.conf
auto.net*      hotplug.d/    nsswitch.conf   services
```

我们发现这时有的文件后有个*，有的文件有个@，还有的有个/结尾，他们分别代表是可执行文件、符号链接、和目录名。

22.4 命令学习四

由于时间关系不在这里一一举例，以下参数供有兴趣者参考：

ls -t 依照最后修改文件时间列出文件名
ls -R 列出当前目录和其子目录的文件名
ls -color=never 不以颜色方式显示
ls -color=always 均以颜色方式显示
ls -color=auto 系统自动判断

2.cd 切换用户当前目录

这条命令使用的语法是 cd 目录绝对路径或者相对路径

这里的绝对路径可以理解为我们每天乘坐公交车一样，假如我们从公交的第一站一直坐到最后一站，这就是一个绝对路径，但是假如有一天我们坐车到半路忽然想到我们忘记拿一件东西，这时我们可以选择就近下车再坐车返回去取，这就是相对路径，也可以使用绝对路径，现坐车到最后一站，再坐车回来去。在 linux 中也是这样在使用这两种路径的时候灵活多变的，通常根据实际情况进行选择。同样举几个例子：

例如，使用绝对路径进入“/etc”目录

```
[root@qiuri ~]# cd /etc
[root@qiuri etc]#
```

这时我们需要进入/dev 下我们就可以使用相对路径

```
[root@qiuri etc]# cd ../dev/
[root@qiuri dev]#
```

22.5 命令学习五

cd ~ 返回用户宿主目录

cd . 用户的当前目录

cd .. 返回上一级目录

3.pwd 查看用户当前所在位置

由于我们使用 cd 命令进入好多的目录这时我们不清楚我们所在的目录位置这时就可以使用 pwd 查看

```
[root@qiuri dev]# pwd
/dev
[root@qiuri dev]#
```

4.mkdir 创建目录

mkdir 可以单独建立目录 dir1

```
[root@qiuri ~]# mkdir dir1
[root@qiuri ~]# ls
anaconda-ks.cfg  dir1  install.log  install.log.syslog
[root@qiuri ~]#
```

mkdir 也可以同时建立多个目录

```
[root@qiuri ~]# mkdir dir2 dir3 dir4
[root@qiuri ~]# ls
anaconda-ks.cfg  dir1  dir2  dir3  dir4  install.log  install.log.syslog
[root@qiuri ~]#
```


22.6 命令学习六

使用绝对路径创建目录 test

```
[root@qiuri ~]# mkdir dir1/test
[root@qiuri ~]# ls dir1
test
[root@qiuri ~]# _
```

5.rmdir 删除目录

rmdir 删除目录 dir2

```
[root@qiuri ~]# rmdir dir2
[root@qiuri ~]# ls
anaconda-ks.cfg  dir1  dir3  dir4  install.log  install.log.syslog
[root@qiuri ~]# _
```

使用 rmdir 删除目录 dir3、dir4

```
[root@qiuri ~]# rmdir dir3 dir4
[root@qiuri ~]# ls
anaconda-ks.cfg  dir1  install.log  install.log.syslog
[root@qiuri ~]# _
```

试着删除 dir1, 这时我们发现提示我们“目录不是空的”所以我们要记住我们删除的目录必须是空的, 不能包含目录或文件, 如果要删除的话请先删除目录里面包含的内容再去删除目录

```
[root@qiuri ~]# rmdir dir1
rmdir: `dir1': Directory not empty
[root@qiuri ~]# _
```

目录有了, 那我们就不得不提一下创建文件的命令了。

22.7 命令学习七

6.touch 创建文件命令

使用 touch 命令在 dir1/test 下创建文件 test

```
[root@qiuri ~]# touch dir1/test/test
[root@qiuri ~]# ls -l dir1/test
total 4
-rw-r--r-- 1 root root 0 Oct 17 08:32 test
```

再次使用 touch 命令在 dir1/test 下创建文件 test, 查看有何变化

