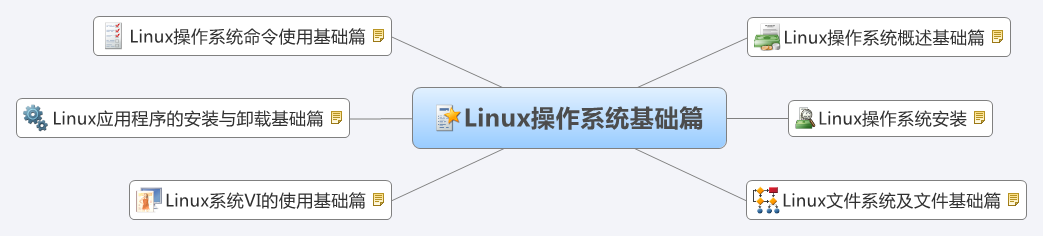
Linux操作系统基础篇



# Linux操作系统概述基础篇

**课程目标**

了解Linux系统的发展

理解Linux系统的特性

掌握Linux系统的组成结构

**重点难点**

Linux系统的组成结构

**考核目标**

       Linux系统的组成结构

**课后练习**

**小贴士/名词解释**

kernel

shell,Bash,Csh

## Linux的发展、组成结构和特性

**发展**

1991 Linus  Linux  0.0.1版  8000行  内核最新版本4.1.4

加入GNU  基于Linux的遵循GPL协议（允许开源、分享传播、修改）非商业用途  GPL3

基于内核之上的Linux操作系统版本至少有300种以上

**组成结构**

**kernel：内核，包围在硬件之上**

1）作用

           管理设备、进程、内存、交换空间、文件系统、精灵进程等；

           控制系统程序与硬件之间打交道的功能；

**shell：外壳，包裹在内核之上**

       1）作用

           提供人机交互界面，用户与内核之间打交道的功能，类似           Windows的cmd

       2）分类：不同shell，命令解释不一样

           Bourne shell(sh)，Korn shell(ksh)，C shell(csh)，           Bourne-again shell(bash)，tcsh

       3）命令提示符：不同shell的命令提示符不一样

           sh,Bash：#root   $user

           csh：#root   %user

**FileSystem：文件系统，管理文件、目录**

**Linux特性**

       多用户，多任务（进程、线程处理），多平台，图形化界面             （x-window）硬件支持（低配置），通信与联网，应用程序支持（编辑器、编程工具、数据库、办公软件、图形处理       、Internet应用、游戏）

# Linux操作系统安装

**课程目标**

       了解Linux系统的两个版本及选择

了解ubuntu的不同版本

理解Linux系统的登陆

掌握vmware和ubuntu官网使用并下载软件

掌握ubuntu的桌面和服务器版的安装

**重点难点**

       ubuntu的桌面和服务器版的安装

**考核目标**

       ubuntu的桌面和服务器版的安装

**课后练习**

安装ubuntu Desktop和Server或者安装Redhat

**小贴士/名词解释**

## Linux的选择与安装准备

**Linux系统的选择**

       red hat：企业级版的，已经开始收费

       ubuntu：桌面开源的，比较纯正的Linux，android官方指定的编译操作       系统，发展快，已支持ARM架构

**安装前准备**

**硬件准备**

           现在的PC都可以满足

**软件准备**

           vmware：http://www.vmware.com/cn      vmware player，           vmwareworkstaticon收费

           ubuntu

               1）ubuntu的版本：最新14.04.3   http://www.ubuntu.com

                  ubuntuDesktop：桌面版，图形界面

                  ubuntuServer：服务器版

                  cloud infrastructure:云版本

               2）安装方式

                  windows install，CD，USB

                3）下载桌面和服务器版：iso安装镜像文件

## 安装ubuntu Desktop & Server

安装vmware workstation

在vmware中新建和配置ubuntu系统

图形模式与命令模式的切换：

ctrl+Alt+F2,F3,F4,F6:图形转命令

ctrl+Alt+F7：命令转图形

# Linux文件系统及文件基础篇

**课程目标**

       了解文件系统的基本定义和功能

了解Linux文件系统

理解不同类别的文件系统及其作用

理解文件系统的基本组成

理解Linux文件系统的特点

掌握Linux文件系统的目录结构内容、特点和作用

       掌握Linux系统的登录和使用ls查看目录文件

**重点难点**

不同类别的文件系统及其作用

Linux文件系统的目录结构内容、特点和作用

       Linux系统的登录和使用ls查看目录文件

**考核目标**

       不同类别的文件系统及其作用

Linux文件系统的目录结构内容、特点和作用

       Linux系统的登录和使用ls查看目录文件

**课后练习**

练习登陆和退出ubuntu系统

       练习ls查看目录文件

       使用ls命令查看ubuntu系统的文件和目录结构

       通过不同的文件标识和颜色识别不同文件类别

**小贴士/名词解释**

 EXT3，JFFS2，YAFFS

BFFS，WINFS，NFS

VFS（Proc），iNode区

## 文件系统概述

### 文件系统定义、分类和功能

**文件系统定义**

文件系统是一种组织计算机文件和资料的方法；

操作系统中封装的系统服务程序，实际上是一个软件程序，用来存储和管理计算机文件和资料；

**文件系统分类**

磁盘文件系统：NTFS，EXT3

闪存文件系统：JFFS2，YAFFS

数据库文件系统：BFFS，WINFS

网络文件系统：NFS

虚拟文件系统：VFS（Proc）

**文件系统功能**

能定义文件的组织方式：文件结构

提供建立和存取文件的环境：目录和文件

能对文件存储器空间进行组织和分配

负责文件存储并对存入的文件进行保护和检索

负责建立文件，存入、读出、修改、转储文件，控制文件的存取，撤销文件等

## Linux文件系统及文件

### Linux文件系统及特点

**Linux文件系统**

**EXT3**

       EXT3是基于日志方式的文件系统；

       系统中每个文件都是有索引，用户对对文件的每一个操作都会记录日志，       形成一个任务队列排着执行

       性能是比较好

**SWAP**

       swap是交换分区的文件系统，类似windows的虚拟内存

       虚拟内存的实现：两种方式

           第一种是进行内存的排列像内存池一样，进行一个优化

           第二种是把硬盘上的空间模拟成内存

       swap是Linux的虚拟内存，在安装时要设好大小，是物理内存的2倍

**Linux文件系统特点**

**Linux系统中一切皆文件**

       Linux系统把设备(硬盘、软驱、光驱等)都看作文件，文件夹也看作文件

**Linux文件类型**

普通文件：- 目录文件：d 链接文件：| 块设备：b 字符设备：c

Socket: s 管道文件：p

**Linux文件属性**

**蓝色：目录**

**绿色：可执行**

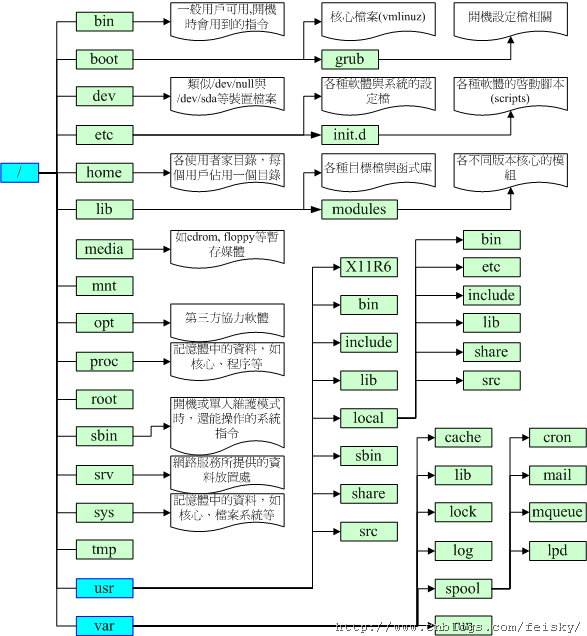
**浅蓝色：链接**

**红色：压缩**

**灰色：其他**

### Linux文件系统目录结构

**Linux文件系统的目录结构**

****

* /bin 基础系统所需要的那些命令位于此目录，也是最小系统所需要的命令；比如 ls、cp、mkdir等命令；功能和/usr/bin类似，这个目录中的文件都是可执行的，普通用户都可以使用的命令。做为基础系统所需要的最基础的命令就是放在这里。
* /boot Linux的内核及引导系统程序所需要的文件，比如 vmlinuz initrd.img 文件都位于这个目录中。在一般情况下，GRUB或LILO系统引导管理器也位于这个目录；
* /dev 设备文件存储目录，比如声卡、磁盘... ...
* /etc 系统配置文件的所在地，一些服务器的配置文件也在这里；比如用户帐号及密码配置文件；
* /home 普通用户家目录默认存放目录；
* /lib 库文件存放目录
* /lost+found 在ext2或ext3文件系统中，当系统意外崩溃或机器意外关机，而产生一些文件碎片放在这里。当系统启动的过程中fsck工具会检查这里，并修复已经损坏的文件系统。 有时系统发生问题，有很多的文件被移到这个目录中，可能会用手工的方式来修复，或移到文件到原来的位置上。
* /media 即插即用型存储设备的挂载点自动在这个目录下创建，比如USB盘系统自动挂载后，会在这个目录下产生一个目录 ；CDROM/DVD自动挂载后，也会在这个目录中创建一个目录，类似cdrom 的目录。这个只有在最新的发行套件上才有，比如Fedora Core 4.0 5.0 等。可以参看/etc/fstab的定义；
* /misc
* /mnt 这个目录一般是用于存放挂载储存设备的挂载目录的，比如有cdrom 等目录。可以参看/etc/fstab的定义。有时我们可以把让系统开机自动挂载文件系统，把挂载点放在这里也是可以的。主要看/etc/fstab中怎么定义了；比如光驱可以挂载到/mnt/cdrom 。
* /opt 表示的是可选择的意思，有些软件包也会被安装在这里，也就是自定义软件包，比如在Fedora Core 5.0中，OpenOffice就是安装在这里。有些我们自己编译的软件包，就可以安装在这个目录中；通过源码包安装的软件，可以通过 ./configure --prefix=/opt/目录 。
* /proc 操作系统运行时，进程（正在运行中的程序）信息及内核信息（比如cpu、硬盘分区、内存信息等）存放在这里。/proc目录伪装的文件系统proc的挂载目录，proc并不是真正的文件系统，它的定义可以参见 /etc/fstab 。
* /root Linux超级权限用户root的家目录；
* /sbin 大多是涉及系统管理的命令的存放，是超级权限用户root的可执行命令存放地，普通用户无权限执行这个目录下的命令，这个目录和/usr/sbin;/usr/X11R6/sbin或/usr/local/sbin目录是相似的； 我们记住就行了，凡是目录sbin中包含的都是root权限才能执行的。
* /tmp 临时文件目录，有时用户运行程序的时候，会产生临时文件。/tmp就用来存放临时文件的。/var/tmp目录和这个目录相似。
* /usr 这个是系统存放程序的目录，比如命令、帮助文件等。这个目录下有很多的文件和目录。当我们安装一个Linux发行版官方提供的软件包时，大多安装在这里。如果有涉及服务器配置文件的，会把配置文件安装在/etc目录中。/usr目录下包括涉及字体目录/usr/share/fonts ，帮助目录 /usr/share/man或/usr/share/doc，普通用户可执行文件目录/usr/bin 或/usr/local/bin 或/usr/X11R6/bin ，超级权限用户root的可执行命令存放目录，比如 /usr/sbin 或/usr/X11R6/sbin 或/usr/local/sbin 等；还有程序的头文件存放目录/usr/include。
* /var 这个目录的内容是经常变动的，看名字就知道，我们可以理解为vary的缩写，/var下有/var/log 这是用来存放系统日志的目录。/var/www目录是定义Apache服务器站点存放目录；/var/lib 用来存放一些库文件，比如MySQL的，以及MySQL数据库的的存放地；

