康康的Linux系统初识

编者 徐振康

**目录**

[第一章 Linux系统简介 5](#_Toc429509601)

[1.1 UNIX与Linux发展历史 5](#_Toc429509602)

[1.1.1 UNIX发展历史和发行版本 5](#_Toc429509603)

[1.1.2 Linux发展历史和发行版本 5](#_Toc429509604)

[1.2 开源软件简介 6](#_Toc429509605)

[1.3 Linux应用领域 6](#_Toc429509606)

[1.4 Linux学习方法 6](#_Toc429509607)

[第二章 Linux系统安装 7](#_Toc429509608)

[2.1 VMware虚拟机安装与使用 7](#_Toc429509609)

[2.2 系统分区 7](#_Toc429509610)

[2.3 Linux系统安装 8](#_Toc429509611)

[2.4 远程登录管理工具 8](#_Toc429509612)

[第三章 初学Linux经验谈 9](#_Toc429509613)

[3.1 学习Linux的注意事项 9](#_Toc429509614)

[3.2 服务器管理和维护建议 9](#_Toc429509615)

[第四章 Linux常用命令 11](#_Toc429509616)

[4.1 文件处理命令 11](#_Toc429509617)

[4.1.1 命令格式 11](#_Toc429509618)

[4.1.2 目录处理命令 11](#_Toc429509619)

[4.1.3 文件处理命令 13](#_Toc429509620)

[4.1.4 链接命令 15](#_Toc429509621)

[4.2 权限管理命令 15](#_Toc429509622)

[4.3 文件搜索命令 17](#_Toc429509623)

[4.4 帮助命令 19](#_Toc429509624)

[4.5 用户管理命令 20](#_Toc429509625)

[4.6 压缩解压命令 21](#_Toc429509626)

[4.7 网络命令 22](#_Toc429509627)

[4.8 关机重启命令 24](#_Toc429509628)

[第五章 文本编辑器Vim 26](#_Toc429509629)

[5.1 Vim常用操作 26](#_Toc429509630)

[5.2 Vim使用技巧 27](#_Toc429509631)

[第六章 Linux软件安装 30](#_Toc429509632)

[6.1 软件包管理简介 30](#_Toc429509633)

[6.2 RPM包管理-rpm命令管理 30](#_Toc429509634)

[6.2.1 包命名与依赖性 30](#_Toc429509635)

[6.2.2 安装升级与卸载 31](#_Toc429509636)

[6.2.3 RPM包查询 31](#_Toc429509637)

[6.2.4 校验与文件提取 32](#_Toc429509638)

[6.3 RPM包管理-yum在线管理 33](#_Toc429509639)

[6.3.1 IP地址配置和网络yum源 33](#_Toc429509640)

[6.3.2 yum命令 34](#_Toc429509641)

[6.3.3 光盘yum源搭建 35](#_Toc429509642)

[6.4源码包管理 35](#_Toc429509643)

[6.4.1 源码包和RPM包的区别 35](#_Toc429509644)

[6.4.2 源码包安装过程 36](#_Toc429509645)

[6.5脚本安装包 36](#_Toc429509646)

[第七章 用户和用户组管理 37](#_Toc429509647)

[7.1 用户配置文件 37](#_Toc429509648)

[7.1.1 用户信息文件/etc/passwd 37](#_Toc429509649)

[7.1.2 影子文件/etc/shadow 38](#_Toc429509650)

[7.1.3 组信息文件/etc/group和组密码文件/etc/gshadow 38](#_Toc429509651)

[7.2 用户管理相关文件 39](#_Toc429509652)

[7.3 用户管理命令 39](#_Toc429509653)

[7.3.1 用户添加命令useradd 39](#_Toc429509654)

[7.3.2 修改用户密码passwd 40](#_Toc429509655)

[7.3.3 修改用户信息usermod修改用户密码状态chage 40](#_Toc429509656)

[7.3.4 删除用户userdel用户切换命令su 41](#_Toc429509657)

[7.4 用户组管理命令 42](#_Toc429509658)

[第八章 权限管理 43](#_Toc429509659)

[8.1 ACL权限 43](#_Toc429509660)

[8.1.1 ACL权限简介与开启 43](#_Toc429509661)

[8.1.2 查看与设定ACL权限 44](#_Toc429509662)

[8.1.3 最大有效权限与删除ACL权限 45](#_Toc429509663)

[8.1.4 默认ACL权限和递归ACL权限 46](#_Toc429509664)

[8.2 文件特殊权限 46](#_Toc429509665)

[8.2.1 SetUID 46](#_Toc429509666)

[8.2.2 SetGID 47](#_Toc429509667)

[8.3.3 Sticky BIT 48](#_Toc429509668)

[8.3 文件系统属性chattr权限 49](#_Toc429509669)

[8.4 系统命令sudo权限 49](#_Toc429509670)

[第九章 文件系统管理 50](#_Toc429509671)

[9.1 回顾分区和文件系统 50](#_Toc429509672)

[9.2 文件系统常用命令 51](#_Toc429509673)

[9.2.1 df命令、du命令、fsck命令和dump2fs命令 51](#_Toc429509674)

[9.2.2 挂载命令 52](#_Toc429509675)

[9.2.3 挂载光盘与U盘 53](#_Toc429509676)

[9.2.4 支持NTFS文件系统 53](#_Toc429509677)

[9.3 fdisk分区 54](#_Toc429509678)

[9.3.1 fdisk命令分区过程 54](#_Toc429509679)

[9.3.2 分区自动挂载与fstab文件修复 55](#_Toc429509680)

[9.4 分配swap分区 55](#_Toc429509681)

[第十章 Shell基础 56](#_Toc429509682)

[10.1 Shell概述 56](#_Toc429509683)

[10.2 Shell脚本的执行方式 56](#_Toc429509684)

[10.3 Bash的基本功能 58](#_Toc429509685)

[10.3.1 历史命令与命令补全 58](#_Toc429509686)

[10.3.2 命令别名与常用快捷键 58](#_Toc429509687)

[10.3.3 输入输出重定向 59](#_Toc429509688)

[10.3.4 多命令顺序执行与管道符 60](#_Toc429509689)

[10.3.5 通配符与其他特殊符号 61](#_Toc429509690)

[10.4 Bash的变量 62](#_Toc429509691)

[10.4.1 用户自定义变量 62](#_Toc429509692)

[10.4.2 环境变量 63](#_Toc429509693)

[10.4.3 位置参数变量 64](#_Toc429509694)

[10.4.4 预定义变量 65](#_Toc429509695)

[10.5 Bash的运算符 66](#_Toc429509696)

[10.5.1 数值运算与运算符 66](#_Toc429509697)

[10.5.2 变量测试与内容替换 67](#_Toc429509698)

[10.6 环境变量配置文件 68](#_Toc429509699)

[10.6.1 环境变量配置文件简介 68](#_Toc429509700)

[10.6.2 环境变量配置文件作用 69](#_Toc429509701)

[10.6.3 其他配置文件和登录信息 69](#_Toc429509702)

[第十一章 Shell编程 70](#_Toc429509703)

[11.1 基础正则表达式 70](#_Toc429509704)

[11.2 字符截取命令 72](#_Toc429509705)

[11.3 字符处理命令 72](#_Toc429509706)

[11.4 条件判断 72](#_Toc429509707)

[11.5 流程控制 72](#_Toc429509708)

# 第一章 Linux系统简介

## 1.1 UNIX与Linux发展历史

### 1.1.1 UNIX发展历史和发行版本

#### UNIX发展历史

UNIX操作系统（尤尼斯），是一个强大的多用户、多任务操作系统，支持多种处理器架构，按照操作系统的分类，属于分时操作系统，最早由KenThompson、Dennis Ritchie和Douglas McIlroy于1969年在AT&T的贝尔实验室开发。目前它的商标权由国际开放标准组织所拥有，只有符合单一UNIX规范的UNIX系统才能使用UNIX这个名称，否则只能称为类UNIX（UNIX-like）。

#### UNIX主要发行版本

|  |  |  |
| --- | --- | --- |
| **操作系统** | **公司** | **硬件平台** |
| AIX | IBM | PowerPC |
| HP-UX | HP | PA-RISC |
| Solaris | Sun | SPARC |
| Linux | Red Hat Linux、Ubuntu Linux …… | IA(Intel、AMD、Cyrix、RISE……) |

### 1.1.2 Linux发展历史和发行版本

#### Linux 发展历史



创始人林纳斯·托瓦兹

Linux 操作系统的诞生、发展和成长过程始终依赖着五个重要支柱：[UNIX](http://baike.baidu.com/view/8095.htm) 操作系统、[MINIX](http://baike.baidu.com/view/390953.htm) 操作系统、[GNU](http://baike.baidu.com/view/36272.htm)计划、[POSIX](http://baike.baidu.com/view/209573.htm) 标准和Internet 网络。

1981 年[IBM](http://baike.baidu.com/view/1937.htm)公司推出微型计算机IBM PC。

1991年，[GNU计划](http://baike.baidu.com/view/755356.htm)已经开发出了许多工具软件，最受期盼的GNU C编译器已经出现，GNU的操作系统核心[HURD](http://baike.baidu.com/view/817940.htm)一直处于实验阶段，没有任何可用性，实质上也没能开发出完整的[GNU操作系统](http://baike.baidu.com/view/8355181.htm)，但是GNU奠定了Linux用户基础和开发环境。

1991年初，[林纳斯·托瓦兹](http://baike.baidu.com/view/117611.htm)开始在一台386sx兼容微机上学习minix操作系统。1991年4月，林纳斯·托瓦兹开始酝酿并着手编制自己的操作系统。

1991 年4 月13 日在comp.os.minix 上发布说自己已经成功地将bash 移植到了minix 上，而且已经爱不释手、不能离开这个shell 软件了。

1991年7月3日，第一个与Linux有关的消息是在comp.os.minix上发布的（当然此时还不存在Linux这个名称，当时林纳斯·托瓦兹的脑子里想的可能是FREAX，FREAX的英文含义是怪诞的、怪物、异想天开等）。

1991年的10月5日，林纳斯·托瓦兹在comp.os.minix[新闻组](http://baike.baidu.com/view/834.htm)上发布消息，正式向外宣布Linux内核的诞生（Freeminix-likekernel sources for 386-AT）。

1993年，大约有100余名程序员参与了Linux内核代码编写/修改工作，其中核心组由5人组成，此时Linux 0.99的代码大约有十万行，用户大约有10万左右。

1994年3月，Linux1.0发布，代码量17万行，当时是按照完全自由免费的协议发布，随后正式采用GPL协议。

1995年1月，Bob Young创办了RedHat（小红帽），以GNU/Linux为核心，集成了400多个源代码开放的程序模块，搞出了一种冠以品牌的Linux，即RedHat Linux,称为Linux"发行版"，在市场上出售。这在经营模式上是一种创举。

1996年6月，Linux 2.0内核发布，此内核有大约40万行代码，并可以支持多个处理器。此时的Linux 已经进入了实用阶段，全球大约有350万人使用。

1998年2月，以Eric Raymond为首的一批年轻的"老牛羚骨干分子"终于认识到GNU/Linux体系的产业化道路的本质，并非是什么自由哲学，而是市场竞争的驱动，创办了"Open Source Intiative"（开放源代码促进会）"复兴"的大旗，在互联网世界里展开了一场历史性的Linux产业化运动。

2001年1月，Linux 2.4发布，它进一步地提升了SMP系统的扩展性，同时它也集成了很多用于支持桌面系统的特性：USB，PC卡（PCMCIA）的支持，内置的即插即用，等等功能。

2003年12月，Linux 2.6版内核发布，相对于2.4版内核2.6在对系统的支持都有很大的变化。

2004年的第1月，SuSE嫁到了Novell，SCO继续顶着骂名四处强行“化缘”， Asianux， MandrakeSoft也在五年中首次宣布季度赢利。3月，SGI宣布成功实现了Linux操作系统支持256个Itanium 2处理器。