```
[root@qiuri ~]# touch dir1/test/test
[root@qiuri ~]# ls -l dir1/test
total 4
-rw-r--r-- 1 root root 0 Oct 17 08:35 test
```

经过我们的观察, 再次创建的文件名, 没有变化只是将时间属性更改了一下, 所以我们得出使用 touch 创建文件时, 如果文件不存在的话直接创建空文件, 如果存在只是更改一下时间属性。其实这条命令并不常用, 这是用于测试或实验当

中。

7.file 查看文件类型

我们使用 vi 编辑一下然后查看 dir1/test/test 文件类型

```
[root@qiuri ~]# file dir1/test/test
dir1/test/test: ASCII text
[root@qiuri ~]# file install.log
install.log: ASCII English text
[root@qiuri ~]#
```

File 命令能够识别 linux 系统中大多数文件类型，可识别文本文件。二进制可执行文件、压缩文件等。

22.8 命令学习八

8.cp 复制文件

cp 源文件 目标位置

```
[root@qiuri ~]# cp dir1/test/test /home
[root@qiuri ~]# ls /home
test
[root@qiuri ~]#
```

9.rm 删除文件

rm 是删除文件的命令，linux 中的文件删除是不可恢复的，使用这条命令时一定要小心。刚才使用 rmdir 删除 dir1 时，由于它包含子目录而没有删除掉，但是有了这条命令我们就不用去一个一个目录的去删除了，只要加上必要的选项，一切化简为零。

rm -r 循环删除直到删除没有

rm -f 强制删除

rm -rf 强制删除文件

```
[root@qiuri ~]# rm -rf dir1
[root@qiuri ~]# ls
anaconda-ks.cfg  install.log  install.log.syslog
[root@qiuri ~]#
```

22.9 先停

10.mv 移动文件或者重命名

将/home/test 移动到当前目录