我们再补充一下一些比较重要的目录的用途；

* /etc/init.d 这个目录是用来存放系统或服务器以System V模式启动的脚本，这在以System V模式启动或初始化的系统中常见。比如Fedora/RedHat；
* /etc/xinit.d 如果服务器是通过xinetd模式运行的，它的脚本要放在这个目录下。有些系统没有这个目录， 比如Slackware，有些老的版本也没有。在Rehat/Fedora中比较新的版本中存在。
* /etc/rc.d 这是Slackware发行版有的一个目录，是BSD方式启动脚本的存放地；比如定义网卡，服务器开启脚本等。
* /etc/X11 是X-Windows相关的配置文件存放地；

比如下面的例子：

[root@localhost ~]# /etc/init.d/sshd start   注：启动sshd服务器

[root@localhost ~]# /etc/init.d/sshd stop 注：停止sshd服务器

启动 sshd：                                                [确定]

这就是典型的sshd 服务器 System V模式启动脚本，通过这运行这个脚本可以启动sshd服务器了。

* /usr/bin 这个目录是可执行程序的目录，普通用户就有权限执行； 当我们从系统自带的软件包安装一个程序时，他的可执行文件大多会放在这个目录。比如安装gaim软件包时。相似的目录是/usr/local/bin； 有时/usr/bin中的文件是/usr/local/bin的链接文件；
* /usr/sbin 这个目录也是可执行程序的目录，但大多存放涉及系统管理的命令。只有root权限才能执行；相似目录是/sbin 或/usr/local/sbin或/usr/X11R6/sbin等；
* /usr/local 这个目录一般是用来存放用户自编译安装软件的存放目录；一般是通过源码包安装的软件，如果没有特别指定安装目录的话，一般是安装在这个目录中。这个目录下面有子目录。自己看看吧。
* /usr/lib 和/lib 目录相似，是库文件的存储目录；
* /usr/share 系统共用的东西存放地，比如 /usr/share/fonts 是字体目录，是用户都共用的吧。
* /usr/share/doc和/usr/share/man帮助文件，也是共用的吧；
* /usr/src 是内核源码存放的目录，比如下面有内核源码目录，比如 linux 、linux-2.xxx.xx 目录等。有的系统也会把源码软件包安装在这里。比如Fedora/Redhat，当我们安装file.src.rpm的时候，这些软件包会安装在/usr/src/redhat相应的目录中。另外Fedhat 4.0 5.0，他的内核源码包的目录位于/usr/src/kernels目录下的某个目录中（只有安装后才会生成相应目录）；
* /var/adm 比如软件包安装信息、日志、管理信息等，在Slackware操作系统中是有这个目录的。在Fedora中好象没有；自己看看吧。
* /var/log 系统日志存放，分析日志要看这个目录的东西；
* /var/spool 打印机、邮件、代理服务器等假脱机目录；

**Linux与Windows目录结构的区别**

**根目录**

Linux：/；Windows：\

**命名大小写区分**

Linux：命名区分大小写；Windows：命名不区分大小写

**结构管理**

           Linux：磁盘逻辑结构管理物理结构，格式化将磁盘分为很多的文件块区

           Windows：物理结构管理逻结构，先分区再格式化建立结构

# Linux操作系统命令使用基础篇

**课程目标**

       熟悉shell的特殊字符的使用

       熟练掌握shell常用命令的使用

       掌握shell命令自我学习方法

**重点难点**

       shell常用命令的使用

**考核目标**

**课后练习**

 把课堂的内容演练一遍

**小贴士/名词解释**

**推荐书籍**

《Linux命令大全》

## shell命令基础

### 命令行编辑与别名使用

**命令行格式**

**$command     [option(s)]     [argument(s)]**

**命令名   空格      选项     空格      参数**

    在命令行每两个部分之间有空格分隔

    每个命令行可使用的最多的命令字符数是256个

    命令要区分大小写

    不同的命令提示符：使用的分隔符为/

    命令中的参数、选项可以是多个，并且参数其实就是要传入命令程序主函数main的参数

**命令行编辑**

**命令行别名使用**

       alias alias-name=‘value ’建立别名

       alias 列出已经建立的别名列表

       unalias取消别名

### shell的特殊字符

**shell的特殊字符**

**通配符：\*,?,[]**

       1）\*：通配0个或多个字符；

ls \*m ls \*b ls \*t

       2）?：通配任意单个字符；

ls d? ls r??

       3）[s]：通配某个范围内的任意一个字符

    ls [b-f]b\* ls [fF]\*

**一行执行多条命令：;**

pwd;ls cd / ; ls cd root;ls cd home;ls

**输入输出重定向：>,<,>>**

1）输出重定向：>,>>

               >：将一个命令的输出放入文件而非屏幕

pwd>ls.txt pwd>pwd.txt ls>ls.txt

               >>：输出重定向但不会把原文件覆盖，在原文件末尾追加

Pwd>>ls.txt

        2）输入重定向：<    从标准输入设备键盘输入

 cat<ls.txt>ls2.txt ls<ls2.txt cat<ls.txt

**管道符：|**

将一个进程的输出作为另一个进程的输入

ls /bin |more ls –l /bin > binlist.txt cp /dev/stdin ml.txt cat/dev/stdin cat /dev/stdout

**其他：%,$,~**

%：作业控制，提示符等

       $：取某一列的值，取变量值等

~:用户主目录

* 使用以下代码输出图形字符
* $ sudo apt-get update;sudo apt-get install sysvbanner toilet figlet
* $ banner feels good
* $ toilet feels good
* $ figlet feels good

1. 物理机系统上可以通过使用[Ctrl]+[Alt]+[F1]～[F6]进行切换登陆，当切换到其中一个终端后想要切换回图形界面，可以按下[Ctrl]+[Alt]+[F7]来完成
2. echo “”可以用来打印字符串
3. tab键可以补全命令，补全目录，补全命令参数
4. Ctrl+c键可以强行终止当前程序
5. tail可以进入显示输入内容的界面并且不会执行输入的命令
6. find可以用来查询内容
7. Ctrl+d 键盘输入结束或退出终端 Ctrl+s 暂定当前程序，暂停后按下任意键恢复运行 Ctrl+z 将当前程序放到后台运行，恢复到前台为命令fg Ctrl+a 将光标移至输入行头，相当于Home键 Ctrl+e 将光标移至输入行末，相当于End键 Ctrl+k 删除从光标所在位置到行末 Alt+Backspace 向前删除一个单词 Shift+PgUp 将终端显示向上滚动 Shift+PgDn 将终端显示向下滚动
8. 使用键盘上的方向上键，恢复你之前输入过的命令
9. 通配符是一种特殊语句，主要有星号（\*）和问号（?），用来对对字符串进行模糊匹配（比如文件名，参数名）。当查找文件夹时，可以使用它来代替一个或多个真正字符
10. touch 可用来创建文件，用touch No.{1..10}.txt 可创建10个连续命名的文件
11. 使用man来查看手册，--help来查看帮助

普通意义上的 Shell 就是可以接受用户输入命令的程序。它之所以被称作 Shell 是因为它隐藏了操作系统底层的细节。Shell 既是用户交互的界面，也是控制系统的脚本语言.常见的 Shell 有 bash，zsh，ksh，csh 等等

**1. 快捷键**

* Tab ---------补全
* Ctrl+c ------强行终止当前程序
* Ctrl+d ------键盘输入结束或退出终端
* Ctrl+s ------暂定当前程序，暂停后按下任意键恢复运行
* Ctrl+z ------将当前程序放到后台运行，恢复到前台为命令fg
* Ctrl+a ------将光标移至输入行头，相当于Home键
* Ctrl+e ------将光标移至输入行末，相当于End键
* Ctrl+k ------删除从光标所在位置到行末
* Alt+Backspace ------向前删除一个单词
* Shift+PgUp ------将终端显示向上滚动
* Shift+PgDn ------将终端显示向下滚动
* up

**2. 最常用的帮助命令**

* man [数字] 命令
* 命令 --help
* ls
* find 关键词

**3. 通配符**

* + -------匹配 0 或多个字符
* ? ------匹配任意一个字符
* [list] ------匹配 list 中的任意单一字符
* [!list] ------匹配 除list 中的任意单一字符以外的字符
* [c1-c2] ------匹配 c1-c2 中的任意单一字符 如：[0-9] [a-z]
* {string1,string2,...} ------匹配 sring1 或 string2 (或更多)其一字符串
* {c2..c2} ------匹配 c1-c2 中全部字符 如{1..10]

**4. 一个好玩的字符显示**

* sudo apt-get update;sudo apt-get install sysvbanner
* banner shiyanlou
* printerbanner -w 50 A

[环境变量与文件查找](https://www.shiyanlou.com/courses/1)

**变量定义**

#定义变量

declare $变量名

#或者直接赋值也自动定义变量,注意变量之前不带"$"

变量名=值

#打印变量的值,"$"代表引用变量的值

echo $变量名

在所有的 UNIX 和类 UNIX 系统中，每个进程都有其各自的环境变量设置，且默认情况下，当一个进程被创建时，处理创建过程中明确指定的话，它将继承其父进程的绝大部分环境设置。

通常我们会涉及到的环境变量有三种：

* 当前 Shell 进程私有用户自定义变量，如上面我们创建的 temp 变量，只在当前 Shell 中有效。
* Shell 本身内建的变量。
* 从自定义变量导出的环境变量。

| **相关命令** | **说明** |
| --- | --- |
|  |  |

* set | 显示当前 Shell 所有环境变量，包括其内建环境变量（与 Shell 外观等相关），用户自定义变量及导出的环境变量
* env | 显示与当前用户相关的环境变量，还可以让命令在指定环境中运行
* export| 显示从 Shell 中导出成环境变量的变量，也能通过它将自定义变量导出为环境变量

**输出环境变量**

#将普通变量输出为环境变量，能被子进程继承

export 变量名

#比较文件的命令

vimdiff 1.txt 2.txt 3.txt ....