#### Linux内核版本

Linux内核官网:[www.kernel.org](file:///C:\Users\Kang\Desktop\www.kernel.org)

#### 内核版本说明

2.6.18

#### 目前最新的内核版本

#### Linux主要发行版本

发行版为许多不同的目的而制作, 包括对不同计算机结构的支持, 对一个具体区域或语言的本地化，实时应用，和嵌入式系统，甚至许多版本故意地只加入免费软件。已经有超过三百个发行版被积极的开发，最普遍被使用的发行版有大约十二个。

**[Fedora Core]**

Fedora Core（自第七版直接更名为Fedora）是众多 Linux 发行版之一。它是一套从Red Hat Linux发展出来的免费Linux系统。Fedora Core 的前身就是Red Hat Linux。Fedora是一个开放的、创新的、前瞻性的操作系统和平台，基于Linux。它允许任何人自由地使用、修改和重发布，无论现在还是将来。它由一个强大的社群开发，这个社群的成员以自己的不懈努力，提供并维护自由、开放源码的软件和开放的标准。Fedora 项目由 Fedora 基金会管理和控制，得到了 Red Hat, Inc. 的支持。Fedora 是一个独立的操作系统，是Linux的一个发行版，可运行的体系结构包括 x86(即i386-i686), x86\_64 和 PowerPC。

**[Debian]**

Debian Project诞生于1993年8月13日，它的目标是提供一个稳定容错的Linux版本。支持Debian的不是某家公司，而是许多在其改进过程中投入了大量时间的开发人员，这种改进吸取了早期Linux的经验。

Debian以其稳定性著称，虽然它的早期版本Slink有一些问题，但是它的现有版本Potato已经相当稳定了。这个版本更多的使用了 pluggable authentication modules (PAM)，综合了一些更易于处理的需要认证的软件（如winbind for Samba）。

Debian的安装完全是基于文本的，对于其本身来说这不是一件坏事。但对于初级用户来说却并非这样。因为它仅仅使用fdisk 作为分区工具而没有自动分区功能，所以它的磁盘分区过程令人十分讨厌。磁盘设置完毕后，软件工具包的选择通过一个名为dselect的工具实现，但它不向用户提供安装基本工具组（如开发工具）的简易设置步骤。最后需要使用anXious工具配置X Windows，这个过程与其他版本的X Windows配置过程类似。完成这些配置后，Debian就可以使用了。

Debian主要通过基于Web的论坛和邮件列表来提供技术支持。作为服务器平台，Debian提供一个稳定的环境。为了保证它的稳定性，开发者不会在其中随意添加新技术，而是通过多次测试之后才选定合适的技术加入。当前最新正式版本是Debian 6，采用的内核是Linux 2.6.32。Debian 6 第一次 包含了一个100%开源的Linux内核，这个内核中不再包含任何闭源的硬件驱动。所有的闭源软件都被隔离成单独的软件包，放到Debian软件源的 "non-free" 部分。由此，Debian用户便可以自由地选择是使用一个完全开源的系统还是添加一些闭源驱动。

**[Mandrake]**

MandrakeSoft，Linux Mandrake的发行商，在1998年由一个推崇Linux的小组创立，它的目标是尽量让工作变得更简单。最终，Mandrake给人们提供了一个优秀的图形安装界面，它的最新版本还包含了许多Linux软件包。

作为Red Hat Linux的一个分支，Mandrake将自己定位在桌面市场的最佳Linux版本上。但该公司还是支持服务器上的安装，而且成绩并不坏。Mandrake的安装非常简单明了，为初级用户设置了简单的安装选项。它完全使用GUI界面，还为磁盘分区制作了一个适合各类用户的简单GUI界面。软件包的选择非常标准，另外还有对软件组和单个工具包的选项。安装完毕后，用户只需重启系统并登录进入即可。

Mandrake主要通过邮件列表和Mandrak 自己的Web论坛提供技术支持。Mandrak对桌面用户来说是一个非常不错的选择，它还可作为一款优秀的服务器系统，尤其适合Linux新手使用。它使用最新版本的内核，拥有许多用户需要在Linux服务器环境中使用的软件——数据库和Web服务器。

Mandrak没有重大的软件缺陷，只是它更加关注桌面市场，较少关注服务器市场。

**[Ubuntu]**

Ubuntu是一个以桌面应用为主的Linux操作系统，其名称来自非洲南部祖鲁语或豪萨语的“ubuntu”一词（译为吾帮托或乌班图），意思是“人性”、“我的存在是因为大家的存在”，是非洲传统的一种价值观，类似华人社会的“仁爱”思想。Ubuntu基于Debian发行版和GNOME桌面环境，与Debian的不同在于它每6个月会发布一个新版本。Ubuntu的目标在于为一般用户提供一个最新的、同时又相当稳定的主要由自由软件构建而成的操作系统。Ubuntu具有庞大的社区力量，用户可以方便地从社区获得帮助。随着云计算的流行，ubuntu推出了一个云计算环境搭建的解决方案，可以在其官方网站找到相关信息。于2012年4月26日发布最终版ubuntu 12.04，ubuntu 12.04是长期支持的版本。

**[Red Hat Linux]**

可能这是最著名的Linux版本了，Red Hat Linux已经创造了自己的品牌，越来越多的人听说过它。Red Hat在1994年创业，当时聘用了全世界500多名员工，他们都致力于开放的源代码体系。

Red Hat Linux是公共环境中表现上佳的服务器。它拥有自己的公司，能向用户提供一套完整的服务，这使得它特别适合在公共网络中使用。这个版本的Linux也使用最新的内核，还拥有大多数人都需要使用的主体软件包。

Red Hat Linux的安装过程也十分简单明了。它的图形安装过程提供简易设置服务器的全部信息。磁盘分区过程可以自动完成，还可以选择GUI工具完成，即使对于 Linux新手来说这些都非常简单。选择软件包的过程也与其他版本类似；用户可以选择软件包种类或特殊的软件包。系统运行起来后，用户可以从Web站点和 Red Hat那里得到充分的技术支持。我发现Red Hat是一个符合大众需求的最优版本。在服务器和桌面系统中它都工作得很好。Red Hat的唯一缺陷是带有一些不标准的内核补丁，这使得它难于按用户的需求进行定制。 Red Hat通过论坛和邮件列表提供广泛的技术支持，它还有自己公司的电话技术支持，后者对要求更高技术支持水平的集团客户更有吸引力。

**[SuSE]**

总部设在德国的SuSE AG在商界已经奋斗了8年多，它一直致力于创建一个连接数据库的最佳Linux版本。为了实现这一目的，SuSE与Oracle 和IBM合作，以使他们的产品能稳定地工作。SuSE还开发了SuSE Linux eMail Server III，一个非常稳定的电子邮件群组应用。

基于2.4.10内核的SuSE 7.3，在原有版本的基础上提高了易用性。安装过程通过GUI完成，磁盘分区过程也非常简单，但它没有为用户提供更多的控制和选择。

在SuSE 操作系统下，可以非常方便地访问Windows磁盘，这使得两种平台之间的切换，以及使用双系统启动变得更容易。SuSE的硬件检测非常优秀，该版本在服务器和工作站上都用得很好。

SuSE拥有界面友好的安装过程，还有图形管理工具，可方便地访问Windows磁盘，对于终端用户和管理员来说使用它同样方便，这使它成为了一个强大的服务器平台。 SuSE也通过基于Web的论坛提供技术支持，另外我还发现它有电话技术支持。

**[Linux Mint]**

Linux Mint是一份基于Ubuntu的发行版，其目标是提供一种更完整的即刻可用体验，这包括提供浏览器插件、多媒体编解码器、对DVD播放的支持、Java和其他组件。它与Ubuntu软件仓库兼容。Linux Mint 是一个为pc和X86电脑设计的操作系统。

因此，一个可以跑得动Windows的电脑也可以使用Linux Mint来代替Windows，或者两个都跑。既有Windows又有Linux的系统就是传说中的“双系统”。同样，MAC，BSD或者其他的Linux版本也可以和Linux Mint 共存。一台装有多系统的电脑在开机的时候会出现一个供你选择操作系统的菜单。Linux Mint可以很好的在一个单系统的电脑上运行，但是它也可以自动检测其他操作系统并与其互动，例如，如果你安装Linux Mint在一个安装了Windows版本的（xp，vista或者其他版本），它会自动检测并建立双启动以供您在开机的时候选择启动哪个系统。并且你可以在Linux Mint下访问Windows分区。Linux是更安全，更稳定，更有效并且日益易于操作的甚至可以和Windows相媲美的系统，它越来越让人感到难以抉择了。

**[Gentoo]**

Gentoo是Linux世界最年轻的发行版本，正因为年轻，所以能吸取在她之前的所有发行版本的优点。Gentoo最初由Daniel Robbins（FreeBSD的开发者之一）创建，首个稳定版本发布于2002年。由于开发者对FreeBSD的熟识，所以Gentoo拥有媲美FreeBSD的广受美誉的ports系统 ——Portage包管理系统。

**[centos]**

CentOS（Community ENTerprise Operating System）是Linux发行版之一，它是来自于Red Hat Enterprise Linux依照开放源代码规定释出的源代码所编译而成。由于出自同样的源代码，因此有些要求高度稳定性的服务器以CentOS替代商业版的Red Hat Enterprise Linux使用。两者的不同，在于CentOS并不包含封闭源代码软件,CentOS 是一个基于Red Hat Linux 提供的可自由使用源代码的企业级Linux发行版本。每个版本的 CentOS都会获得十年的支持（通过安全更新方式）。新版本的 CentOS 大约每两年发行一次，而每个版本的 CentOS 会定期（大概每六个月）更新一次，以便支持新的硬件。这样，建立一个安全、低维护、稳定、高预测性、高重复性的 Linux 环境。CentOS是Community Enterprise Operating System的缩写。

CentOS 是RHEL（Red Hat Enterprise Linux）源代码再编译的产物，而且在RHEL的基础上修正了不少已知的 Bug ，相对于其他 Linux 发行版，其稳定性值得信赖。

RHEL 在发行的时候，有两种方式。一种是二进制的发行方式，另外一种是源代码的发行方式。

## 1.2 开源软件简介

#### 开源软件

（1）使用的自由

绝大多数开元软件免费

（2）研究的自由

可以获得软件源代码

（3）散布及改良的自由

可以自由传播、改良甚至销售

#### 支撑互联网的开源技术

LAMP(Linux操作系统+ApacheWeb服务器+MySQL数据库+PHP编程语言)

## 1.3 Linux应用领域

（1）基于Linux的企业服务器

**1) Google：Goobuntu，内部使用的Linux定制系统**

Google可能是使用Linux桌面系统的最著名公司，它内部使用的Goobuntu是Ubuntu长期支持版本（LTS）的改进版本。如果Google工作人员要使用Windows操作系统，必须要特别申请才行。据Google内部项目负责人介绍，Windows存在“特殊”的安全问题，所以员工使用Windows操作系统必须获得上级负责人许可。

Goobuntu的用户数量近万名，使用人群从设计师、销售人员到工程师、部门主管，有对UNIX系统驾轻就熟的员工，也有对计算机知之甚少的员工。

**2) NASA**

NASA负责人介绍说，之所以将国际空间站的关键部分操作系统从Windows切换到Linux，主要是出于稳定性和可靠性的考虑，Linux的可定制性也考虑在内，因为NASA科学家可以灵活地对空间站操作系统进行调整、安装补丁。

**3) 法国警察部队**

法国准军事警察部队（国家宪兵），与城市慕尼黑一样，也花了将近10年多的时间将办公系统切换到Linux和开源软件上来。整个迁移项目在今年最终完成，72000台PC由Windows XP迁移到GendBuntu——一个基于Ubuntu LTS定制的系统。