```
[root@qiuri ~]# mv /home/test .  
[root@qiuri ~]# ls  
anaconda-ks.cfg  install.log  install.log.syslog  test  
[root@qiuri ~]#
```

利用 mv 重命名，就是说源文件和目标文件在同一目录就是文件重命名。

23. vi 编辑器

23.1 vi 编辑器学习一

说道命令，就不得不提这个特殊的命令“vi”它不光是命令，还是一个 linux 下的编辑器。由于时间关系，这里不在举例说明，将在以后配置网络服务时介绍一些 vi 的实际应用。

vi 命令是 linux 下的全屏幕文本编辑，vi 编辑器提供了丰富的编辑功能。这个编辑器十分简陋，对于初学者很不友好。但是在 linux 中 vi 编辑器作用非常大，以后尤其在配置各种服务器时修改配置文件时十分有用。

vi 编辑器有 3 种模式：命令模式、输入模式、末行模式。掌握这三种模式十分重要：

命令模式：vi 启动后默认进入的是命令模式，从这个模式使用命令可以切换到另外两种模式，同时无论在任何模式下只要按一下[Esc]键都可以返回命令模式。在命令模式中输入字母“i”就可以进入 vi 的输入模式编辑文件。

输入模式：在这个模式中我们可以编辑、修改、输入等编辑工作，在编辑器最后一行显示一个“--INSERT--”标志着 vi 进入了输入模式。当我们完成修改输入等操作的时候我们需要保存文件，这时我们需要先返回命令模式，在进入末行模式保存。

末行模式：在命令模式输入“:”即可进入该模式，在末行模式中有好多好用的命令。

1.vi 的启动和退出

直接输入命令 vi 不指定文件名，由于这是一个没有命名的空文件，显示了 vi 的版本信息。



23.2 vi 编辑器学习二

vim 是 vi 编辑器的改进版本，在 vi 编辑器的基础上扩展了很多实用的功能。大多数 linux/unix 版本都使用 vim 代替了原有的 vi 编辑器。

或者直接使用 vi 打开文件，如果文件存在打开存在文件；如果不存在的话，将以我们指定的参数作为文件名。

输入 q! 是强制退出，同时没有保存当前文件内容。如果文件没有进行修改，或者已经保存完毕也可以使用 q 退出。保存退出使用 wq，保存强制退出 wq!

2. 保存文件

如果文件已经有文件名，我们在末行模式直接输入 w 保存，如果没有文件名，输入 w 文件名进行保存。例如：w test 其中 test 是文件名。

那么在 vi 中也可以另存为，就是在保存完以后，我们再使用 w 另存为文件名。

3. 光标的移动

在命令模式中可以使用命令进行以下操作。

光标方向的移动，除了可以使用方向键，还可以使用以下命令

向上移动光标：k

向左移动光标：h

向右移动光标：l

向下移动光标：j

翻页命令

Ctrl+F 向前翻整页 Ctrl+U 向前翻半页

Ctrl+B 向后翻整页 Ctrl+D 向后翻半页

行内快速跳转

^移动到本行行首

\$移动到本行行尾

显示行号和取消行号(末行模式使用)

set nu 显示行号

set nonu 取消行号

在命令模式下, 使用以下命令可以快速的在行间跳转

1G 跳转到文件首行

G 跳转到尾行

#G 跳转到文件的#行

23.3 vi 编辑器学习三

4. 编辑操作

进入输入模式命令

i 插入命令 a 附加命令 o 打开命令 c 修改命令

r 取代命令 s 替换命令 Esc 退出命令

输入模式的操作

Home 光标到行首

End 光标到行尾

Page Up 和 Page Down 上下翻页

Delect 删除光标位置的字符

删除操作(命令模式使用)

x 删除光标处的单个字符

dd 删除光标所在行

dw 删除当前字符到单词尾包括空格的所有字符

#x 例如 3x 删除光标处向右的三个字符

#dd 例如 3dd 从当前行开始向下删除三行文本

撤销操作

u 命令取消最近一次的操作, 可以使用多次来恢复原有的操作

U 取消所有操作

Ctrl+R 可以恢复对使用 u 命令的操作

复制操作

yy 命令复制当前整行的内容到 vi 缓冲区

yw 复制当前光标所在位置到单词尾字符的内容到 vi 缓存区, 相当于复制一个单词

y\$复制光标所在位置到行尾内容到缓存区

y[^] 复制光标所在位置到行首内容到缓存区

#yy 例如：5yy 就是复制 5 行

#yw 例如：2yw 就是复制两个单词

如果要复制第 m 行到第 n 行之间的内容，可以在末行模式中输入 m, ny 例如：
3, 5y 复制第三行到第五行内容到缓存区。

23.4 vi 编辑器学习结束

5. 查找和替换

vi 的查找和替换功能主要在末行模式完成：

至上而下的查找

/ 要查找的字符串，其中/代表从光标所在位置起开始查找，例如：/ work

至下而上的查找

? 要查找的字符串 例如：/ work

替换

:s/old/new 用 new 替换行中首次出现的 old

: s/old/new/g 用 new 替换行中所有出现的 old

:#, # s/old/new/g 用 new 替换从第 # 行到第 # 行中出现的 old

: % s/old/new/g 用 new 替换整篇中出现的 old

如果替换的范围较大时，在所有的命令尾加一个 c 命令，强制每个替换需要用户进行确认，例如:s/old/new/c 或 s/old/new/gc

6 恢复文件

vi 在编辑某一个文件时，会生成一个临时文件，这个文件以 . 开头并以 .swp 结尾。正常退出该文件自动删除，如果意外退出例如忽然断电，该文件不会删除，我们在下次编辑时可以选择一下命令处理：

O 只读打开，不改变文件内容

E 继续编辑文件，不恢复 .swp 文件保存的内容

R 将恢复上次编辑以后未保存文件内容

Q 退出 vi

D 删除 .swp 文件

或者使用 vi -r 文件名来恢复未保存的内容

vi 命令大致给大家介绍这些，其实它的功能很强大的，我们在日常实际使用过程中不断积累命令的使用方法，相信大家会很快掌握这个命令的。

24. 回到命令学习

24.1 回到命令学习一

由于我们已经可以使用 vi 编辑自己的文件，如果我们想查看这些文件，总不能每次都使用 vi 打开文件去查看吧。所以首先给大家介绍几条关于文件查看命令。

1. 文件查看命令

cat 命令

这条命令适用于最简单的文本查看，它在显示文件内容的时候是以第一行开始，在显示过程中没有停顿，文件较长的话，只能看到最后一页内容在屏幕上显示，不适合查看长文件。

例如：我们查看“/etc”目录下的“inittab”文件，直接在命令提示符下输入：