**3.命令的查找路径与顺序**

$PATH环境变量设置命令搜索路径

#输出当前命令搜索路径的设置

echo $PATH

#./程序名 表示执行当前目录下的指定程序

**$PATH中临时添加自定义路径,用冒号连接路径**

PATH=$PATH:/home/shiyanlou/Code

如果永久添加需要修改配置文件 zsh是.zshrc bash是.bashrc 本次修改用户目录下的文件

$ echo "PATH=$PATH:/home/shiyanlou/mybin" >> .zshrc

**变量修改**

变量的修改有以下几种方式：

| **变量设置方式** | **说明** |
| --- | --- |
| ${变量名#匹配字串} | 从头向后开始匹配，删除符合匹配字串的最短数据 |
| ${变量名##匹配字串} | 从头向后开始匹配，删除符合匹配字串的最长数据 |
| ${变量名%匹配字串} | 从尾向前开始匹配，删除符合匹配字串的最短数据 |
| ${变量名%%匹配字串} | 从尾向前开始匹配，删除符合匹配字串的最长数据 |
| ${变量名/旧的字串/新的字串} | 将符合旧字串的第一个字串替换为新的字串 |
| ${变量名//旧的字串/新的字串} | 将符合旧字串的全部字串替换为新的字串 |

**变量删除**

可以使用unset命令删除一个环境变量：

$ unset temp

**6.如何让环境变量立即生效**

$ source .zshrc

或者

$ . ./.zshrc

注意第二种方式必须指定完整的绝对或相对路径

**二、搜索文件**

与搜索相关的命令常用的有如下几个whereis,which,find,locate。

* **whereis简单快速**

whereis搜索很快，因为它并没有从硬盘中依次查找，而是直接从数据库中查询。whereis只能搜索二进制文件(-b)，man帮助文件(-m)和源代码文件(-s)。

* **locate快而全**

通过"/var/lib/mlocate/mlocate.db"数据库查找，系统会使用定时任务每天自动执行updatedb命令更新一次该数据库，有时需要手动执行一次updatedb命令将新的文件数据添加进数据库，它可以用来查找指定目录下的不同文件类型,可以递归子目录进行查找，如查找 /etc 下所有以 sh 开头的文件：

$ locate /etc/sh

$ locate /home/shiyanlou/Code/\\*.txt

经测试好像locate 后面要跟绝对路径,另外"*"必须用\*来转义

-c 是仅统计数目 -i 是不区分大小写

* **which小而精**

它只从PATH环境变量指定的路径中去搜索命令

$ which man

* **find精而细** find 命令非常强大，最好看专门的资料 推荐链接<http://www.cnblogs.com/wanqieddy/archive/2011/06/09/2076785.html> <http://www.jz123.cn/text/1720453.html> 以下列出几个例子

#查找系统中所有文件名中含D且在第三位的.txt文件

find / -name \?\?D\\*.txt

#显示结果含有权限不够的信息,可以如下屏蔽

find / -name \?\?D\\*.txt 2>/dev/null

#还可以加sudo执行

#另外，文件名加引号，可以直接用?或\*，不用转义

#查找大小在10k-20k之间的\*.txt

find / -name '\*.txt' -size +10k -size -20k

#查找空目录和空文件

find . -empty

#查找实验楼组的文件

find . -group shiyanlou

#在/tmp目录中查找大于100000000字节并且在48小时内修改的某个文件

find /tmp -size +10000000c -and -mtime +2

#在/tmp目录中查找所有不属于panda的文件

find /tmp ! -user panda

#查找属于shiyanlou用户的文件并显示文件信息

find . -user shiyanlou -ls

#删除找到的\*.txt

find . -name '\*.txt' -exec rm {} \;

#以上 \; 是为了转义;字符

### 联机帮助、清屏与历史记录命令

**联机帮助**

man name

man -k keyword

spacebar     翻屏或翻页

Enter          翻行

b                向前翻一屏或一页

f                 向后翻一屏或一页

q                退出

/string        查找前面符合string指定的信息

n                查找下一个符合string指定的信息

**清屏**

       clear

**历史记录**

       history

       history n

       !n

       !!

       方向上键

## 文件管理

### 查看文件目录

**查看文件和目录**

**pwd**

   显示当前工作目录

**cd**

改变当前工作目录

       cd [directory\_name]



**ls**

列出当前目录的文件和子目录

       ls

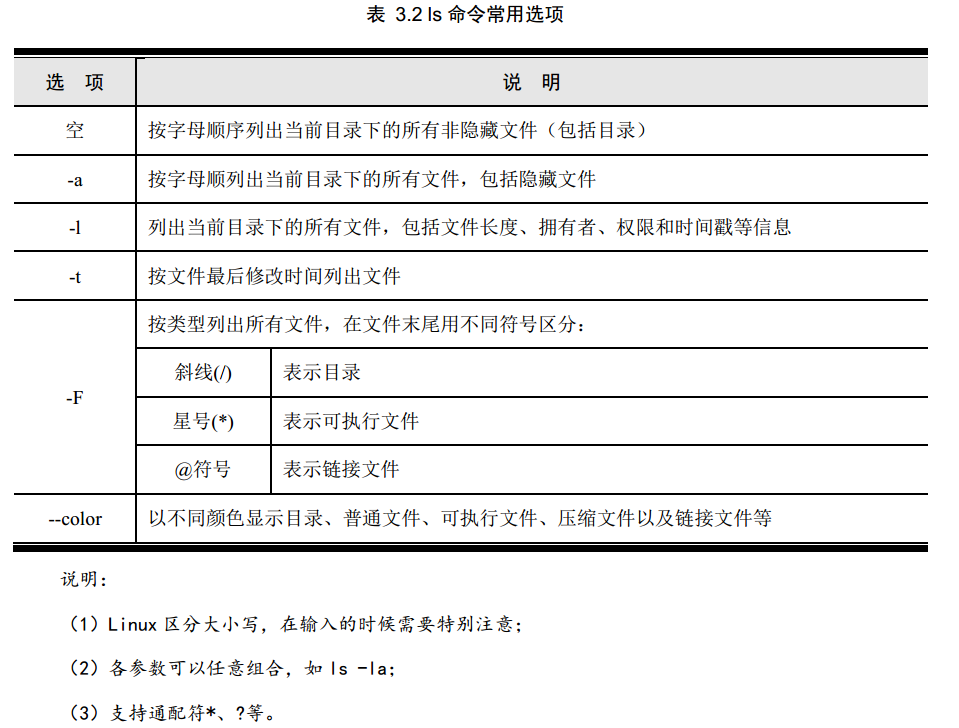
       ls -a     列出所有目录、子目录、文件和隐藏文件

       ls -R    列出从当前目录开始的所有的子目录、文件并且一层层往                  下显示

       ls -F     列出文件、目录名并显示出文件类型

       ls -t     以修改时间为时间倒序来列出文件、子目录

       ls -l      以长列表格式显示文件、目录的详细信息



**file**

       file filename



### 创建和删除文件目录

### touch,mkdir,rm,rmdir

**创建文件**

    touch filename

    filename 不存在则创建一个新的空文件

    filename 存在则更新该文件的修改访问时间

touch ml.txt ls –l ml.txt

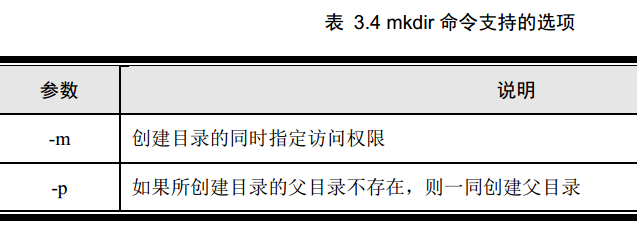
**创建目录**

    mkdir [-p] directory\_name

    -p    目录不存在时可以建立目录

 mkdir dir1 mkdir /home/dir1 mkdir dir1/dir2/dir3 mkdir dir2 dir3 dir4

mkdir ~/game mkdir –p dir7/dir8/dir9



**删除文件**

    rm [-i/f] filename(s)

    -i 询问是否删除

    -f 强制删除

    rm ml1.txt ml2.txt ml3.txt

**删除目录**

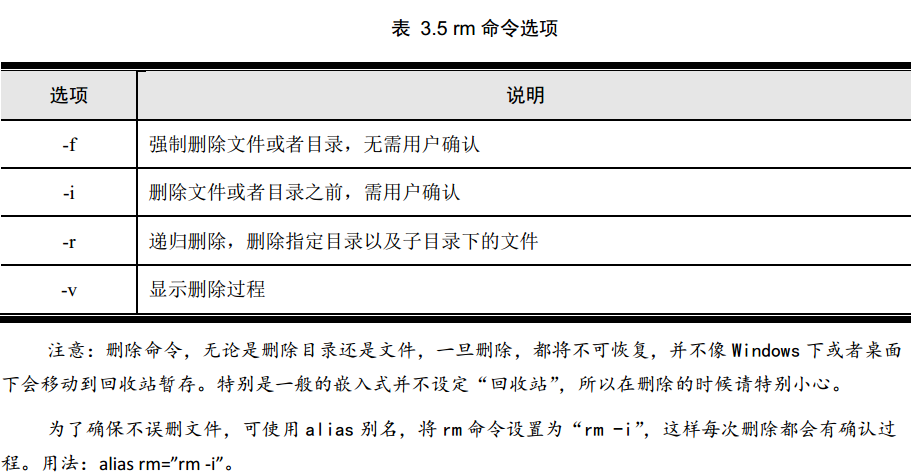
    rmdir directory\_name(s)    删除空目录

    rm -r[i] directory\_name(s)

    -r 删除非空目录

    -i 删除前询问是否删除

rmdir dir1 rm –r dir2 rm –ri dir3 rm –rf dir7



### ln

**创建链接文件**

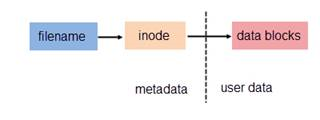
**Linux文件管理特性**

    文件都有文件名与数据，在 Linux 上被分成两个部分：用户数据     (user data) 与元数据 (metadata)。

    用户数据，即文件数据块 (data block)，数据块是记录文件真实内容    的地方

    元数据则是文件的附加属性，如文件大小、创建时间、所有者等信息。

    在Linux中元数据中的inode 号（inode 是文件元数据一部分但其并不    包含文件名，inode 号即索引节点号）才是文件的唯一标识而非文件名    。文件名仅是为了方便人的记忆和使用，系统或程序通过 inode 号找    正确文件数据块。



**为什么会有链接文件**

      为解决文件的共享使用，Linux 系统引入了链接，链接为Linux 系统解决了文件的共享使用，还带来了隐藏文件路径、增加权限安全及节省存储等好处。创建链接文件可以给文件和目录创建替代名或别名。

**分类**

       硬链接 (hard link) 与软链接（又称符号链接，即 soft link 或        symbolic link）

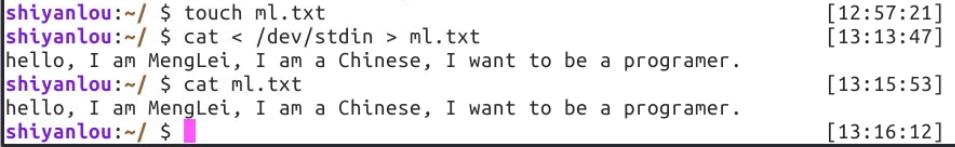
       ln source\_file target\_file

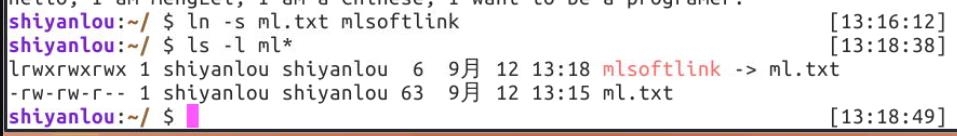
       建立硬链接

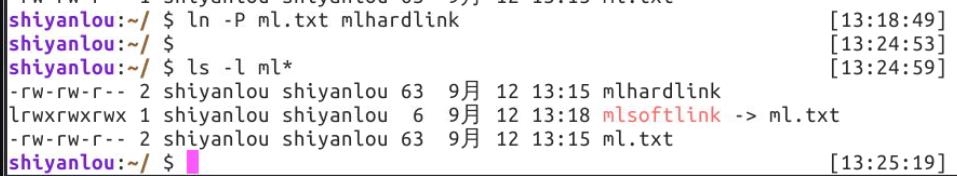
       ln [-s或P] source\_file target\_file