除了迁移操作系统，警察部队也将办公套件从MicroSoft Office迁移到OpenOffice，浏览器从IE切换为Firefox，采用Thunderbird来收发邮件，同样也采用了图像编辑器软件GIMP、视频播放器VLC。GendBuntu操作系统的迁移采取了分步实施，先是在5000台PC上进行测试，接着，又在30000台机子上进行正式部署实施，以至最终全部迁移过来。

**4.美国国防部**

美国国防部出于隐私保护方面的考虑，自己开发了一个基于Linux的U盘操作系统，可以帮助政府人员在不受信任的电脑上安全使用网络。该系统可以直接安装在CD或U盘上，插拔方便，不会在电脑硬盘上留下活动痕迹。

**5. CERN（欧洲核研究实验室）**

欧洲核研究实验室CERN也是一个长期的Linux用户。CERN的研究员们使用的3000多台电脑都运行着定制的Scientific Linux。

（2）嵌入式应用

1）手机、平板电脑

2）其他嵌入式应用

智能家电：机顶盒、游戏机、数码相机

智能卡系统

航空系统

银行系统

卡拉OK点歌系统

……

（3）Linux在电影娱乐业

## 1.4 Linux学习方法

（1）为什么选择命令行

实际上，许多命令的功能都可以通过图形化界面来实现，学习这些命令的意义在哪里呢？

在UNIX发育的大部分历史上，用户都是通过shell来工作的。大部分命令都已经经过了几十年的发展和改良，功能强大，性能稳定。Linux继承自UNIX，自然也是如此。此外Linux的图形化界面并不好，并不是所有的命令都有对应的图形按钮。更别说在图形化界面崩溃的情况下，你就要靠shell输入命令来恢复计算机了。

命令本身是一个函数 (function)，是一个小的功能模块。当我们想要让计算机做很复杂的事情 (比如说: 在晚上12:00下载某个页面的所有链接，然后复制到移动硬盘)的时候，不断地去按各个图形化按钮并不是个很聪明的事情 (1. 要点很多下，2. 必须等到12:00)。我们通常是通过shell编程来实现这样一些复杂任务，这时，就可以把命令作为函数，嵌入到我们的shell程序中, 从而让不同命令协同工作 (比如使用date来查询时间，再根据时间来使用wget下载等等)。

（2）解决问题的智慧

帮助、文档、示例、查找

（3）英文提示的困惑

Command not found

No Such file or directory

（4）忘掉Windows的思维方式

计划、专注、坚持

# 第二章 Linux系统安装

## 2.1 VMware虚拟机安装、卸载与使用

（一）安装

（1）首先下载VMware软件

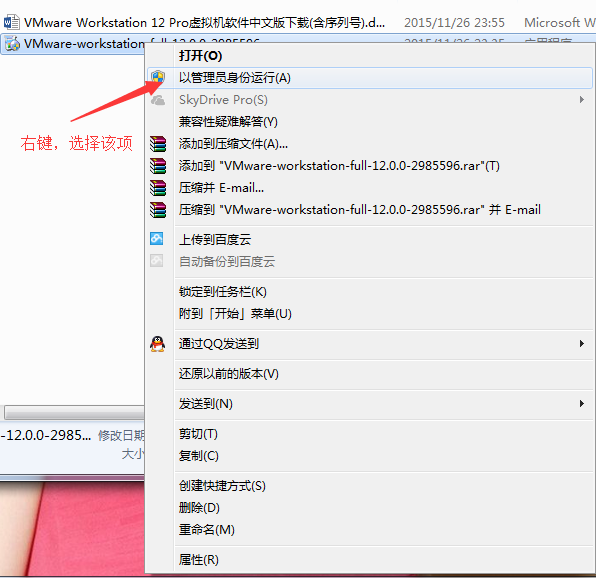
VMware官方下载地址http://www.vmware.com/。由于官网下载速度较慢而且没有破解密钥，所以，下面给出对应的软件官方文件以及破解密钥的百度云分享链接。

VMware软件的下载地址如下：

链接：http://pan.baidu.com/s/1V6TNg 密码：gd9m

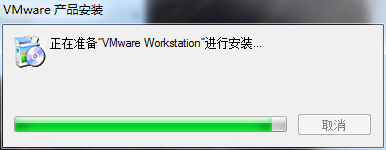
（2）VMware软件的安装

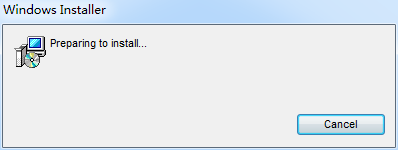
打开下载好的文件夹，右键VMware-workstation-full-12.0.0-2985596.exe，以管理员身份运行；



然后会依次弹出这么几个框框；



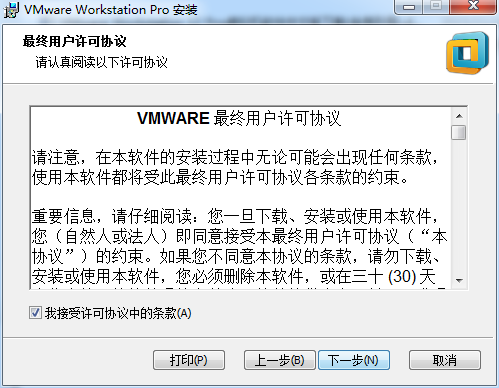




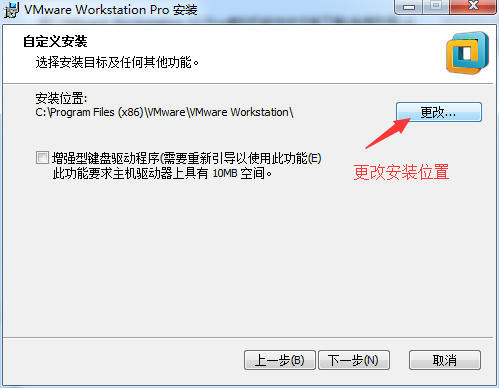
点击下一步；



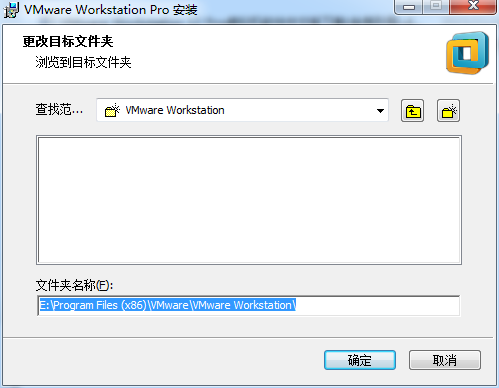
接受协议，下一步；



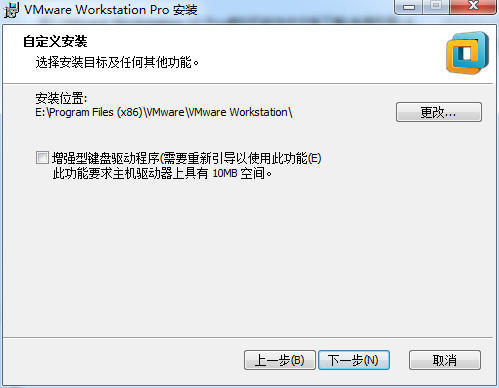
更改安装位置；



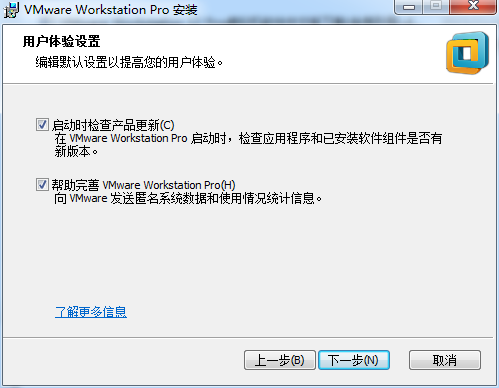
这里修改为：E:\Program Files (x86)\VMware\VMware Workstation\，确定；



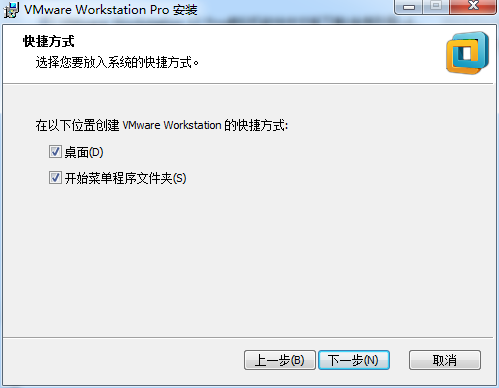
下一步；



下一步；



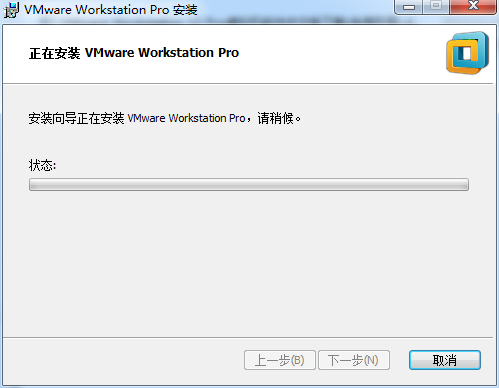
下一步；



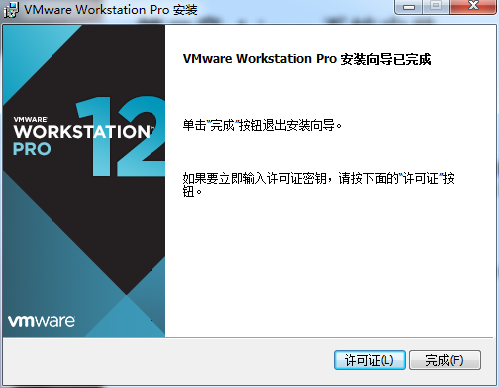
下一步；



安装；



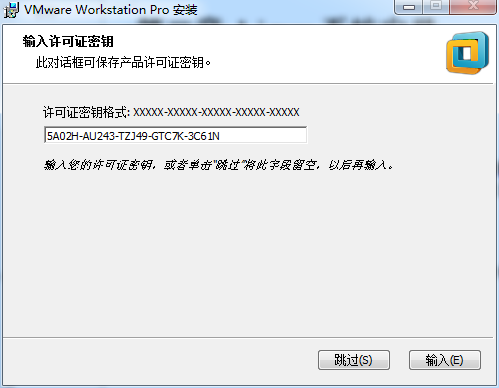
点击许可证；



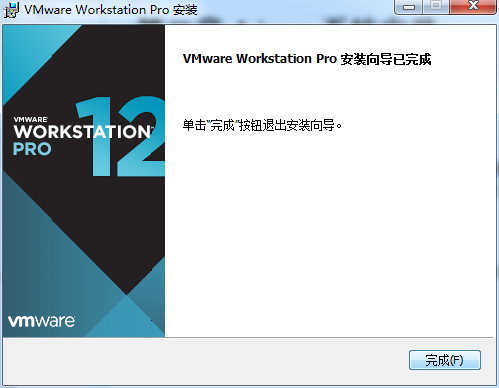
打开“VMware Workstation 12 Pro虚拟机软件中文版下载(含序列号).docx”，找到