```
[root@qiuri ~]# cat /etc/inittab
```

由于这个文件较长，我们只看到了文件的最后一页在屏幕上的显示。如果在 cat /etc/inittab 后加上 -n 将在显示的结果中显示行号。

```
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# When our UPS tells us power has failed, assume we have a few minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"

# If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Run xdm in runlevel 5
x:5:respawn:/etc/X11/prefdm -nodaemon
[root@qiuri ~]#
```

24.2 回到命令学习二

接下来这个命令比较有意思，写法是将 `cat` 反过来些，并且它和 `cat` 命令的功能刚好相反，是以文件的最后一行开始到文件的第一行反向的显示到屏幕。它就是 `tac`，用法也和 `cat` 一样，这里不做详细介绍。

more 命令

当文件很大的时候 `more` 命令相比 `cat` 命令要好用的多，它的用法也是 `more [文件名]`，它的显示方法是分屏显示文件内容。

例如：同样还是“/etc”目录下的“`inittab`”文件，也是在命令提示符下输入：

```
[root@qiuri ~]# more /etc/inittab
#
# inittab      This file describes how the INIT process should set up
#              the system in a certain run-level.
#
# Author:      Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#              Modified for RHS Linux by Marc Ewing and Donnie Barnes
#
# Default runlevel. The runlevels used by RHS are:
#  0 - halt (Do NOT set initdefault to this)
#  1 - Single user mode
#  2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#  3 - Full multiuser mode
#  4 - unused
#  5 - X11
#  6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault:
# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
l0:0:wait:/etc/rc.d/rc 0
--More--(48%)
```

我们看到这个命令的确是分屏显示，在屏幕的最下方还显示出当前显示内容在整个文件中的百分比。那么我们怎样可以看到 40% 以下的内容，这里介绍几个常用的操作按键，如果想了解详细使用方法，在分屏显示界面中输入“`h`”会显示 `more` 命令的帮助信息。

输入空格将显示下一屏的文件内容，如果文件已经显示到文件尾，`more` 命令会自动退出。

输入字母 `b` 将显示上一屏的文件内容，如果显示已经在文件头部，则屏幕显示文件保持不变。

使用回车键可以将文件内容向上滚动一行，也就是显示一行新的内容到当前屏幕。

输入 `q` 或 `Q` 退出 `more` 命令阅读环境。

总结，这条命令可以分屏显示内容和翻页功能，较适合阅读较长的文件。

24.3 回到命令学习三

less 命令

less 命令和 more 的功能几乎差不多，只是多了一些功能。更适用阅读较长较大的文件。使用方法也是 less [文件名]。

还是以查看“/etc”目录下的“inittab 文件，也是在命令提示符下输入：