       -s：建立软连接

       -P：建立硬链接







**硬链接和软连接的区别**

       1）硬链接

           若一个 inode 号对应多个文件名，则称这些文件为硬链接。换言之 ，硬链接就是同一个文件使用了多个别名。

           只能在同一个文件系统来链接文件，源文件和目标文件同一个文件，当有多个硬链接文件时，删除其中一个则是删除了一个文件 名而已，硬链接数少1个。

           由于硬链接是有着相同 inode 号仅文件名不同的文件，因此硬链 接存在以下几点特性：

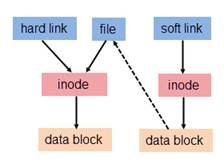
               文件有相同的 inode 及 data block；

               只能对已存在的文件进行创建；

               不能交叉文件系统进行硬链接的创建；

               不能对目录进行创建，只可对文件创建；

               删除一个硬链接文件并不影响其他有相同 inode 号的文件。



       2）软链接

           若文件用户数据块中存放的内容是另一文件的路径名的指向，则该文件就是软连接。软链接就是一个普通文件，只是数据块内容有点特殊。软链接有着自己的 inode 号以及用户数据块。

           可在不同的文件系统来链接文件，源文件和目标文件是不同文件 ，有不同的大小，是2个文件， 目标文件的内容是源文件的inode号指向源文件，像windows中的快捷方式一样。

           建立了软连接后，软连接文件的大小是指向的目标文件的文件名的大小

           软链接与硬链接不同，软链接创建与使用没有类似硬链接的诸多 限制

               软链接有自己的文件属性及权限等；

               可对不存在的文件或目录创建软链接；

               软链接可交叉文件系统；

               软链接可对文件或目录创建；

               创建软链接时，链接计数 i\_nlink 不会增加；

               删除软链接并不影响被指向的文件，但若被指向的原文件被删除，则相关软连接被称为死链接（即 dangling link，若被指向路径文件被重新创建，死链接可恢复为正常的软链接）。

**链接文件命令操作**

       创建软硬链接

       查看软硬链接文件及其大小

       删除软硬链接

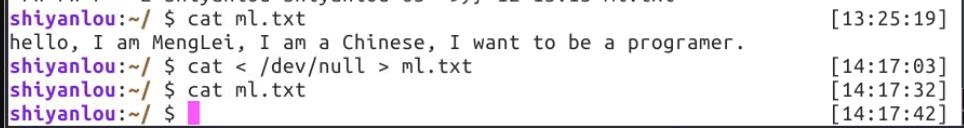
### 显示文本文件内容

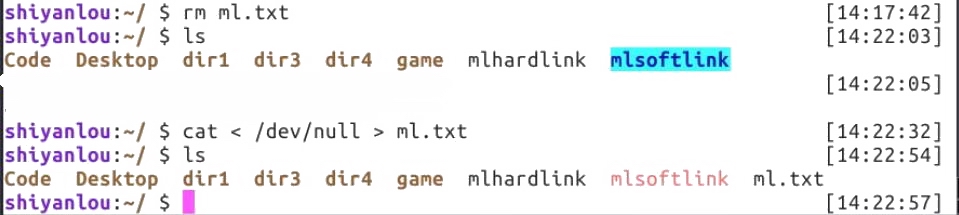
**cat**

    不分屏显示文本内容

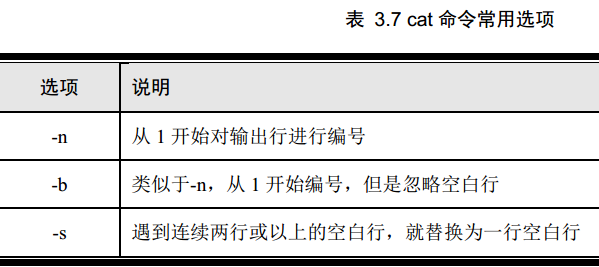
    把小文件连接成大文件

    cat filename(s)



****

**cat /etc/passwd | awk –F: ‘{print $1 “\t” $6}’ \ > |sort > userinfo**

**more**

    more filename(s)

    spacbebar      向下显示一屏

    Enter             向下显示一行

    b                   向上一屏

    f                    向下一屏

    h                   显示帮助菜单

    q                   退出

    /string           查找字符串string

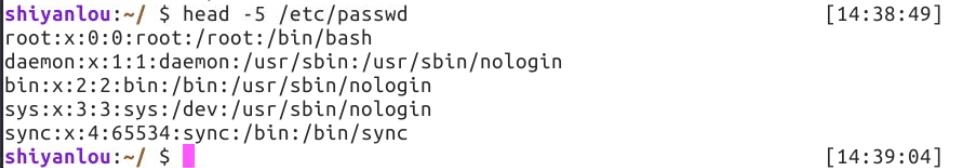
    n                   查找下一个string

 man pwd | more

**head**

    查找文件的前多少行

    head [-n] filename(s) 



**tail**

    查找文件的末尾多少行

    tail [-n] filename(s)



### 拷贝和移动文件目录

**cp**

    拷贝复制文件目录

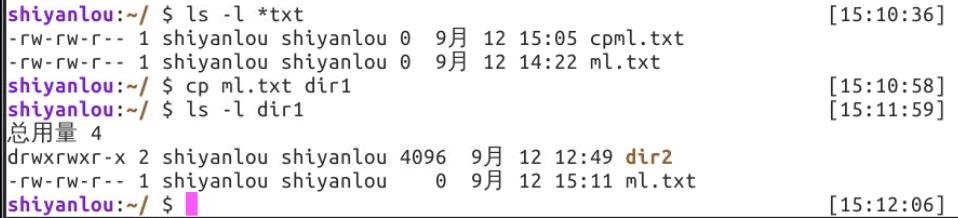
    拷贝文件

    cp [-i] source\_file destination\_file

    cp [-i] source\_file(s) destination\_directory

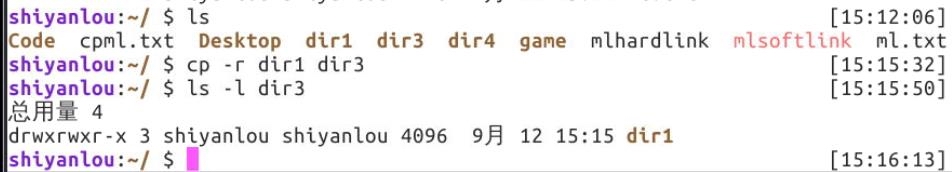
    -i选项作用：当目标文件存在，会询问是否覆盖，没有-i选项则不询问直接覆盖





拷贝目录

    cp -r source\_directory(s) destination\_directory(s)