粘贴到对应的位置，点击输入，如图；



完成。

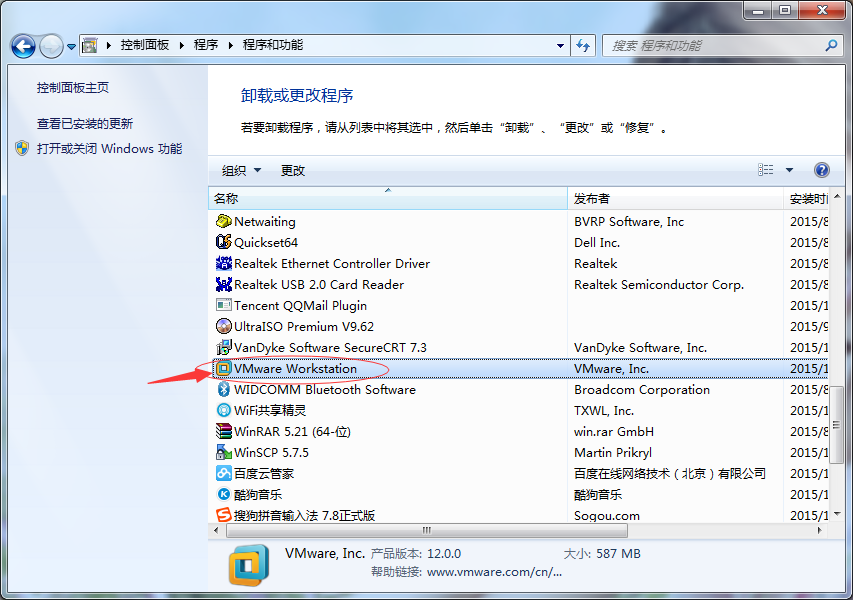


（二）卸载

点击开始菜单，控制面板，然后选择卸载程序，如图；



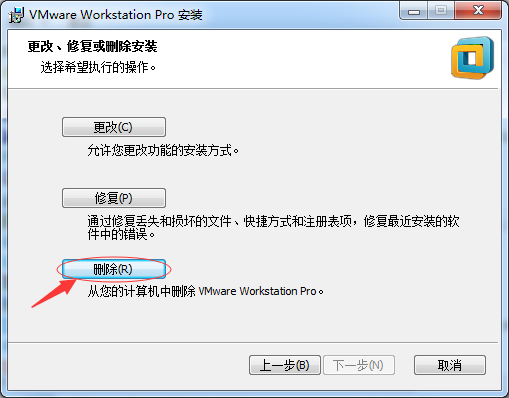
找到VMware Workstation，双击；



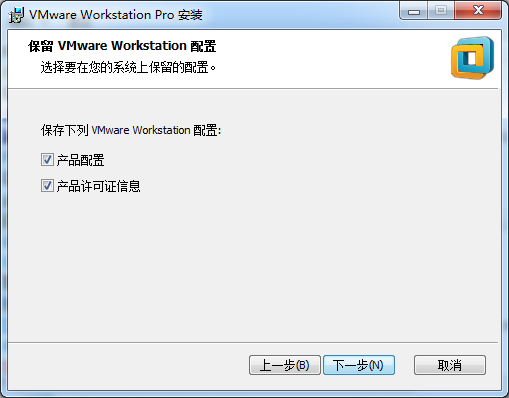
点击下一步；



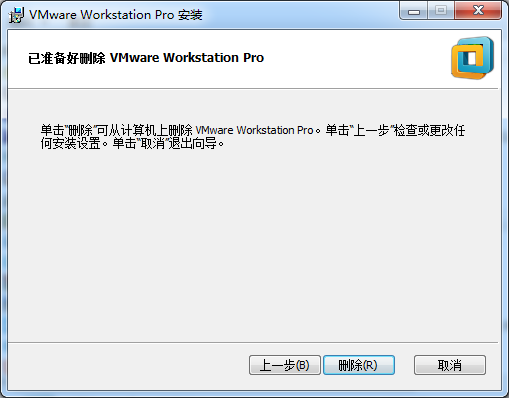
选择删除；



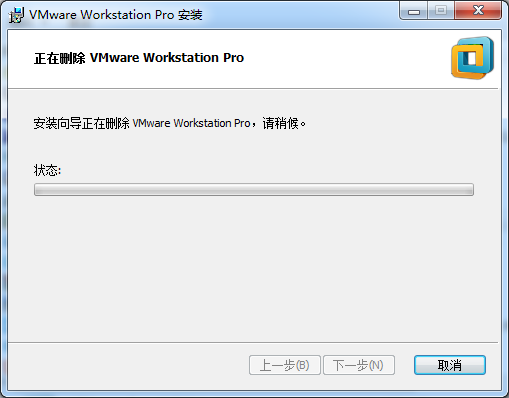
可以保留对应的产品配置和产品许可证信息， 方便下次安装，也可以不保留，卸载干净，然后选择下一步；



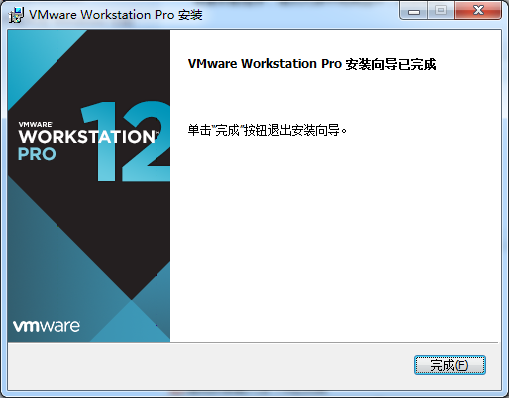
点击删除；



正在删除……

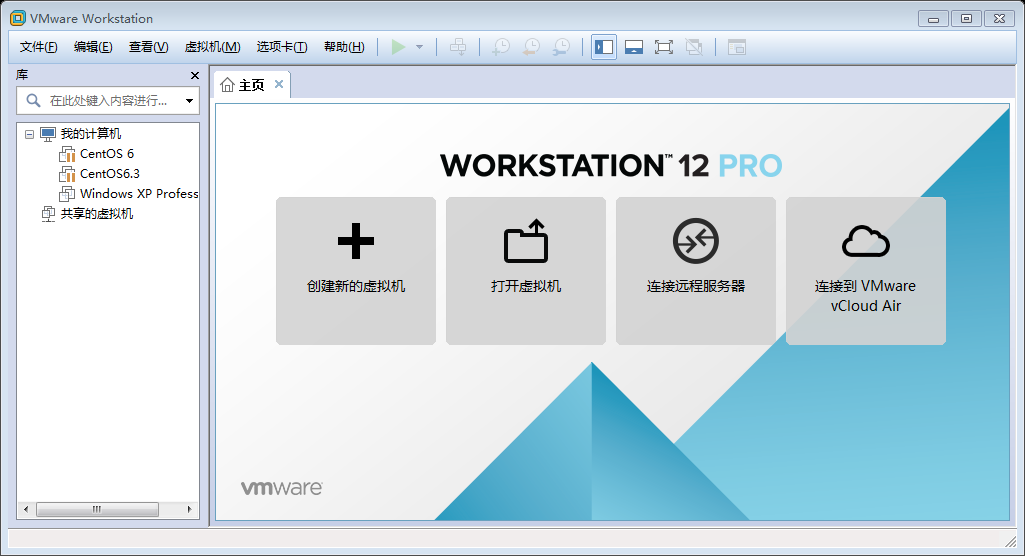


卸载完成；

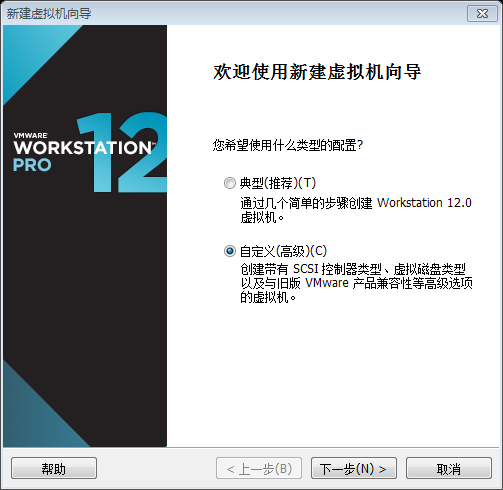


（三）在虚拟机上新建一台新的虚拟机

（1）点击创建新的虚拟机

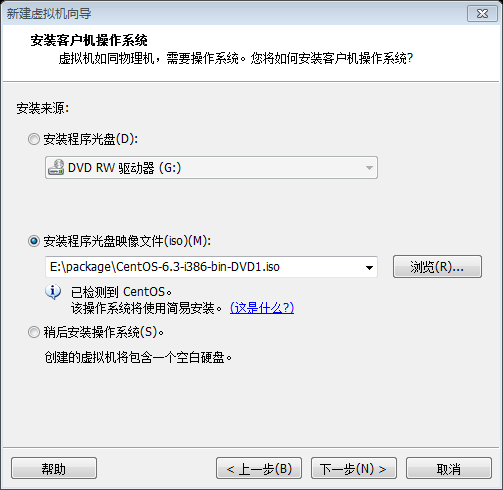


选择自定义，下一步；

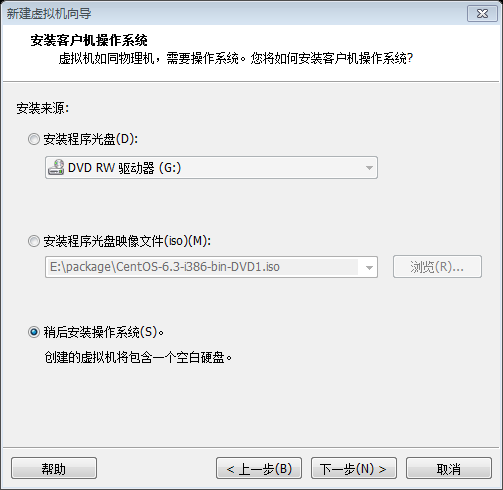




下一步，注意此处有两种安装方法，您可以任选一种：1.选择安装程序光盘镜像文件iso(M):，选择浏览，找到该iso文件的位置，注意要选择CentOS-6.3-i386-bin-DVD1.iso文件的位置，第一张盘中存放的是centOS6.3的系统文件以及相关的软件包，第二张盘中存放的是其他一些软件包（这里的iso文件是linux系统的文件镜像，本教程采用centOS6.3， 官方下载地址：http://vault.centos.org/6.3/isos/i386/CentOS-6.3-i386-bin-DVD1.iso, http://vault.centos.org/6.3/isos/i386/CentOS-6.3-i386-bin-DVD2.iso），选择好后，就会被检测到CentOS系统，可以简易安装，然后下一步， 后续步骤特别简单，填写Linux全名，密码等等，几乎是傻瓜式的这里不再详述；2.选择稍后安装操作系统(S)，这种方式是我们要细说的；