```
[root@qiuri ~]# less /etc/inittab
```

less 也是以全屏幕的显示文件，最后一行显示出了当前阅读文件的名称。

```
#
# inittab          This file describes how the INIT process should set up
#                  the system in a certain run-level.
#
# Author:          Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#                  Modified for RBS Linux by Marc Ewing and Donnie Barnes
#
# Default runlevel. The runlevels used by RBS are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault:
# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
/etc/inittab
```

less 命令中的“回车键、空格键和 b”操作键功能和 more 的相同，还可以使用 Page Up 和 Page Down 进行上下翻页，使用上下方向键对文本内容一行一行的上下移动，退出命令和 more 的也一样。还可以查找字符串，输入/要查找的字符串。

24.4 回到命令学习四

head 和 tail 命令

这两条命令可以查看文件的前几行或后几行，两条命令都是默认显示文件内容 10 行，但是 head 是显示最前 10 行，而 tail 是最后 10 行。

head 和 tail 命令中添加选项“-n”，可以设置查看最前 n 行或最后 n 行，例如查看“/etc”目录下的“inittab 文件中的前 3 行使用命令如下：

```
[root@qiuri ~]# head -3 /etc/inittab
#
# inittab          This file describes how the INIT process should set up
#                  the system in a certain run-level.
[root@qiuri ~]# _
```

nl 命令

nl 命令查看文件的同时可以显示行号，用法和 cat -n 相似。

```
[root@qiuri ~]# nl /etc/inittab_
28 ca::ctrlaltdel:/sbin/shutdown -t3 -r now
29 # When our UPS tells us power has failed, assume we have a few minutes
30 # of power left.  Schedule a shutdown for 2 minutes from now.
31 # This does, of course, assume you have powerd installed and your
32 # UPS connected and working correctly.
33 pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Do
wn"
34 # If power was restored before the shutdown kicked in, cancel it.
35 pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancell
ed"
36 # Run gettys in standard runlevels
37 1:2345:respawn:/sbin/mingetty tty1
38 2:2345:respawn:/sbin/mingetty tty2
39 3:2345:respawn:/sbin/mingetty tty3
40 4:2345:respawn:/sbin/mingetty tty4
41 5:2345:respawn:/sbin/mingetty tty5
42 6:2345:respawn:/sbin/mingetty tty6
43 # Run xdm in runlevel 5
44 x:5:respawn:/etc/X11/prefdm -nodaemon
[root@qiuri ~]#
```

24.5 回到命令学习五

2.文件或目录的查找

find 命令

find 命令的在查找文件或目录的功能非常强大，可以根据文件的大多数属性来查找文件，它的使用形式也是多变的，基本命令格式：

find [path] [expression] 使用这条命令最常用的是按照文件名查找，文件名表达式的格式为：-name 文件名，其中文件名可以使用*或? 通配符进行表示。符合条件的将被显示出来。

例如：查找“inittab”文件，可以使用 find / -name inittab

```
[root@qiuri ~]# find / -name inittab
/etc/inittab
[root@qiuri ~]# _
```

我们在查找的过程中会发现它的查找速度相当的慢，而且这条命令十分耗硬盘。所以这条命令虽然功能十分的强大，但是最好先使用其它命令查找不到的时候使用。

其它可选选项：

- adminN 在过去 N 分钟内读过的文件
- anewerFILE 比文件 FILE 更晚读取的文件
- cnewer FILE 比文件 FILE 更新的文件
- atime[±]N 在过去 N 天以内(-)或以外(+)读取过的文件
- cmin[±]N 在过去 N 分钟以内(-)或以外(+)修改过的文件
- ctime[±]N 在过去 N 天以内(-)或以外(+)修改过的文件
- empty 空文件
- gN 组 id 号是 N 的文件
- groupNAME 组名是 NAME 的文件
- ipath P,path P 路径名符合 P 的文件，ipath 会忽略大小写
- name NAME,-iname NAME 文件名称符合 NAME 的，iname 会忽略大小写。
- sizeN[b|c|k|w] 文件大小是 N，单位 b 代表 512 位的块，c 表示一个 char，k 表示 kilobytes，w 是一个 word。
- typeT 文件类型是 T 的文件，文件类型有 d 目录、c 字符设备文件、b 块设备文件、f 普通文件、l 符号链接、s 套接字文件。
- mount,-xdev 只检查和指定目录在同一个文件系统中的文件，避免列出其它文件系统中的文件。

24.6 回到命令学习六

which 命令

which 的基本功能是通过环境变量 **PATH** 到该路径内寻找可执行文件，所以用于查找可执行文件。其实这个时候想一想，怎样可以看出文件是可执行文件，我在前面已经介绍了一种方法，学习就是结合起来运用。在我们还不熟悉这个系统的时候，我们可以先在某个目录查看一下是否有可执行文件，然后在使用 **which** 验证。

whereis 命令

whereis 可以迅速的找到文件，而且还提供了这个文件的二进制可执行文件、源代码文件和使用手册页存放位置。

例如：查找 **inittab** 文件