****

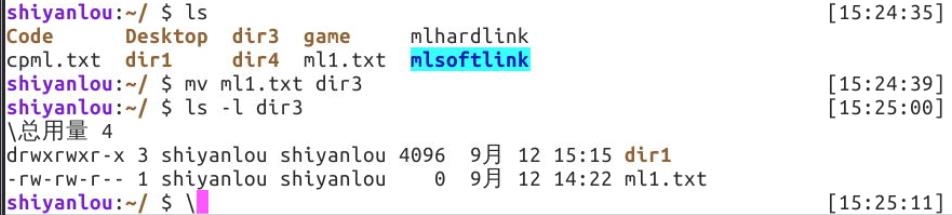
**mv**

    移动文件目录或重命名文件目录

    mv [-i] source\_file target\_file 重命名源文件为目标文件

    mv [-i] source\_file target\_directory 移动文件到目标目录





### 文件目录权限

### Linux的文件目录权限概述

**什么是文件目录权限**

    Linux系统中的每个文件和目录都有访问许可权限，用他来确定谁能通过何种方式对文件和目录进行访问和操作。

**权限的分类**

    r  读权限：可以打开文件、目录读取查看；

    w 写权限：对文件、目录可以编写更改

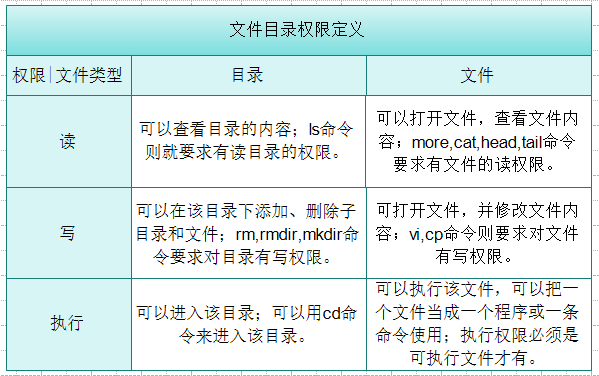
    x 执行权限：对文件可执行(可执行文件)、对目录可查找该目录下的内容

    - 没有权限

       如 ls -l

       -rwxr-xr-x

**文件目录权限定义**

****

**权限所属对象**

拥有者：生成文件或目录时登陆的当前人，权限最高，用u表示

       同组人：系统管理员分配的同组的一个或几个人，用g表示

       其他人：除拥有着，同组人以外的人，用o表示

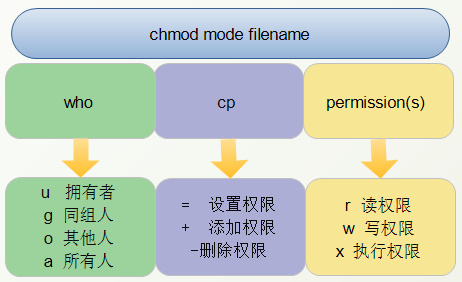
       所有人：包括拥有着、同组人及其他人，用a表示

### chmod,chown,chgrp

**chmod**

修改文件目录的访问权限，修改权限的前提条件是在修改权限时，要注意自己是文件

**使用字母表示权限**

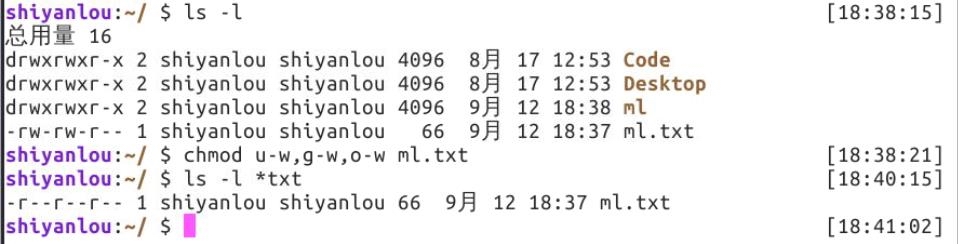
****

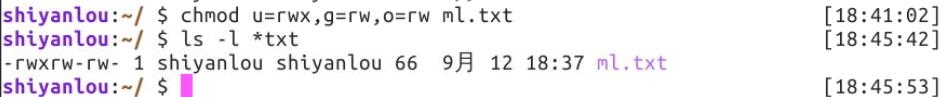
chmod u  =  r

                  g +  w

                  o -   x

                  a



****

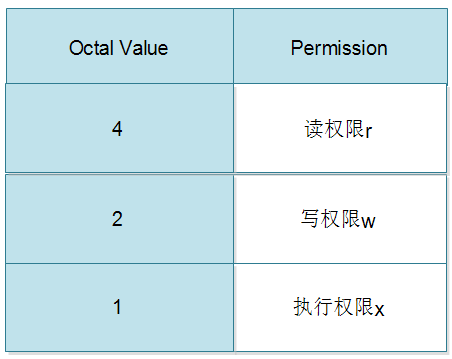
**使用数字表示权限**

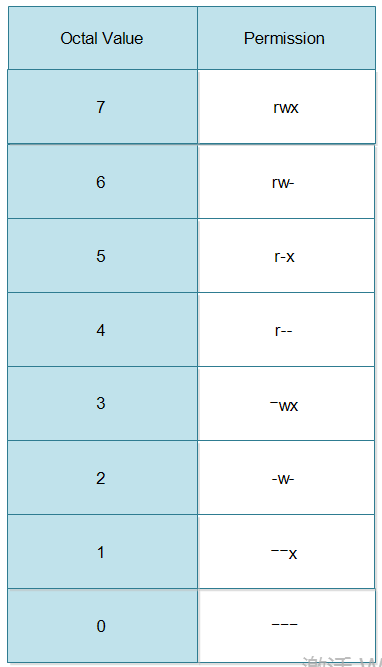
       使用八进制数字来表示权限

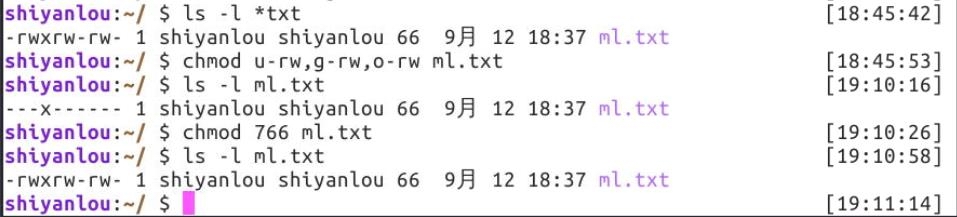
       r  w  x

       0  0  0     无权限

       1  1  1    有权限







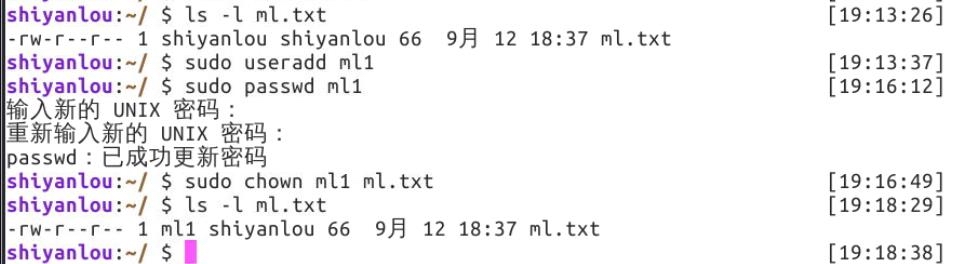


**chown**

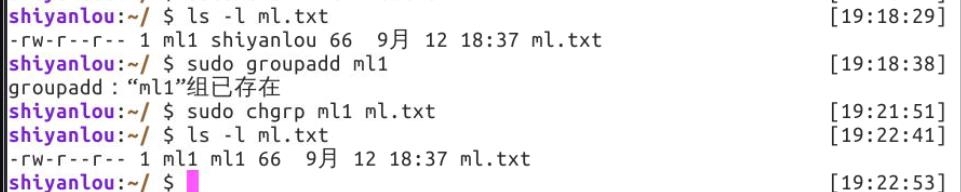
       更改某个文件或目录的属主和属组，可用于授权；

       例如root用户把自己的一个文件拷贝给用户xu，为了让用户xu能够存取这个文件，root用户应该把这个文件的属主设为xu，否则，用户xu无法存取这个文件。

       chown [选项] 用户或组 文件



       chown将指定文件的拥有者改为指定的用户或组。用户能是用户名或用户ID。组能是组名或组ID。



       文件是以空格分开的要改动权限的文件列表，支持通配符。

       - R 递归式地改动指定目录及其下的所有子目录和文件的拥有者。

       - v 显示chown命令所做的工作。

       $ chown wang shiyan.c

       $ chown -R wang users /his

**chgrp**

       改动文件或目录所属的组。

       chgrp [选项] group filename

       该命令改动指定指定文件所属的用户组。其中group能是用户组        ID，也能是/etc/group文件中用户组的组名。

       文件名是以空格分开的要改动属组的文件列表，支持通配符。

       如果用户不是该文件的属主或终极用户，则不能改动该文件的组。

       - R 递归式地改动指定目录及其下的所有子目录和文件的属组。

       $ chgrp –R book /opt/local /book

       改动/opt/local /book/及其子目录下的所有文件的属组为book。

### 查找文件

**find**

    file path expression [action]  查找文件和目录

    前提条件：要对被查找的目录及其所有子目录有读权限才能查找

    查找选项：通过文件属性来查找

    -name 按文件名

    -user 按用户(文件属主)

    -size 按大小

    -mtime 按最后一次修改时间

    -atime 按最后一次访问时间

    -type 按文件类型  f:file d:directory

    -perm 按权限

    find / -name b\*

    find . -mtime 10 -print

    find /etc -user 0 -size +400 -print

    find ~ -perm 777 > ~/holes

    find /export/home -type f -atime +365 -exec rm {}\;

  find ~-perm 644 >~/ml1.txtS

**locate**

    locate [-d <数据库文件>][--help][--version][keywords]

    locate指令用于查找符合条件的文件，它会去保存文件与目录名称的数据库内查找合乎范本样式条件的文件或目录。

    -d<数据库文件>或--database=<数据库文件> ：设置locate指令使用的数据库

    locate指令预设的数据库位于/var/lib/slocate目录里，文件slocate.db，您可使用这个参数另行指定。

    --help 　在线帮助。

    --version 　显示版本信息。

    速度快很多，它是通过inode，文件索引来找，它会把文件索引维护在一个数据库里面，它在数据库去找；比较麻烦的是需要更新数据updatedb;

locate . ml\* locate –c ml\*

### 过滤与统计

**grep**

       查出包含某些字符串的结果，对文件或输出结果进行过滤，大小写敏感

       grep [option(s)] string filename

       -i 忽略大小写

       -v 反向匹配（查出不包含字符串的结果）



**wc**

       统计文件或输出结果

       wc [option(s)] filename(s)

       -l  统计多少行

       -w 统计多少个单词

       -c  统计多少个字符

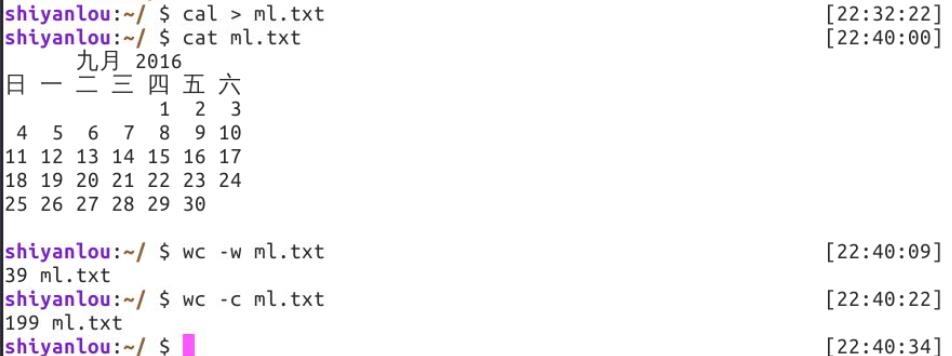
who | grep shiyanlou | wc -l

wc –l /etc/passwd

grep li /etc/passwd | wc –l

wc –c ml.txt

wc –w ml.txt



grep 是 Linux 系统中一个强大的文本搜索工具，用法很多很复杂，这里也仅仅介绍简单的常用用法。语法格式：

$grep 选项

只要提供查找的关键字，用 grep 命令就可以完成查找。例如，想知道 pcf8563 这个关键字在“ arch/arm”目录下哪些地方用到了，可以输入下列命令：

$grep ―pcf8563‖ -R arch/arm

关键字最好加上双引号，特别是包含空格的关键字，对于单个关键字倒是可以不用引号。

“ -R”表示进行递归查找，而不是仅仅在指定的目录下查找。

命令和实际操作结果如图 3.26 所示，可以看到，凡是用了“ pcf8563”的地方都被列了出来，并且红色高亮显示。



Shell 文件

Shell 文件是以某种方式将一些命令放在一起得到的文件，常称为 Shell 脚本。 Shell 文件通常以“ #!/bin/sh”开始，#!后面指定解释器，如下是一个简单的 Shell 文件的内容：

#!/bin/sh

echo ―hello, I am shell script‖

假定此文件名为 a.sh，增加可执行权限后，在 Shell 中即可运行，将在终端打印“ hello,I am shell script”字符串。

$chmod +x a.sh

$./a.sh

hello, I am shell script

执行 Shell 脚本有多种方式：

（ 1） **点+斜线+文件名**， 这种方式要求文件必须有可执行权限；

（ 2） **点+空格+文件名**，这种方式不要求文件一定具有可执行权限。

（ 3） **sh+空格+文件名**，这种方式不要求文件一定具有可执行权限。

（ 4） **source+空格+文件名**，这种方式不要求文件一定具有可执行权限。

### 日期时间与进程查看

**date**

       显示当前日期

**cal**

       显示日历

       cal 月份 年份

       cal 显示当前月份的日历

       cal 年份

**ps**

       列出当前系统中已在运行的进程

       进程：一个能完成一定功能的一个程序

       ps [options]

       ps  列出与自己当前终端相关的进程

       -e  列出正在运行的进程

       -f   列出一个完整的进程列表形式

       -U  查找出由某个用户启动的进程

       -auwx  列出一个完整的进程列表形式，形式与-f不一样

切换用户、设置普通用户的超级权限与关机重启

**su**

       su 用户名

**sudo**

       sudo 命令行

**shutdown**

       安全地关闭或重启Linux系统

       它在系统关闭之前给系统上的所有登录用户提示一条警告信息。

       该命令还允许用户指定一个时间参数，可以是一个精确的时间，也可以是       从现在开始的一个时间段。

       精确时间的格式是hh:mm，表示小时和分钟；时间段由+ 和分钟数表示       。系统执行该命令后会自动进行数据同步的工作。

       shutdown [-efFhknr][-t 秒数][时间][警告信息]