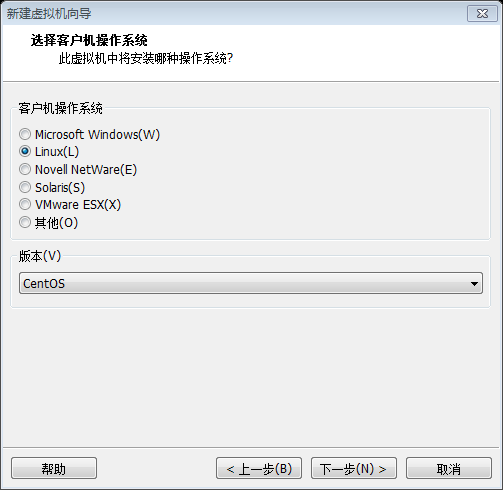


方式1：后续步骤不再详述

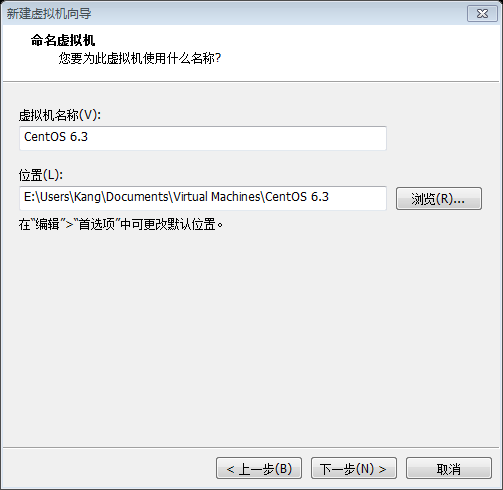


方式2

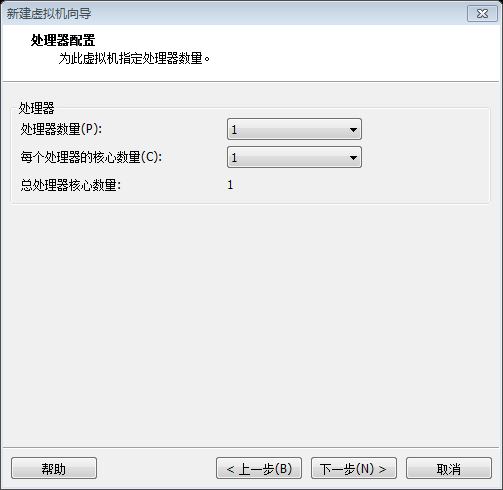
采用第二种方式按下一步之后，选择Linux，版本CentOS；



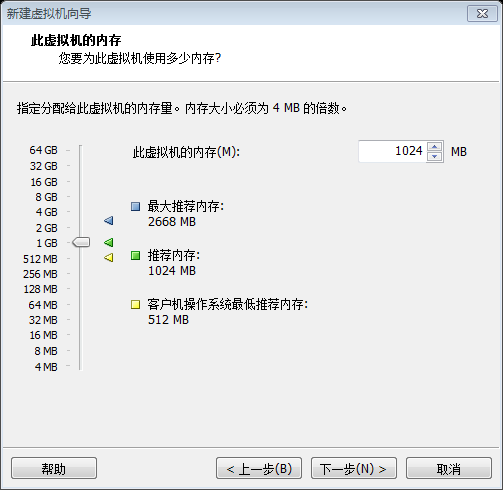
下一步之后填写虚拟机名称：这里填写CentOS 6.3，位置我保存在E:\Users\Kang\Documents\Virtual Machines\CentOS 6.3，下一步；



下一步；



再下一步；



然后出现如下页面，这里说明一下桥接，nat和主机模式的区别：

**桥接模式**：

在桥接模式下，VMWare虚拟出来的操作系统就像是局域网中的一台独立的主机（主机和虚拟机处于对等地位），它可以访问网内任何一台机器。在桥接模式下，我们往往需要为虚拟主机配置ＩＰ地址、子网掩码等（注意虚拟主机的ｉｐ地址要和主机ｉｐ地址在同一网段）。

使用桥接模式的虚拟系统和主机的关系，就如同连接在一个集线器上的两台电脑；要让它们通讯就需要为虚拟系统配置ip地址和子网掩码。如果我们需要在局域网内建立一个虚拟服务器，并为局域网用户提供服务，那就要选择桥接模式。

但对于想进行种种网络实验的朋友这种模式是不太适合的，因为你无法对虚拟机的网络进行控制，它直接出去了。

**NAT模式**：

在NAT模式下，虚拟系统需要借助NAT(网络地址转换)功能，通过宿主机器所在的网络来访问公网。也就是说，使用NAT模式虚拟系统可把物理主机作为路由器访问互联网。NAT模式下的虚拟系统的TCP/IP配置信息是由VMnet8(NAT)虚拟网络的DHCP服务器提供的，无法进行手工修改，因此虚拟系统也就无法和本局域网中的其他真实主机进行通讯。采用NAT模式最大的优势是虚拟系统接入互联网非常简单，你不需要进行任何其他的配置，只需要宿主机器能访问互联网即可。

这种方式也可以实现Host OS与Guest OS的双向访问。但网络内其他机器不能访问Guest OS，Guest OS可通过Host OS用NAT协议访问网络内其他机器。NAT方式的IP地址配置方法是由VMware的虚拟DHCP服务器中分配一个IP ，在这个IP地址中已经设置好路由，就是指向192.168.138.1的。如果你想利用VMWare安装一个新的虚拟系统，在虚拟系统中不用进行任何手工配置就能直接访问互联网，建议你采用NAT模式。

这种方式下host内部出现了一个虚拟的网卡vmnet8（默认情况下），如果你有过 做nat服务器的经验，这里的vmnet8就相当于连接到内网的网卡，而虚拟机本身则相当于运 行在内网上的机器，虚拟机内的网卡（eth0）则独立于vmnet8。

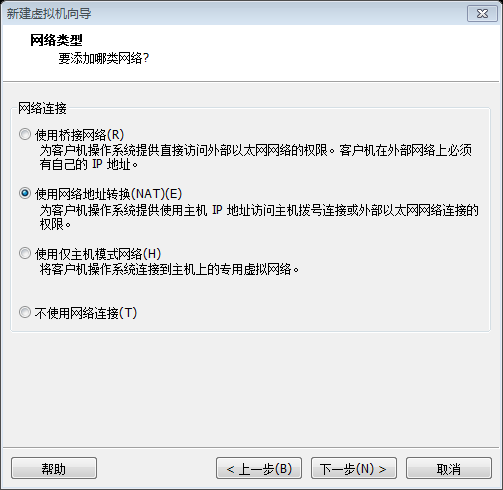
你会发现在这种方式下，vmware自带的dhcp会默认地加载到vmnet8界面上，这样虚拟机就可以使用dhcp服务。更为重要的是，vmware自带了nat服务，提供了从vmnet8到外网的地址转 换，所以这种情况是一个实实在在的nat服务器在运行，只不过是供虚拟机用的。很显然，如果你只有一个外网地址，此种方式很合适。

**主机模式**：

主机模式下，真实环境和虚拟环境是隔离开的；在这种模式下，所有的虚拟系统是可以相互通信的，但虚拟系统和真实的网络是被隔离开。（虚拟系统和宿主机器系统是可以相互通信的，相当于这两台机器通过双绞线互连。）

这应该是最为灵活的方式，有兴趣的话可以进行各种网络实验。和nat唯一的不同的是，此种方式下，没有地址转换服务，因此，默认情况下，虚拟机只能到主机访问。

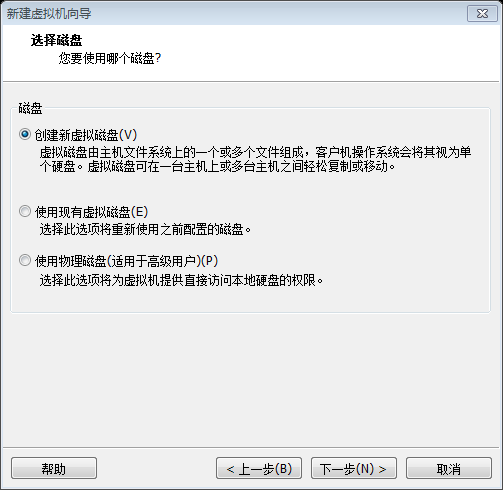
我们这里先选择NAT模式，即默认的；



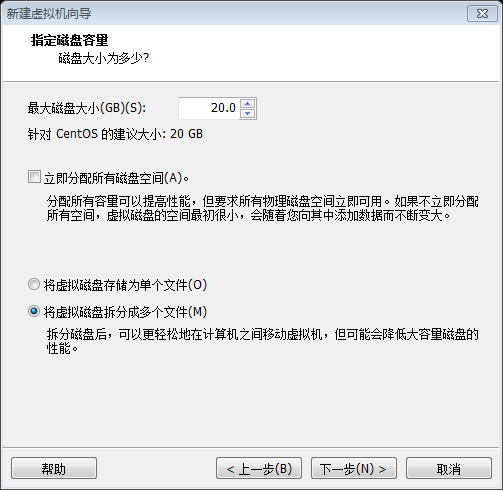
下一步；



再下一步；



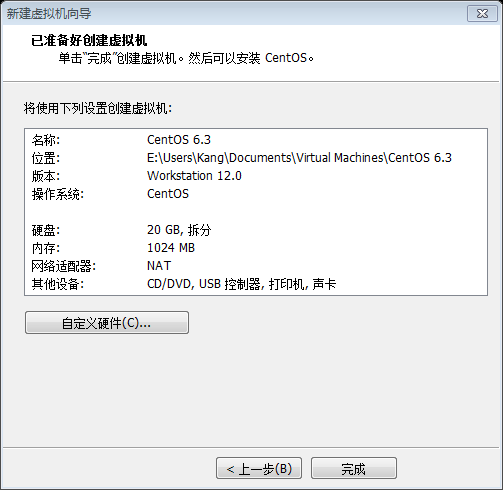
下一步，先使用推荐的20GB作为磁盘大小；



下一步， 选择磁盘文件的保存位置：我这里选择：E:\Users\Kang\Documents\Virtual Machines\CentOS 6.3，下一步；



然后完成；



## 2.2 系统分区

磁盘分区是使用分区编辑器(partition editor)在磁盘上划分几个逻辑部分。碟片一旦划分成数个分区(Partition)，不同类的目录与文件可以存储进不同的分区。

#### 分区类型

（1）主分区：最多只能有4个

（2）扩展分区：最多只能有1个；主分区加扩展分区最多有4个；不能写入数据，只能包含逻辑分区。

（3）逻辑分区

#### 格式化

格式化（高级格式化）又称逻辑格式化，它是指根据用户选定的文件系统（如FAT16、FAT32、NTFS、EXT2、EXT3、EXT4等），在磁盘的特定域写入特定数据，在分区中划出一片用于存放文件分配表、目录表等用于文件管理的磁盘空间。

#### 硬件设备文件名

|  |  |
| --- | --- |
| **硬件** | **设备文件名** |
| IDE硬盘 | /dev/hd[a-d] |
| SCSI/SATA/USB硬盘 | /dev/sd[a-p] |
| 光驱 | /dev/cdrom或/dev/hdc |
| 软盘 | /dev/fd[0-1] |
| 打印机(25针) | /dev/lp[0-2] |
| 打印机(USB) | /dev/usb/lp[0-15] |
| 鼠标 | /dev/mouse |

/dev/hda1(IDE硬盘接口)

/dev/sda1(SCSI硬盘接口、SATA硬盘接口)

#### 挂载

（1）必须分区

/ (根分区)

swap分区 （交换分区，内存2倍，不超过2GB）

（2）推荐分区

/boot （启动分区，200MB）

（3）建议的分区方案

跟分区分配最大、swap分区1GB、启动分区200MB、/home分区自定义

分区：把大硬盘分为小的逻辑分区

格式化：写入文件系统

分区设备文件名：给每个分区定义设备文件名

挂载：给每个分区分配挂载点

## 2.3 Linux系统安装

#### 安装欢迎界面

#### 密码原则

（1）复杂性

1）八位字符以上、大小写字母、数字、符号

2）不能是英文单词

3）不能是和用户相关的内容

（2）易记忆性

（3）时效性

#### 软件包选择

#### 安装日志

/root/install.log:存储了安装在系统中的软件包及其版本信息

/root/install.log.syslog:存储了安装过程中留下的事件记录

/root/anaconda-ks.cfg:以Kickstart配置文件的格式记录安装过程中设置的选项信息

## 2.4 远程登录管理工具

虚拟机网络配置

（1）SecureCRT

（2）Winscp

# 第三章 初学Linux经验谈

## 3.1 学习Linux的注意事项

（1）Linux严格区分大小写

（2）Linux中所有内容以文件形式保存，包括硬件

硬盘文件是/dev/sd[a-p]

光盘文件是/dev/sr0等

（3）Linux不靠扩展名区分文件类型

压缩包：“\*.gz”、 “\*.bz2”、 “\*.tar.bz2”、 “\*.tgz”等

二进制软件包：“.rpm”

网页文件：“\*.html”、“\*.php”

脚本文件：“\*.sh”