```
[root@qiuri ~]# whereis inittab
inittab: /etc/inittab /usr/share/man/man5/inittab.5.gz
[root@qiuri ~]# _
```

我们可以看到 **inittab** 是存放在子目录/etc 下，而且他的使用手册在 /usr/share/man/man5 下的 inittab.5.gz

3.硬盘装载命令

mount 命令

mount 命令用于物理设备(例如: 光盘、U 盘、硬盘)中的文件系统挂载到 linux 文件系统的某个目录中, 在 mount 命令不使用任何选项和参数的时候将显示当前 linux 系统中以挂载的文件系统信息。首先介绍光盘的挂载方法:

```
[root@qiuri ~]# mount
/dev/sda5 on / type ext3 (rw)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
usbfs on /proc/bus/usb type usbfs (rw)
/dev/sda1 on /boot type ext3 (rw)
none on /dev/shm type tmpfs (rw)
/dev/sda2 on /usr type ext3 (rw)
/dev/sda3 on /var type ext3 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
[root@qiuri ~]#
```

24.7 回到命令学习七

mount 命令的功能强大, 命令格式也比较丰富。但是常用挂载文件系统命令格式: `mount -t type dev dir`

其中 `-t type` 选项指出需要挂载的文件系统类型, 光盘文件系统类型是: `iso9660`; `dev` 表示需要挂载文件系统的设备名称, 光盘驱动器的设备名称是 `/dev/cdrom`; `dir` 表示挂载点, 即挂载到的文件目录路径。光盘设备在 linux as4 中系统中的默认路径是 `/media/cdrom`。

```
[root@qiuri ~]# ll /media/cdrom
total 0
[root@qiuri ~]# mount -t iso9660 /dev/cdrom /media/cdrom
mount: block device /dev/cdrom is write-protected, mounting read-only
[root@qiuri ~]# mount
/dev/sda5 on / type ext3 (rw)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
usbfs on /proc/bus/usb type usbfs (rw)
/dev/sda1 on /boot type ext3 (rw)
none on /dev/shm type tmpfs (rw)
/dev/sda2 on /usr type ext3 (rw)
/dev/sda3 on /var type ext3 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/hdc on /media/cdrom type iso9660 (ro)
[root@qiuri ~]#
```

由于光盘是只读的存储介质, 因此 mount 命令在挂载光盘的时候会提示光盘一只读(read-only)模式进行挂载, 挂载后使用 mount 命令查看已挂载的文件系统, 可以在最后一行看到设备的挂载信息, 证明挂载成功。

读取光盘中的内容非常的简单了, 使用 mount 命令挂载后, 可以访问挂载目录完成对光盘内容的读取。