       　 -c 　当执行"shutdown -h 11:50"指令时，只要按+键就可中断关机           的指令。

　        -f 　重新启动时不执行fsck。

　        -F 　重新启动时执行fsck。

　        -h 　将系统关机。

　        -k 　只是送出信息给所有用户，但不会实际关机。

　        -n 　不调用init程序进行关机，而由shutdown自己进行。

　        -r 　shutdown之后重新启动。

　        -t<秒数> 　送出警告信息和删除信息之间要延迟多少秒。

　    [时间] 　设置多久时间后执行shutdown指令。

　    [警告信息] 　要传送给所有登入用户的信息。

       # shutdown –r +10       系统在十分钟后关机并且马上重新启动

       # shutdown –h now       系统马上关机并且不重新启动

**reboot**

       reboot的工作过程差不多跟halt一样。不过它是引发主机重启，而halt是       关机。它的参数与halt相差不多。

## 用户管理

### 密码设置与修改

**用户密码设置要求**

       6-8个字符

       至少包含2个字母，1个数字或特殊符号

       不同于用户ID

       不同于以前的密码，并且至少3个字符不同于以前的密码

**修改密码命令**

       passwd

       输入命令passwd

       输入原密码

       输入新密码

       确认新密码

### passwd

### 查找用户

**id**

       查看用户ID(用户名)，所属组ID(组名)

**users**

查看已经登陆到当前系统中的用户，只显示出用户名

**who**

查看用户的详细信息

**who am i**

       查看当前用户自己的信息

**whoami**

       查看当前用户自己的用户名

## 磁盘管理

### 查看磁盘信息

**du**

       显示磁盘使用摘要信息

       du 以Block为单位方式显示

       -k  以k字节方式显示

       -m 以m字节方式显示

       -s  显示当前目录下的内容总的占用磁盘的大小，以Block为单位

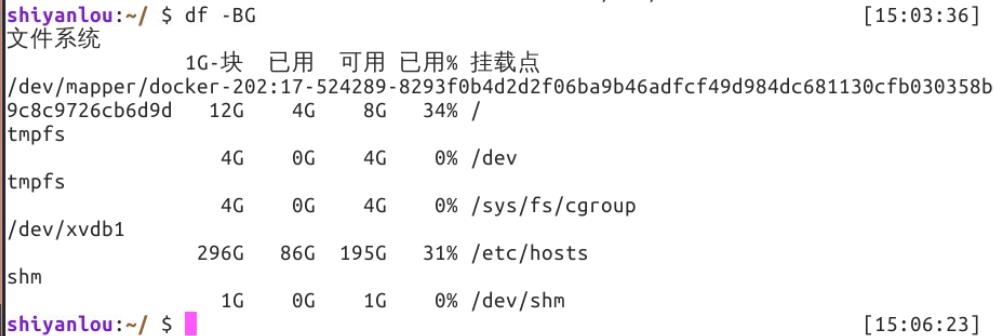
       以Block单位显示的数字是以k字节方式显示的数字的2倍：1k字节=2个Block

**df**

       显示整个文件系统的空间使用磁盘情况

       -k 以k字节方式显示

df –BG



## 网络配置管理

### 查看网络连接

**ping**

       查看当前机器与另一台机器的联通状况

       ping 主机IP/主机名

       向ping后面的主机发送数据包，若被ping的主机有回复则表明是联通的

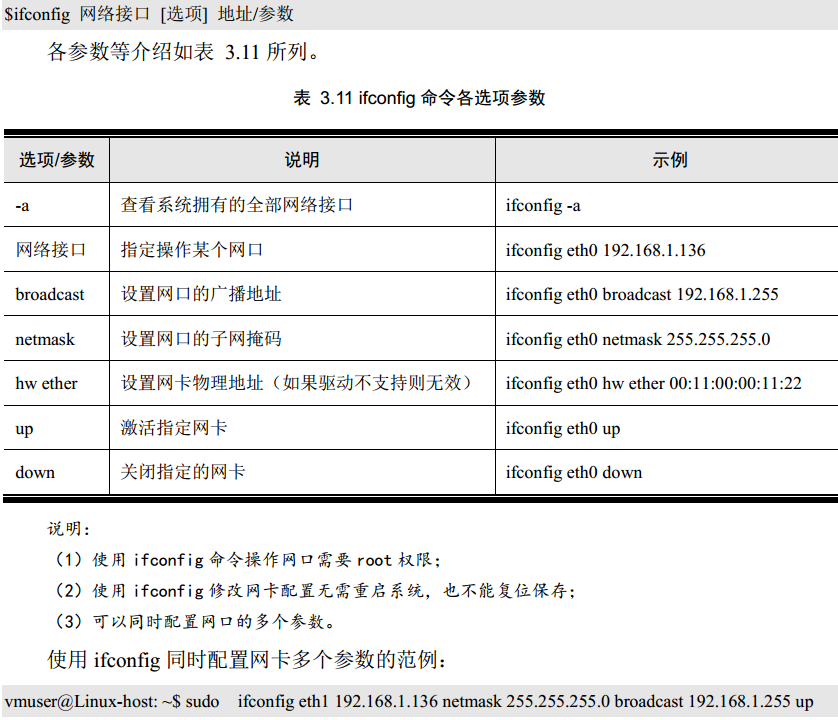
**ifconfig**

       查看和配置当前机器的网络参数信息

       ifconfig -a    显示查看当前机器的IP、Netmask、Gateway等网络信                           息

       ifconfig eth0 up(down)    激活与关闭某个网络适配卡

       ifconfig eht0 [ipaddress] netmask [address]  设置IP和子网掩码



# Linux应用程序的安装与卸载基础篇

**课程目标**

       了解常用的软件包deb & rpm

了解ubuntu下三种软件安装卸载方式

**重点难点**

知道ubuntu下三种软件安装卸载方法的命令

**考核目标**

**课后练习**

**小贴士/名词解释**

       deb，debian，rpm

## Linux应用程序安装与卸载基础知识

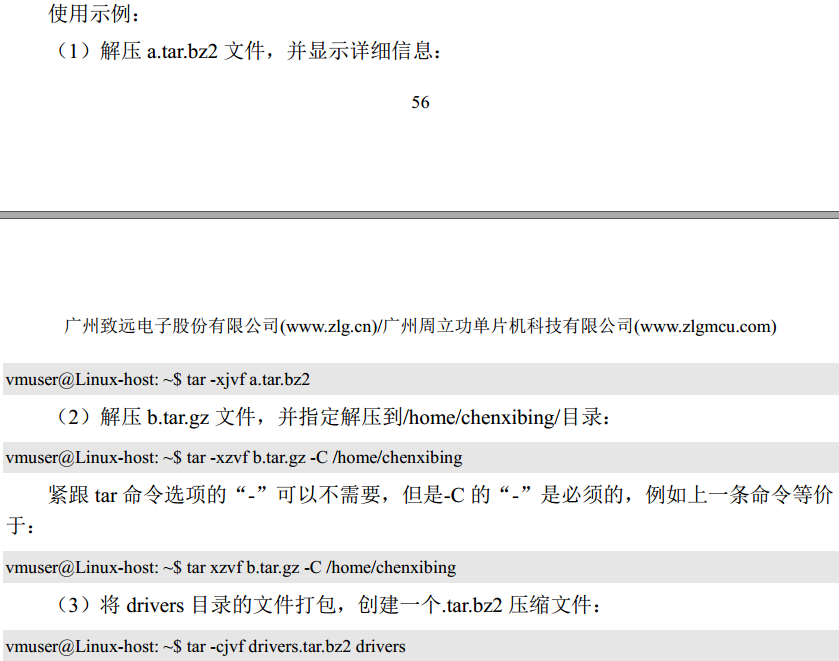
**Linux软件安装包的概念**

**什么是软件安装包**

       类似windows下的安装程序，如打好包的exe文件

       在Linux下的打包文件通常都是tar，打包格式可自己定义，打包文件可以自定义一个后缀名；如tar.abc





**Linux下的常见的软件程序包**

       rpm,deb

       rpm是红帽子系统定义的软件包文件格式

       deb是ubuntu下面的主要的安装包的格式

       安装包的具体格式规范是什么，具体不同的安装包的格式规范要到具体官网去查看其格式规范。

**Linux下的安装包的命名格式**

       软件包名称\_版本号-修订版本\_体系架构.扩展名

       aptitude\_0.6.3-3.2ubuntu1\_i386.deb

       libboost-iostreamsl.42.0\_1.42.0-4ubuntu2\_i386.deb

**Linux下安装卸载应用程序的方式**

       安装包离线安装和卸载：dpkg

          dpkg -i <package> 安装包

          dpkg -P <package> 移除包和配置文件

       源文件编译安装和卸载：配置configure、编译make和安装 make              install

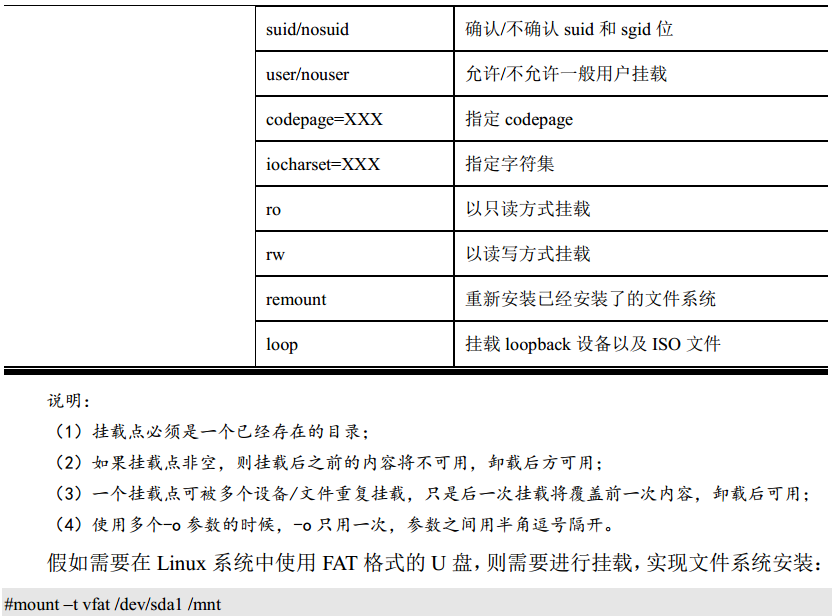
       程序管理包在线安装和卸载:aptitude

           apt-get install <package>  安装

apt-get  remove -purge <package> 卸载完全

文件系统挂载  
Linux 允许多个文件系统存在于同一个系统中，也允许用户在系统运行中安装内核所支  
持的文件系统。例如，将一个 FAT 格式的 U 盘插入到 Linux 系统中。  
往系统安装文件系统需要用到 mount 命令，并且需要 root 权限。用法：  
# mount [-参数] [设备名称] [挂载点]