配置文件：“\*.conf”

（4）Linux所有的存储设备都必须挂载后用户才能使用，包括硬盘、U盘和光盘

（5）Windows下的程序不能直接在Linux中安装和运行

## 3.2 服务器管理和维护建议

#### Linux个目录的作用

|  |  |
| --- | --- |
| **目录名** | **目录作用** |
| /bin/ | 存放系统命令的目录，普通用户和超级用户都可以执行。放在/bin/下的命令在单用户模式下也可以执行 |
| /sbin/ | 保存和系统环境设置相关的命令，只有超级用户可以使用这些命令进行系统环境设置，但是有些命令可以允许普通用户查看 |
| /usr/bin/ | 存放系统命令的目录，普通用户和超级用户都可以执行。这些命令和系统启动无关，在单用户模式下不能执行 |
| /usr/sbin/ | 存放根文件系统不必要的系统管理命令，例如多数服务程序。只有超级用户可以使用。大家其实可以住到Linux的系统，在所有“sbin”目录中保存的命令只有超级用户可以使用，“bin”目录中保存的命令所有用户都可以使用 |
| /boot/ | 系统启动目录，保存系统启动相关的文件，如内核文件和启动引导程序（grub）文件等 |
| /dev/ | 设备文件保存位置。我们已经说过Linux中所有内容以文件形式保存，包括硬件。那么这个目录就是用来保存所有设备文件的 |
| /etc/ | 配置文件保存位置。系统内所有采用默认安装（rpm安装）的服务的配置文件全部都保存在这个目录中，如用户账户和密码，服务的启动脚本，常用服务的配置文件等 |
| /home/ | 普通用户的家目录。建立每个用户时，每个用户要有一个默认登录位置，这个位置就是这个用户的家目录，所有普通用户的家目录就是在/home/下建立一个和用户名相同的目录。如用户user1的家目录就是/home/user1 |
| /lib/ | 系统调用的函数库保存位置 |
| /lost+found/ | 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，病修复已经损坏的文件系统。这个目录只在每个分区中出现，例如/lost+found/就是根分区的备份恢复目录，/boot/lost+found/就是/boot/分区的备份恢复目录 |
| /media/ | 挂载目录。系统建议是用来挂载媒体设备的，例如软盘和光盘 |
| /mnt/ | 挂载目录，早起Linux中只有这个挂载目录，并没有细分。现在这个目录系统建议挂载额外设备，如U盘，移动硬盘和其他操作系统的分区 |
| /misc/ | 挂载目录。系统建议用来挂载NFS服务的共享目录。我们在刚刚已经解释了挂载，童鞋们应该知道只要是一个已经建立的空目录就可以作为挂载点。那么系统虽然准备了三个默认挂载目录/media/、/mnt/、/misc/，但是到底在哪个目录中挂载什么设备都可以由管理员自己决定。 |
| /opt/ | 第三方安装的软件保存位置。这个目录就是放置和安装其他软件的位置，我手工安装的源码包软件都可以安装到这个目录当中。不过我还是更加习惯抱软件放置到/usr/local/目录当中，也就是说/usr/local/目录也是用来安装软件的 |
| /proc/ | 虚拟文件系统，该目录中的数据并不保存到硬盘中，而是保存到内存当中。主要保存系统的内核，进程，外部设备状态和网络状态等。如/proc/cpuinfo是保存CPU信息的，/proc/devices是保存设备驱动的列表的,/proc/filesystems是保存文件系统列表的，/proc/net是保存网络协议信息的 |
| /sys/ | 虚拟文件系统。和/proc目录相似，都是保存在内存当中的，主要是保存于内核相关信息的 |
| /root/ | 超级用户的家目录。普通用户家目录在“/home”下，超级用户家目录直接放在“/”下 |
| /srv/ | 服务数据目录。一些系统服务启动之后，可以在这个目录中保存所需要的数据 |
| /tmp/ | 临时目录。系统存放临时文件的目录，该目录下所有用户都可以访问和写入。我们建议次目录中不能保存重要数据，最好每次开机都该目录清空 |
| /usr/ | 系统软件资源目录。注意usr不是user的缩写，而是“Unix Software Resource”的缩写，所以不是存放用户数据，而是存放系统软件资源的目录。系统中安装的软件大多是保存在这里 |
| /var/ | 动态数据保存位置。主要保存缓存、日志以及软件运行所产生的文件 |

#### 服务器注意事项

（1）远程服务器不允许关机，只能重启

（2）重启时应该关闭服务

（3）不要在服务器访问高峰运行高负载命令

（4）远程配置防火墙时不要把自己提出服务器

（5）指定合理的密码规范并定期更新

（6）合理分配权限

（7）定期备份重要数据和日志

# 第四章 Linux常用命令

## 4.1 文件处理命令

### 4.1.1 命令格式

#### 命令格式

命令格式：命令 [-选项] [参数]

例：ls -la /etc

说明：1）个别命令使用不遵循此格式

2）当有多个选项时，可以写在一起

3）简化选项与完整选项

-a等于—all

### 4.1.2 目录处理命令

#### 目录处理命令：ls

命令名称：ls

命令英文原意：list

命令所在路径：/bin/ls

执行权限：所有用户

功能描述：显示目录文件

语法：ls 选项[-ald] [文件或目录]

-a 显示所有文件，包括隐藏文件

-l 详细信息显示

-d 查看目录属性

-rw-r—r--

-文件类型（-文件 d目录 l软连接文件）

rw-r—r--

u g o

u 所有者 g所属组 o其他人

r读 w写 x执行

#### 目录处理命令：mkdir

命令名称：mkdir

命令英文原意：make directories

命令所在路径：/bin/mkdir

执行权限：所有用户

功能描述：创建新目录

语法：mkdir -p [目录名]

-p递归创建

范例：$ mkdir -p /tmp/Japan/boduo

$ mkdir /tmp/Japan/longze /tmp/Japan/cangjing

#### 目录处理命令：cd

命令名称：cd

命令英文原意：change directory

命令所在路径：shell内置命令

执行权限：所有用户

功能描述：切换目录

语法：cd [目录名]

范例：$ cd /tmp/Japan/boduo

$ cd ..

#### 目录处理命令：pwd

命令名称：pwd

命令英文原意：print working directory

命令所在路径：/bin/pwd

执行权限：所有用户

功能描述：显示当前目录

语法：pwd

范例：$ pwd

/tmp/Japan

#### 目录处理命令：rmdir

命令名称：rmdir

命令英文原意：remove empty directories

命令所在路径：/bin/rmdir

执行权限：所有用户

功能描述：删除空目录

语法：rmdir [目录名]

范例：$ rmdir /tmp/Japan/boduo

#### 目录处理命令：cp

命令名称：cp

命令英文原意：copy

命令所在路径：/bin/cp

执行权限：所有用户

功能描述：复制文件或目录

语法：cp -rp [原文件或目录] [目标目录]

-r 复制目录

-p 保留文件属性

范例：$ cp –r /tmp/Japan/cangjing /root

将目录/tmp/Japan/cangjing复制到目录/root下

$ cp –rp /tmp/Japan/boduo /tmp/Japan/longze /root

将/tmp/Japan目录下的boduo和longze目录复制到/root下，保持目录属性

#### 目录处理命令：mv

命令名称：mv

命令英文原意：move

命令所在路径：/bin/mv

执行权限：所有用户

功能描述：剪切文件、改名

语法：mv [原文件或目录] [目标目录]

#### 目录处理命令：rm

命令名称：rm

命令英文原意：remove

命令所在路径：/bin/rm

执行权限：所有用户

功能描述：删除文件

语法：rm -rf [文件或目录]

-r 删除目录

-f 强制执行

范例：$ rm /tmp/yum.log

删除文件/tmp/yum.log

$ rm -rf /tmp/Japan/longze

删除目录/tmp/Japan/longze

### 4.1.3 文件处理命令

#### 文件处理命令：touch

命令名称：touch

命令所在路径：/bin/touch

执行权限：所有用户

功能描述：创建空文件

语法：touch [文件名]

范例：$ touch Japanlovestory.list

#### 文件处理命令：cat

命令名称：cat

命令所在路径：/bin/cat

执行权限：所有用户

功能描述：显示文件内容

语法：cat [文件名]

-n 显示行号

范例：$ cat /etc/issue

$ cat -n /etc/services

#### 文件处理命令：tac

命令名称：tac

命令所在路径：/bin/tac

执行权限：所有用户

功能描述：显示文件内容（反向列示）

语法：tac [文件名]

范例：$ tac /etc/issue

#### 文件处理命令：more

命令名称：more

命令所在路径：/bin/more

执行权限：所有用户

功能描述：分页显示文件内容

语法：more [文件名]

（空格）或f 翻页

（Enter） 换行

b 回退

Q或q 退出

范例：$ more /etc/services

#### 文件处理命令：less

命令名称：less

命令所在路径：/usr/bin/less

执行权限：所有用户

功能描述：分页显示文件内容

语法：less [文件名]

范例：$ less /etc/services

#### 文件处理命令：head

命令名称：head

命令所在路径：/usr/bin/head

执行权限：所有用户

功能描述：显示文件内容的前几行

语法：head [文件名]

-n 指定行数

范例：$ head -n 20 /etc/services

#### 文件处理命令：tail

命令名称：tail

命令所在路径：/usr/bin/tail

执行权限：所有用户

功能描述：显示文件内容的后几行

语法：tail [文件名]

-n 指定行数

-f 动态显示文件末尾内容

范例：$ tail -n 18 /etc/services

### 4.1.4 链接命令

#### 链接处理命令：ln

命令名称：ln

命令英文原意：link

命令所在路径：/bin/ln

执行权限：所有用户

功能描述：生成链接文件

语法：ln -s [原文件] [目标文件]

-s 创建软链接

范例：$ ln -s /etc/issue /tmp/issue.soft

创建文件/etc/issue的软链接/tmp/issue.soft

$ ln -s /etc/issue /tmp/issue.hard

创建文件/etc/issue的硬链接/tmp/issue.hard

软链接特征：类似Windows快捷方式

1. lrwxrwxrwx l软链接

软链接文件权限都未rwxrwxrwx

2. 文件大小-只是符号链接

3. /tmp/issue.soft->/etc/issue

箭头指向原文件

硬链接特征：

1. 拷贝cp -p + 同步更新

echo "this is a test" >> /etc/motd

2. 可以通过i节点识别

3. 不能跨分区

4. 不能针对目录使用

## 4.2 权限管理命令

#### 权限管理命令：chmod

命令名称：chmod

命令英文原意：change the permissions mode of file

命令所在路径：/bin/chmod

执行权限：所有用户

功能描述：改变文件或目录权限

语法：chmod [{ugoa}{+-=}{rwx}] [文件或目录]

[mode=421] [文件或目录]

-R 递归修改

范例：$ chmod g+w testfile

赋予文件testfile所属组写权限

$ chmod -R 777 testdir

修改目录testdir及其目录下文件为所有用户具有全部权限

权限的数字表示

r----4

w----2

x----1

rwxrw-r--

7 6 4

文件目录权限总结

|  |  |  |  |
| --- | --- | --- | --- |
| **代表字符** | **权限** | **对文件的含义** | **对目录的含义** |
| r | 读权限 | 可以查看文件内容 | 可以列出目录中的内容 |
| w | 写权限 | 可以修改文件内容 | 可以在目录中创建、删除文件 |
| x | 执行权限 | 可以执行文件 | 可以进入目录 |

#### 权限管理命令：chown

命令名称：chown

命令英文原意：change file owernership

命令所在路径：/bin/chown

执行权限：所有用户

功能描述：改变文件或目录的所有者

语法：chown [用户] [文件或目录]