```
[root@qiuri ~]# ll /media/cdrom
total 1183
dr-xr-xr-x 1 root root 1418 Aug 1 2006 ????
-r-xr-xr-x 1 root root 1286859 Aug 1 2006 autorun.exe
-r-xr-xr-x 1 root root 47 Aug 1 2006 autorun.inf
dr-xr-xr-x 1 root root 244 Aug 1 2006 documents and settings
dr-xr-xr-x 1 root root 918 Aug 1 2006 pcxp
dr-xr-xr-x 1 root root 1278 Aug 1 2006 programs
```

24.8 回到命令学习八

umount 命令

umount 命令用于卸载已经挂载的文件系统，基本格式如：umount dir device [...] 使用 umount 卸载文件系统可以指定挂载设备文件名或挂载目录作为参数，可以将卸载指定的设备或挂载点目录上的文件系统。

对于光盘文件系统的卸载可以使用，以下两条命令中的任意一条：注意在卸载的时候要退出光盘的挂载目录，否则提示文件系统在使用。

```
[root@qiuri cdrom]# pwd
/media/cdrom
[root@qiuri cdrom]# umount /media/cdrom 卸载命令之一
umount: /media/cdrom: device is busy
umount: /media/cdrom: device is busy 错误提示信息
[root@qiuri cdrom]# cd ~
[root@qiuri ~]# umount /media/cdrom
[root@qiuri ~]#
[root@qiuri ~]# umount /dev/cdrom 另一条卸载命令
```

eject 命令

使用 eject 命令可以将光盘驱动器中的光盘轻轻弹出和收回，不必使用光盘驱动器的弹出按钮。

```
[root@qiuri ~]# eject 弹出光盘命令
[root@qiuri ~]# eject -t 光盘驱动器自动收回功能，需要硬件支持，多数pc上的光盘驱动器都有。笔记本的光盘
[root@qiuri ~]# _
```

24.9 回到命令学习九

U 盘的使用方法

在 linux 系统中，USB 设备被视为 SCSI 设备，使用这个设备文件的格式为“sdX”系统中的第一个 SCSI 设备表示为“sda”第二个表示为“sdb”由于我使用的系统硬盘是串口硬盘，所以我的 U 盘被识别为“sdb”

U 盘设备插入系统后每次分配的设备文件名是不固定的，依赖于当前系统拥

有几个 SCSI 设备，在不确定设备名称时，我们可以使用“fdisk”命令，列出系统中所有存储设备：

```
[root@qiuri ~]# fdisk -l

Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1           16       128488+   83  Linux
/dev/sda2                17          340       2602530   83  Linux
/dev/sda3            341          595       2048287+   83  Linux
/dev/sda4            596         1044       3606592+    5  Extended
/dev/sda5            596          722       1020096   83  Linux
/dev/sda6            723          787        522081   82  Linux swap

Disk /dev/sdb: 1008 MB, 1008730112 bytes
16 heads, 32 sectors/track, 3848 cylinders
Units = cylinders of 512 * 512 = 262144 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1  *           1         3848        985072    b  W95 FAT32

[root@qiuri ~]# _
```

我们可以看出当前设备文件名是“/dev/sdb”，U 盘上唯一的主分区设备文件的名称是“/dev/sdb1”，分区中的文件系统类型是 fat。知道这些以后我们就可以使用 mount 命令挂载 U 盘到当前文件系统。“/mnt”目录通常用于挂载系统中临时使用的文件系统，可以将“/mnt”目录作为 U 盘系统的挂接点。使用“vfat”文件系统类型表示所有的 fat 文件系统类型，包括 fat16 和 fat32，ntfs 还是使用 ntfs 表示。

```
[root@qiuri ~]# mount -t vfat /dev/sdb1 /mnt/
[root@qiuri ~]# mount -t ntfs /dev/sdb1 /mnt/_
```

24.10 结束

U 盘的卸载

```
[root@qiuri ~]# umount /dev/sdb1
[root@qiuri ~]# _
```

其实，在 linux 系统中还有好多好多的命令，比如磁盘管理、用户管理的命令等等，我将他们总结成了一个单一模块来记忆。让我们慢慢的了解和记忆，能够掌握和运用才是我们最终的目的。