在进行嵌入式 Linux 开发过程中， mount 命令经常被使用，特别是进行 NFS 连接和调试

的时候，通过 NFS 挂载，将远程主机 Linux 的某个共享目录挂载到嵌入式系统本地，当成

本地设备进行操作。 NFS 挂载范例：

[root@zlg /]#mount -t nfs 192.168.1.138:/home/chenxibing/lpc3250 /mnt -o nolock

nolock 表示禁用文件锁，当连接到一个旧版本的 NFS 服务器时常加该选项。

此外，嵌入式开发中常用的文件系统还有 cramfs、 jffs2、 yaffs/yaffs2 以及 ubifs 等，特

别是用于 NOR Flash 的 jffs2 和用于 NAND Flash 的 yaffs/yaffs2、 ubifs 等，在进行系统操作

中通常需要对各设备进行挂载或者卸载，需要在挂载的时候指定正确的文件系统类型。挂载

yaffs2 分区的命令示例：

#mount -t yaffs2 /dev/mtdblock2 /mnt

\*使用条件：需要内核已经支持 yaffs2 文件系统。

挂载 ubifs 分区的命令示例：

#mount -t ubifs ubi0:rootfs /mnt

\*使用条件：需要内核已经支持 ubifs 文件系统。

2. 文件系统卸载

当不再需要某个文件系统的时候，可以将其卸载。 umount 用于卸载已经挂载的设备或

者文件。用法：

#umount 挂载点

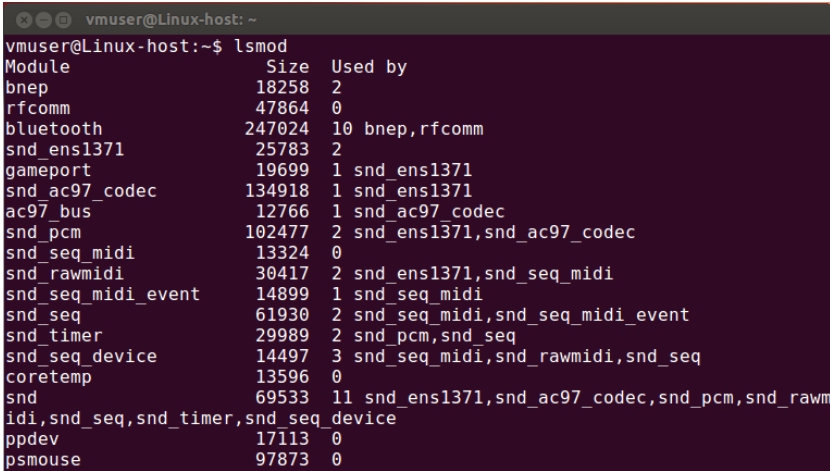
如果已经将 U 盘挂载到/mnt 目录下，用完后的卸载命令为：

[root@zlg /]# umount /mnt

使用内核模块和驱动  
1. 加载（插入） 模块  
Linux 是一个具有模块化特性操作系统，允许在内核运行中插入模块或者卸载不再需要  
的模块。能够动态加载和卸载模块是 Linux 引以为豪的特性之一，如果某些功能平时用不到，  
可以不用编进内核，而采取模块方式编译，在需要的时候再插入内核，不再需要的时候卸载，  
这样可以精简内核，提高效率，并提高系统的灵活性。 Linux 中最常见的模块是内核驱动，  
掌握模块的加载和卸载，也是使用 Linux 必须掌握的方法之一。  
通过 insmod 命令可以往正在运行中的内核插入某些模块而无需重启系统。用法：  
# insmod [选项] 模块 [符号名称=值]  
insmod 常用选项介绍如表 3.13 所列。  
表 3.13 insmod 命令常用选项

|  |  |
| --- | --- |
| 选项 | 说明 |
| -f | 强制将模块载入，不检查目前 kernel 版本与模块编译时的 kernel 版本是否一致 |
| -k | 将模块设置为自动卸载 |
| -p | 测试模块是否能正确插入 |
| -x | 不导出模块符号 |
| -X | 导出模块所有外部符号（默认） |
| -v | 显示执行过程 |

一般情况下，如果一个模块的版本与所运行的内核不一致，模块将无法插入系统。就算  
是同一版本内核编译得到，如果内核配置文件不同，也有可能无法插入。使用-f 选项强制插  
入后，可能会出现运行不正确的情况。  
插入和卸载模块需要 root 权限。插入模块比较简单，如需要往系统插入 beep.ko 驱动模  
块，可用：  
[root@zlg beep]# insmod beepdrv.ko  
有些驱动/模块可以接受外部参数，在插入模块的时候为相应的符号赋值。一个模块/驱  
动能否接受外部参数，能够接受几个外部参数，取决于模块/驱动的具体实现，符号以及赋  
值请参考相应模块/驱动的说明。这是一个范例， pcm-8032a.ko 模块能接收 irq 和 addr 两个  
符号参数，插入模块的时候指定：  
# insmod pcm-8032a.ko irq=3 addr=0x300  
2. 查看系统已经加载的模块  
如果想要知道某个模块是否已经插入系统，或者想知道系统加已经载了哪些模块，可用  
lsmod 命令查看。 Lsmod 命令用法：  
vmuser@Linux-host: ~$lsmod  
lsmod 命令结果实际上就是列出/proc/modules 的内容，结果如图 3.24 所示。



卸载驱动模块  
当某个内核模块或者驱动不再需要被使用，则可以将其从系统中卸载，以释放所占用的  
资源。卸载模块用 rmmod 命令， 用法：  
# rmmod [选项] 模块  
rmmod 命令常用可选项如表 3.14 所列。

|  |  |
| --- | --- |
| 选项 | 说明 |
| -f | 强制卸载正在被使用的模块，非常危险！需要内核支持（ CONFIG\_MODULE\_FORCE \_UNLOAD 使能），否则无效 |
| -w | 通常情况下不能卸载正在被使用的模块，加上-w 选项，指定模块将会被孤立，直到不再被 使用 |
| -s | 将错误信息写入 syslog，而不是标准错误 |
| -v | 显示执行过程 |

如果一个模块正在被另外一个模块所依赖，或者正在被某个应用程序使用，则一般情况  
下无法卸载这个模块。如果内核支持强制卸载模块功能，加上-f 可以卸载，但是不要轻易使  
用，否则有可能会带来严重错误。假定系统的 beep.ko 不再需要，卸载命令：  
[root@zlg beep]# rmmod beepdrv.ko  
4. 自动处理可加载模块  
前面提到的 insmod/rmmod 分别用于加载和卸载模块，但是每次只能加载/卸载一个模块，  
如果一个模块依赖于多个模块，则需要进行多次操作，比较繁琐。 modprobe 命令集加载/卸载功能于一身，并且可以自动解决模块的依赖关系，将某模块所依赖的其它模块全部加载。  
用法：  
# modprobe [选项] 模块 [符号=值]  
modprobe 也支持很多选项，常用选项如表 3.15 所列。  
表 3.15 modprobe 常用选项

|  |  |
| --- | --- |
| 选项 | 说明 |
| -C <文件> | 不使用默认配置文件，使用指定文件作为配置文件 |
| -i | 忽略配置文件中的加载和卸载命令 |
| -r | 卸载指定模块，包括依赖模块 |
| -f | 强制安装 |
| -l | 显示所有匹配模块 |
| -a | 安装所有匹配的模块 |
| --show-depends | 显示模块的依赖关系 |
| -v | 显示执行过程 |
| -q | 不显示任何信息 |
| -V | 显示版本信息 |

modprobe 处理模块时忽略模块的路径，这要求系统模块和驱动是按照 make modues\_install 方式安装的，即模块必须放在/lib/modules/$(uname -r)目录下，并且有正确的/lib/modules/$(uname -r)/modules.dep 文件， modprobe 根据该文件来寻找和解决依赖关系。

创建设备节点  
如果系统不能自动创建设备节点，加载驱动后，则需要为驱动建立对应的设备节点，否则无法通过驱动来操作设备。只有 root 用户才能创建设备节点，命令为 mknod，用法：  
#mknod 设备名 设备类型 主设备号 次设备号  
如需要创建一个字符设备 led，主设备号为 231，次设备号为 0，则命令如下：  
#mknod /dev/led c 231 0

文件同步

经常会遇到这样的情形：刚刚修改了某个文件，突然断电，重启后发现刚刚做的修改并没有保存，或者被修改的文件已经损坏。这是由于 Linux 中，对文件的操作都是先保存在缓存中，并没有立即写入磁盘，经系统调度后方可写入磁盘。如果修改了缓存，还没来得及写到磁盘就断电，自然会造成文件改变丢失。要避免这种情况发生，就是修改文件后，立即强制进行一次文件同步操作，将缓存的内容写入磁盘，确保文件系统的完整性。能完成这样功能的命令是 sync。只需在关闭文本编辑器后在 Shell 输入 sync 即可。

**Linux** 环境变量  
Linux 是一个多用户操作系统，每个用户都有自己专有的运行环境。用户所使用的环境由一系列变量所定义，这些变量被称为环境变量。系统环境变量通常都是大写。每个用户都可以根据需要修改自己的环境变量，以达到自己的使用要求。常见的环境变量如表 3.16 所列。

|  |  |
| --- | --- |
| 变量 | 说明 |
| PATH | 决定了 Shell 将到哪些目录中寻找命令或程序，这个变量是在日常使用中经常需要修 改的变量。 |
| TERM | 指定系统终端 |
| SHELL | 当前用户 Shell 类型 |
| HOME | 当前用户主目录 |
| LOGNAME | 当前用户的登录名 |
| USER | 当前用户名 |
| HISTSIZE | 历史命令记录数 |
| HOSTNAME | 指主机的名称 |
| LANGUGE | 语言相关的环境变量，多语言可以修改此环境变量 |
| MAIL | 当前用户的邮件存放目录 |
| PS1 | 基本提示符，对于 root 用户是#，对于普通用户是$ |
| PS2 | 附属提示符，默认是“ >” |
| LS\_COLORS | ls 命令结果颜色显示 |

在 Shell 下通过美元符号（ $）来引用环境变量，使用 echo 命令可以查看某个具体环境变量的值。例如，查看 TERM 的值：  
$echo $TERM  
使用 env 或者 printenv 命令可以查看系统全部的环境变量设置，例如：  
chenxibing@gitserver-zhiyuan:~$ env  
TERM=xterm  
SHELL=/bin/bash  
USER=chenxibing  
MAIL=/var/mail/chenxibing  
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games  
PWD=/home/chenxibing  
LANG=zh\_CN.UTF-8  
SHLVL=1  
HOME=/home/chenxibing  
LANGUAGE=zh\_CN:zh  
LOGNAME=chenxibing  
**3.4.2** 修改环境变量  
登录用户可以根据需要修改和设置环境变量。 Linux 下修改环境变量既可以在终端通过Shell 命令修改，也可以通过修改系统的配置文件来进行。

（ 1）通过 Shell 命令设置环境变量，常用于临时设置环境变量，一旦关闭当前终端或者新开一个终端，所设置的环境变量都将丢失。可直接用等号（ =）为变量赋值，或者用 export命令为变量赋值，用法：  
$ 变量=$变量:新增加变量值  
$ export 变量=$变量:新增加变量值  
新增加的变量值既可以放在变量原有值的末尾（ $变量:新变量值），也可以放在原有变量值的开头（新变量值:$变量）。  
例如， 需要为系统 PATH 变量增加/opt/usr/bin 目录，操作示例：  
vmuser@Linux-host: ~$ echo $PATH  
/usr/lib/qt-3.3/bin:/usr/kerberos/bin:/usr/local/ruby/bin:/opt/mysql5/bin:/usr/lib/ccache:/usr/local/bin:/bin:/usr/bin:/  
usr/local/sbin:/usr/sbin:/sbin  
[chenxibing@fedora ~]PATH=$PATH:/opt/usr/bin  
[chenxibing@fedora ~]echo $PATH  
/usr/lib/qt-3.3/bin:/usr/kerberos/bin:/usr/local/ruby/bin:/opt/mysql5/bin:/usr/lib/ccache:/usr/local/bin:/bin:/usr/bin:/  
usr/local/sbin:/usr/sbin:/sbin:/opt/usr/bin  
（ 2）修改系统配置文件。修改系统配置文件，可以达到永久改变环境变量的目的。修  
改某个配置文件后，在 Shell 下运行该文件即可使新的设置生效，或者重新登录使用新的变  
量。运行文件可用“ source 文件”的方式操作。通常修改/etc/profile 文件或者~/.bashrc（有  
的发行版上为~/.bash\_profile）文件：  
⚫修改/etc/profile 文件会影响使用本机的全部用户；  
⚫修改~/.bashrc 则仅仅影响当前用户；  
⚫推荐修改~/.bashrc 文件

# Linux系统VI的使用基础篇

**课程目标**

       了解VI的不同模式和特点

       熟练掌握VI的常用命令操作

**重点难点**

**考核目标**

**课后练习**

把课程中的内容演练一遍

       用VI敲一个java,c的程序

VI的基本介绍

**什么是Vi**

       它是Linux系统下的鼎鼎大名的全屏方式的文本编辑器，非常有用；

       Vi filename

       filename不存在则创建filename文件并可编辑和保存退出

       filename存在则打开文件编辑

**Vi使用的模式**

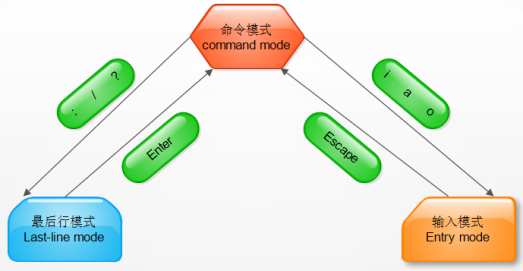
**三种模式**

       命令模式 Command Mode

       插入模式 Edit Mode

       末行模式 ExMode

**不同模式的切换**

****

**VI的常用命令**

这里指的编辑命令是指：插入（ i 或者 I）、附加（ a 或者 A）以及打开（ o 或者 O）命令。

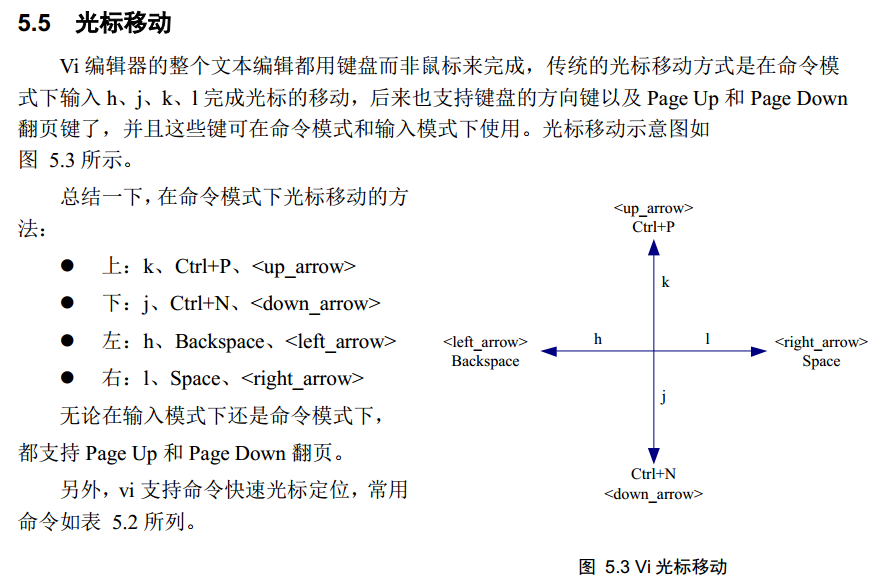
在命令模式下输入如表 5.1 所示的命令都可以退出 Vi 编辑器，回到 Shell 界面。

退出 Vi 的命令

|  |  |
| --- | --- |
| 命令 | 说明 |
| :q | 退出未被编辑过的文件 |
| :q! | 强行退出 vi，丢弃所做改动 |
| :x | 存盘退出 vi |
| :wq | 存盘退出 vi |
| ZZ | 等同于:wq |

VI在三种不同模式下有不同的命令实现不同的功能，具体如下:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 命令模式  Command mode | | 末行模式  Last-line mode | | 输入  模式  Entry mode |
| 分类 | 命令 | 分类 | 命令 |
| 光标移动 | k 向上  j 向下  h 向左  l 向右  0 移到到行首  $ 移动到行尾  G 移动到文件末尾  1G 移动到文件首  [n]G 移动到第[n]行  [n]+ 向下移动[n]行  [n]- 向上移动[n]行 | 保存退出 | :w 保存  :w newfile 存成新文件  :q 退出  :q! 强制退出不保存  :wq 存盘退出 |  |
| 删除 | X 删除光标位置的1个字符  dw 删除光标位置的当前词  [n]dw 删除从光标开始的n个词  dd 删除光标所在行  [n]dd 删除从光标开始的n行 | 行号设置与取消 | :set nu 设置行号  :set nonu 取消行号 |
|  |  |  | [n1],[n2]dd  删除n1到n2行 |
| 复制粘贴 | yw 复制光标所在的1个词  yy 复制光标所在行  [n]yy 复制从光标开始的n行  p 当前行下粘贴 | 拷贝移动 | :n1,n2con3  拷贝n1,n2行在n3行之后  :n1,n2mn3  移动n1,n2行到n3行之后 |



光标快速定位

|  |  |
| --- | --- |
| 命令 | 说明 |
| G | 将光标定位到最后一行 |
| nG | 将光标定位到第 n 行 |
| gg | 将光标定位到第 1 行 |
| ngg | 将光标定位到第 n 行 |
| :n | 将光标定位到第 n 行 |

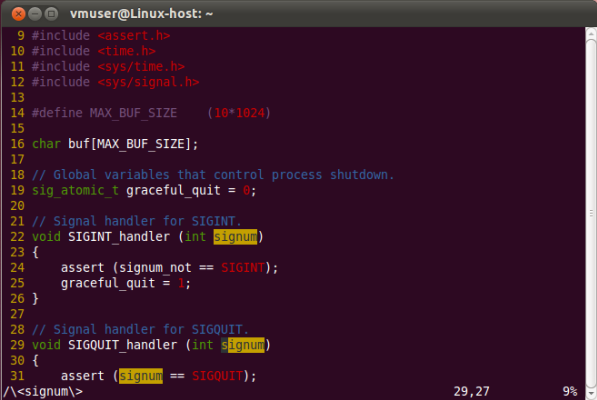
剪切和删除命令

|  |  |
| --- | --- |
| 命令 | 说明 |
| x 或 nx | 剪切从光标所在的位置开始的一个或 n 个字符 |
| X 或 nX | 剪切光标前的一个或 n 个字符 |
| dd | 删除光标所在的行 |
| D | 删除从光标位置开始至行尾 |
| dw | 删除从光标位置至该词末尾的所有字符 |
| d0 | 删除从光标位置开始至行首 |
| dnd | 删除光标所在行开始的 n 行 |
| dnG | 将光标所在行至第 n 行删除 |

文本查找  
在命令模式下，输入“ /字符串”即可从光标位置开始向下查找字符串，如输入“ /text”，即从光标所在位置向下开始查找 text 字符串。输入“ ?字符串”则从光标位置开始向上查找字符串。无论向下还是向上查找，查找下一个，按键盘 n 键即可。如图 5.6 所示是在 Vi 中搜索字符串 signun 得到的结果。



默认情况下搜索到的字符串不会高亮显示，在命令模式下输入“ :set hlsearch”可以实现高亮显示。  
用“ /字符串”或者“ ?字符串”方式搜索，将以局部匹配结果显示搜索结果，例如搜素字符串 abc，字符串 abcd 也将会被显示在搜索结果中。如图 5.6 所示，搜素 signum，而signum\_not 也被搜索到了。如果不希望将 abcd 列入显示结果中，则可用全局匹配搜素。方法，先将光标移动到字符串 abc，然后按“ SHIFT+\*”，完成搜索。如图 5.7 所示，用这种方法搜素字符串 signum，字符串 signum\_not 已经不在搜索结果中。



文本替换  
文本替换的命令稍微复杂一些，在命令模式下，输入：  
:%s /old/new/g  
能够将文本内全部的字符串 old 替换为 new。为了安全起见，可以在替换命令尾部加上 c，这样每次替换前都需要确认一下。  
**6. 撤销和恢复**在命令命令模式下输入 u，可撤销所做的更改，恢复编辑前的状态。这里的编辑以保存命令为界，例如，打开一篇文本，在编辑过程中被保存了 3 次，则可撤销 3 次。最多能撤销的次数由 Vi 的 undolevels 决定，一般是 500。不小心多按了 u 时，可以用 Ctrl+R 来恢复。注意，一旦文本被关闭，再次打开将无法使用 u 撤销所做更改。

配置 viVi 编辑器支持很多配置选项，如设置和取消行号、设置 TAB 键字符数、设置语法高亮等，在命令模式下输入“ :set”可以对 Vi 进行配置。在一般模式下，输入“ :set number”可以显示行号，输入“ :set nonumber”取消显示行号。常用的配置命令如表 5.5 所示。  
表 5.5 配置 vi 命令

|  |  |
| --- | --- |
| 命令 | 说明 |
| :set number | 显示行号 |
| :set ignorecase | 不区分大小写 |
| :set tabstop=n | 按下 tab 键则实际输入 n 个空格 |
| :set hlsearch | 搜索时高亮显示 |
| :syntax on | 开启语法高亮 |

灵活利用 Vi 的这些配置特性，在编程过程中能带来极大便利，如在 C 编程中，设置语法高亮，能极大减少编码中的书写错误。设置语法高亮和没有设置语法高亮的效果对比如图5.8 所示



再例如显示行号在编写代码的时候也非常有帮助，但是在用鼠标选择代码片段并复制的时候，如果开启了显示行号的话，会连同行号一起复制，如果不需要行号，可以在命令模式下输入“ :set nu!”将行号关闭。显示与不显示行号的效果如图 5.9 所示。



在 Vi 内执行配置命令的效果是临时的，关闭 Vi，再次打开 Vi，需要重新配置。 Vi 有自己的配置文件，可以是“ /etc/vim/vimrc”或者“ ~/.vimrc”。两者的区别是前者全局的，影响登录本机的全部用户，后者仅仅对当前用户有效。把配置命令放在配置内，每次启动 Vi 就会自动载入配置文件中的设置。如程序清单 5.1所示是一个简单的 Vi 配置文件范例。  
程序清单 5.1 vi/vim 配置文件范例  
" 在窗口标题栏显示文件名称  
set title  
" 编辑的时候将所有的 tab 设置为空格  
set tabstop=4  
" 设置自动对齐空格数  
set shiftwidth=4  
" 显示行号  
set number  
" 搜索时高亮显示  
set hlsearch  
" 不区分大小写  
set ignorecase  
" 语法高亮  
syntax on  
说明：以"开始的是注释。

文件对比  
在编码过程中，经常会用到文件对比功能。 Vim 包含了文件对比工具 vimdiff。用 vimdiff工具可以很容易实现文件对比。  
用法：  
vmuser@Linux-host: ~$ vimdiff file1 file2 file3  
vimdiff 可以同时进行 2 个以上文件的对比，但大多数情况下是进行两个文件的对比。  
如图 5.10 所示是两个有差异的文件的对比效果