范例：$ chown xuzhenkang lixiaodan

改变文件lixiaodan的所有者为xuzhenkang

#### 权限管理命令：chgrp

命令名称：chgrp

命令英文原意：change file group ownership

命令所在路径：/bin/chgrp

执行权限：所有用户

功能描述：改变文件或目录的所属组

语法：chgrp [用户组] [文件或目录]

范例：$ chgrp xujia lixiaodan

改变文件lixiaodan的所属组为xujia

#### 权限管理命令：umask

命令名称：umask

命令英文原意：the user file-creation mask

命令所在路径：Shell内置命令

执行权限：所有用户

功能描述：显示、设置文件的缺省权限

语法：umask [-S]

-S 以rwx形式显示新建文件缺省权限

范例：$ umask -S

0022

0特殊权限

做异或运算

022 --- -w -w-

777 rwx rwx rwx

022 --- -w- -w-

----------------

755 rwx r-x r-x 目录

rw- r-- r-- 文件

缺省权限不建议更改

## 4.3 文件搜索命令

#### 文件搜索命令：find

命令名称：find

命令所在路径：/bin/find

执行权限：所有用户

功能描述：文件搜索

语法：find [搜索范围] [匹配条件]

-name 区分大小写按文件名搜索

-iname 不区分大小写按文件名搜索

-size 按尺寸搜索（+大于 -小于）

-user 根据用户名查找

-group 根据所属组查找

-cmin 按指定时间查找被修改过的文件和目录

-amin访问时间 (access)

-cmin文件属性(change)

-mmin文件内容(modify)

-a 两个条件同时满足

-o 两个条件满足一个即可

-exec/-ok 命令 {} \; 对搜索结果执行操作（ok代表含有询问）

-type 根据文件类型查找

f 文件 d目录 l软链接文件

-inum 根据i节点查找

范例：$ find /etc/ -name init

在目录/etc/中查找文件init

$ find / -size +204800

在根目录下查找大于100MB的文件

+n 大于n -n 小于n n 等于n

$ find /home -user xuzhenkang

在根下查找所有者为xuzhenkang的文件

$ find /etc/ -cmin -5

在/etc/下查找5分钟内被修改过属性的文件和目录

$ find /etc/ -size +163840 -a -size -204800

在目录/etc/下查找大于80MB小于100MB的文件

$ find /etc/ -name inittab -exec ls -l {} \;

在目录/etc/下查找inittab文件并显示其详细信息

#### 文件搜索命令：locate

命令名称：locate

命令所在路径：/usr/bin/locate

执行权限：所有用户

功能描述：在文件资料库中查找文件

语法：locate [文件名]

范例：$ locate inittab

$ updatedb

更新locate的索引数据库，不含/tmp中的内容

#### 文件搜索命令：which

命令名称：which

命令所在路径：/usr/bin/which

执行权限：所有用户

功能描述：搜索命令所在目录及别名信息

语法：which 命令

范例：$ which ls

#### 文件搜索命令：whereis

命令名称：whereis

命令所在路径：/usr/bin/whereis

执行权限：所有用户

功能描述：搜索命令所在目录及帮助文档路径

语法：whereis 命令

范例：$ whereis ls

#### 文件搜索命令：grep

命令名称：grep

命令所在路径：/bin/grep

执行权限：所有用户

功能描述：在文件中搜索字符串匹配的行并输出

语法：grep -iv [指定字符串] [文件]

-i 不区分大小写

-v 排除指定字符串

范例：# grep mysql /root/install.log

# grep -i multiuser /etc/inittab

不区分大小写在/etc/inittab中查找multiuser

# grep -v ^# /etc/inittab

在/etc/inittab中过滤掉以#开头的行，即查看非注释行。

## 4.4 帮助命令

#### 帮助命令：man

命令名称：man

命令英文原意：manual

命令所在路径：/usr/bin/man

执行权限：所有用户

功能描述：获得帮助信息

语法：man [命令或配置文件]

1 命令的帮助

5 配置文件的帮助

范例：$ man ls

查看ls命令的帮助信息

$ man services

查看配置文件services的帮助信息

$ man 5 passwd

因为既有passwd配置文件又有passwd命令，所以这里要使用数字指定。本条命令的意思是查看配置文件passwd的帮助信息

#### 帮助命令：help

命令名称：help

命令所在路径：Shell内置命令

执行权限：所有用户

功能描述：获得Shell内置命令的帮助信息(用whereis或者which无法找到路径的命令)

语法：help 命令

范例：$ help umask

查看umask命令的帮助信息

$ man services

查看配置文件services的帮助信息

$ man 5 passwd

因为既有passwd配置文件又有passwd命令，所以这里要使用数字指定。本条命令的意思是查看配置文件passwd的帮助信息

#### 其他帮助命令：info

说明：与man的功能大同小异，使用方法非常相似

#### 其他帮助命令：whatis

说明：得到命令的简略信息

范例：$ whatis ls

查看ls命令的简略信息

#### 其他帮助命令：apropos

说明：得到配置文件的简略信息

范例：$ apropos services

得到配置文件services的简略信息

#### 其他帮助命令：--help

说明：简单的查看前面的命令的用法

范例：$ ls --help

得到ls命令的用法

## 4.5 用户管理命令

#### 用户管理命令：useradd

命令名称：useradd

命令所在路径：/usr/sbin/useradd

执行权限：root

功能描述：添加新用户

语法：useradd 用户名

范例：# useradd lixiaodan

#### 用户管理命令：passwd

命令名称：passwd

命令所在路径：/usr/bin/passwd

执行权限：所有用户

功能描述：设置用户密码

语法：passwd 用户名

范例：$ passwd lixiaodan

#### 用户管理命令：who

命令名称：who

命令所在路径：/usr/bin/who

执行权限：所有用户

功能描述：查看登录用户信息（tty代表本地终端）

语法：who 用户名

范例：$ who

#### 用户管理命令：w

命令名称：w

命令所在路径：/usr/bin/w

执行权限：所有用户

功能描述：查看登录用户详细信息

语法：w

范例：$ w

## 4.6 压缩解压命令

#### 压缩命令：gzip

命令名称：gzip

命令英文原意：GNU zip

命令所在路径：/bin/gzip

执行权限：所有用户

功能描述：压缩文件

压缩后文件格式：.gz

语法：gzip [文件]

#### 解压命令：gunzip

命令名称：gunzip

命令英文原意：GNU unzip

命令所在路径：/bin/gunzip

执行权限：所有用户

功能描述：解压缩.gz的压缩文件

语法：gzip [文件]

范例：$ gunzip boduo.gz

#### 压缩命令：tar

命令名称：tar

命令所在路径：/bin/tar

执行权限：所有用户

功能描述：打包目录

压缩后文件格式：.tar.gz

语法：tar 选项[-zcf] [压缩后文件名] [目录]

-c 打包

-v 显示详细信息

-f 指定文件名

-z 打包同时压缩

-x 解包

范例：$ tar -zcf Japan.tar.gz Japan

将目录Japan打包并压缩为.tar.gz文件

$ tar -zxvf Japan.tar.gz

解压缩Japan.tar.gz文件

#### 压缩命令：zip

命令名称：zip

命令所在路径：/usr/bin/zip

执行权限：所有用户

功能描述：压缩文件或目录（该格式可在Windows和Linux下直接使用）

压缩后文件格式：.zip

语法：zip 选项[-r] [压缩后文件名] [文件或目录]

-r 压缩目录

范例：$ zip boduo.zip boduo

压缩文件

$ zip -r Japan.zip Japan

压缩目录

#### 解压命令：unzip

命令名称：unzip

命令所在路径：/usr/bin/unzip

执行权限：所有用户

功能描述：解压缩.zip的压缩文件

语法：unzip [压缩文件]

范例：$ unzip test.zip

#### 压缩命令：bzip2

命令名称：bzip2

命令所在路径：/usr/bin/bzip2

执行权限：所有用户

功能描述：压缩文件

压缩后文件格式：.bz2

语法：bzip2 选项[-k] [文件]

-k 产生压缩文件后保留原文件

范例：$ bzip2 -k boduo

$ tar -cjf Japan.tar.bz2 Japan

#### 解压命令：bunzip2

命令名称：bunzip2

命令所在路径：/usr/bin/bunzip2

执行权限：所有用户

功能描述：解压缩

语法：bunzip2 选项[-k] [压缩文件]

-k 解压缩后保留原文件

范例：$ bunzip2 -k boduo.bz2

$ tar -xjf Japan.tar.bz2

## 4.7 网络命令

#### 网络命令：write

命令名称：write

命令所在路径：/usr/bin/write

执行权限：所有用户

功能描述：给用户发信息，以Ctrl+D保存结束

语法：write <用户名>

范例：$ write lixiaodan

#### 网络命令：wall

命令名称：wall

命令所在路径：/usr/bin/wall

执行权限：所有用户

功能描述：发送广播消息

语法：wall [message]

范例：$ wall LiXiaodan is XuZhenkang's wife!

#### 网络命令：ping

命令名称：ping

命令所在路径：/bin/ping

执行权限：所有用户

功能描述：测试网络连通性

语法：ping 选项 IP地址

-c 指定发送次数

范例：$ ping 192.168.121.1

#### 网络命令：ifconfig

命令名称：ifconfig

命令英文原意：interface configure

命令所在路径：/sbin/ifconfig

执行权限：root

功能描述：查看和设置网卡信息

语法：ifconfig 网卡名称 IP地址

范例：$ ifconfig eth0 192.168.121.2

#### 网络命令：mail

命令名称：mail

命令所在路径：/bin/mail

执行权限：所有用户

功能描述：查看发送电子邮件

语法：mail [用户名]

范例：$ mail root

#### 网络命令：last

命令名称：last

命令所在路径：/usr/bin/last

执行权限：所有用户

功能描述：列出目前与过去登入系统的用户信息

语法：last

范例：# last

#### 网络命令：lastlog

命令名称：lastlog

命令所在路径：/usr/bin/lastlog

执行权限：所有用户

功能描述：检查某特定用户上次登录时间

语法：lastlog

范例：# lastlog

# lastlog -u 502

#### 网络命令：traceroute

命令名称：traceroute

命令所在路径：/bin/traceroute

执行权限：所有用户

功能描述：显示数据包到主机间的路径

语法：traceroute

范例：$ traceroute www.ikang.tk

#### 网络命令：netstat

命令名称：netstat

命令所在路径：/bin/netstat

执行权限：所有用户

功能描述：显示网络相关信息

语法：netstat [选项]

-t TCP协议

-u UDP协议

-l 监听

-r 路由

-n 显示IP地址和端口号

范例：# netstat -tlun 查看本机监听的端口

# netstat -an 查看本机所有的网络连接

# netstat -rn 查看本机路由表

#### 网络命令：setup

命令名称：setup

命令所在路径：/usr/bin/setup

执行权限：root

功能描述：配置网络（redhat系列特有命令）

语法：setup

范例：# setup

## 4.8 关机重启命令

#### 关机重启命令：shutdown

命令名称：shutdown

命令所在路径：/sbin/shutdown

执行权限：root

功能描述：关机重启等操作

语法：shutdown [选项] 时间

-c 取消前一个关机命令

-h 关机

-r 重启

范例：# shutdown -h now

#### 其他关机命令：halt

范例：# halt

#### 其他关机命令：poweroff

说明：该命令是直接断电（注意，这种关机不许使用！！非常不安全）

范例：# poweroff

#### 其他关机命令：init 0

范例：# init 0

#### 其他重启命令：init 6

范例：# init 6

#### 其他重启命令：reboot

范例：# reboot

#### 系统运行级别

Default runlevel. The runlevels used 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)

NFS服务是linux和unix之间文件共享的服务。

X11是一种图形环境的意思（一般大写的X都是代表图形界面的意思）

#### 修改系统运行级别

[root@localhost ~]# vim /etc/inittab

id:3:initdefault:

#### 查询系统运行级别

[root@localhost ~]# runlevel

N 3

#### 查看该Linux服务器运行时间

[root@localhost ~]# uptime

19:28:22 up 17:27, 1 user, load average: 0.01, 0.00, 0.00

#### 将某用户提出Linux系统

[root@localhost ~]# pkill -U 用户名

#### 退出登录

[root@localhost ~]# logout

注意：做完操作之后，一定要logout！

# 第五章 文本编辑器Vim

## 5.1 Vim常用操作

#### Vim简介

Vim是一个功能强大的全屏幕文本编辑器，是Linux/UNIX上最常用的文本编辑器，它的作用是建立、编辑、显示文本文件。

Vim没有菜单，只有命令。

#### Vim工作模式



#### 插入命令

|  |  |
| --- | --- |
| **命令** | **作用** |
| a | 在光标所在字符后插入 |
| A | 在光标所在行尾插入 |
| i | 在光标所在字符前插入 |
| I | 在光标所在行首插入 |
| o | 在光标下插入新行 |
| O | 在光标上插入新行 |

#### 定位命令

|  |  |
| --- | --- |
| **命令** | **作用** |
| :set nu | 设置行号 |
| :set nonu | 取消行号 |
| gg | 到第一行 |
| G | 到最后一行 |
| nG | 到第n行 |
| :n | 到第n行 |
| $ | 移至行尾 |
| 0 | 移至行首 |

#### 定位命令

|  |  |
| --- | --- |
| **命令** | **作用** |
| x | 删除光标所在处字符 |
| nx | 删除光标所在处后n个字符 |
| dd | 删除光标所在行，ndd删除n行 |
| dG | 删除光标所在行到文件末尾内容 |
| D | 删除光标所在处到行尾内容 |
| :n1,n2d | 删除指定范围的行 |

#### 复制和剪切命令

|  |  |
| --- | --- |
| **命令** | **作用** |
| yy | 复制当前行 |
| nyy | 复制当前行以下n行 |
| dd | 剪切当前行 |
| ndd | 剪切当前行以下n行 |
| p、P | 粘贴在当前光标所在行下或行上 |

#### 替换和取消命令

|  |  |
| --- | --- |
| **命令** | **作用** |
| r | 取代光标所在处字符 |
| R | 从光标所在处开始替换字符，按ESC结束 |
| u | 取消上一步操作 |

#### 搜索和搜索替换命令

|  |  |
| --- | --- |
| **命令** | **作用** |
| /string | 搜索指定字符串  搜索是忽略大小写:set ic |
| n | 搜索指定字符串的下一个出现位置 |
| :%s/old/new/g | 全文替换指定字符串 |
| :n1,n2s/old/new/g | 在一定范围内替换指定字符串 |

#### 保存和退出命令

|  |  |
| --- | --- |
| **命令** | **作用** |
| :w | 保存修改 |
| :w new\_filename | 另存为指定文件 |
| :wq | 保存修改并退出 |
| ZZ | 快捷键，保存修改并退出 |
| :q! | 不保存修改退出 |
| :wq! | 保存修改并退出(文件所有者及root可使用) |

## 5.2 Vim使用技巧

#### 导入命令执行结果

:r !命令

#### 定义快捷键

:map 快捷键 触发命令

范例：:map ^P I#<ESC>

:map ^B 0x

#### 连续行注释

:n1,n2s/^/#/g

:n1,n2s/^#//g

:n1,n2s/^/\/\//g

#### 替换

:ab mymail xuzhenkang@hotmail.com

#### Vim的配置文件

vim的配置文件是在用户的家目录下的.vimrc。它可能存在也可能不存在。可以把一些配置写到该文件中去。这样可以长久保存。

下面给出一个功能相对强大的配置文件

文件：.vimrc

"这是Vim文本编辑器的配置文件。

"This is a configuration file of Vim.

"设置显示行号

"Set the line number.

set nu

"设置我的邮箱:xuzhenkang@hotmail.com

ab mymail xuzhenkang@hotmail.com

"去掉讨厌的有关vi一致性模式，避免以前版本的一些bug和局限

"set nocompatible

"到处使用鼠标

set mouse=a

"检测文件类型

"filetype=100

"记录历史行数

"set history=1000

"背景使用黑色

set background=dark

"语法高亮

syntax on

"下面两行在进行编写代码时，在格式对齐上很有用;

"第一行，vim使用自动对齐,也就是把当前行的对其格式应用到下一行

"第二行，依据上面的对齐格式，只能地选择对齐方式，对于类似C语言编写上很有用

set autoindent

set smartindent

"支持C/C++的缩进

set cin

"方便使用taglist

nmap <F8> : Tlist <CR>

"第一行设置tab键为4个空格，第二行设置当行之间交错时使用4个空格

set tabstop=2

set shiftwidth=2

"设置匹配模式，类似当输入一个左括号时会匹配相应的那个右括号

set showmatch

"去除vim的GUI版本中的toolbar

"set guioptions=-T

"当vim进行编辑时，如果命令错误，会发出一个响声，该设置去掉响声

"set vb t\_vb

"在编辑过程中，在右下角显示光标位置的状态行

set ruler

"寻找匹配是高亮度显示的

set hls

"<F2>关闭高亮

map <F2> : nohlsearch <CR>

"查询时非常方便，如果查找book单词，当输入到/b时，会自动找到第一个b开头的单词，当输入到/bo时，会自动找到第一个bo开头的单词，以此类推，进行查找时，使用此设置会快速找到答案，当你找到要匹配的单词时，别忘记回车。

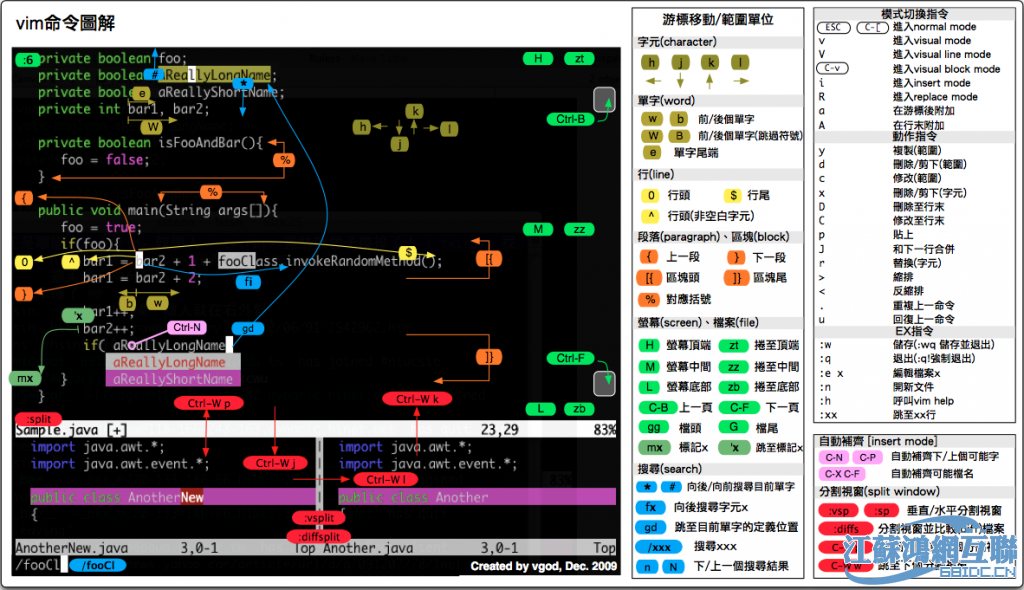
"set incsearch

"修改一个文件后，自动进行备份，备份的文件名为原文件名加"~"后缀

"set backup

## 5.3 Vim命令图解







# 第六章 Linux软件安装

## 6.1 软件包管理简介

#### 软件包分类

源码包：源代码安装包、脚本安装包

二进制包：（RPM包--系统默认包）

#### 源码包

优点：

（1）开源，如果有足够的能力，可以修改源代码

（2）可以自由选择所需的功能

（3）软件是编译安装，所以更加适合自己的系统，更加稳定也效率更高

（4）卸载方便

缺点：

（1）安装过程步骤较多，尤其安装较大的软件集合时（如LAMP环境搭建），容易出现拼写错误

（2）编译过程时间较长，安装比二进制安装时间长

（3）因为是编译安装，安装过程中一旦报错新手很难解决

#### RPM包（二进制包，系统默认包）

优点：

（1）包管理系统简单，只通过几个命令就可以实现包的安装、升级、查询和卸载

（2）安装速度比源码包安装快的多

缺点：

（1）二进制包是经过编译的安装包，不再可以看到源代码

（2）功能选择不如源码包灵活

（3）依赖性是一个很闹心的问题

## 6.2 RPM包管理-rpm命令管理

### 6.2.1 包命名与依赖性

#### RPM包命名原则

httpd-2.2.15-15.el6.centos.1.i686.rpm

httpd 软件包名

2.2.15 软件版本

15 软件发布的次数

el6.centos 适合的Linux平台

i686 适合的硬件平台

rpm rpm包扩展名

#### RPM包依赖性

（1）树形依赖： a→b→c

（2）环形依赖： a→b→c→a

（3）模块依赖： 模块依赖查询网站： [www.rpmfind.net](file:///C:\Users\Kang\Desktop\www.rpmfind.net)

noarch代表任何linux平台都可以安装

.so.数字结尾 代表库依赖包

### 6.2.2 安装升级与卸载

#### 包名与包全名

包全名：操作的包是没有安装的软件包时，使用包全名。而且要注意路径

包名：操作已经安装的软件包时，使用包名。是搜索/var/lib/rpm/中的数据库

#### RPM安装

rpm -ivh 包全名

选项：

-i（install） 安装

-v（verbose） 显示详细信息

-h（hash） 显示进度

--nodeps 不检测依赖性

#### RPM包升级

rpm -Uvh 包全名

选项：

-U（upgrade） 升级

注意！！！！

如果远程连接Linux服务器，不能使用这样的命令：

yum -y update

这条命令的意思是升级服务器上面所有的软件包，包括Linux内核，升级Linux内核需要在本地手工进行一些相应的配置才能使用，如果远程连接的Linux服务器，就会导致这个Linux服务器无论如何都无法开机，永远无法连接。这条命令会导致服务器崩溃！！！！！！！！！！！！

#### 卸载

rpm -e 包名

选项：

-e（erase） 卸载

--nodeps 不检查依赖性

### 6.2.3 RPM包查询

#### 查询是否安装

[root@localhost ~]# rpm -q 包名

#查询包是否安装

选项：

-q 查询（query）

[root@localhost ~]# rpm –qa

#查询所有已经安装的RPM包

选项：

-a 所有（all）

#### 查询软件包详细信息

[root@localhost ~]# rpm –qi 包名

选项：

-i 查询软件信息（information）

-p 查询未安装包信息（package）

#### 查询包中文件安装位置

[root@localhost ~]# rpm –ql 包名

选项：

-l 列表（list）

-p 查询未安装包信息（package）

#### 查询系统文件属于哪个RPM包

[root@localhost ~]# rpm –qf 系统文件名

选项：

-f 查询系统文件属于哪个软件包（file）

#### 查询软件包的依赖性

[root@localhost ~]# rpm –qR 包名

选项：

-R 查询软件包的依赖性（requires）

-p 查询未安装包信息（package）

### 6.2.4 校验与文件提取

#### RPM包校验

[root@localhost ~]# rpm –V 已安装的包名

选项：

-V 校验指定RPM包中的文件（verify）

验证内容中的8个信息的具体内容如下：

S 文件大小是否改变

M 文件的类型或文件的权限（rwx）是否被改变

5 文件MD5校验和是否改变（可以看成文件内容是否改变）

D 设备的中，从代码是否改变

L 文件路径是否改变

U 文件的属主（所有者）是否改变

G 文件的属组是否改变

T 文件的修改时间是否改变

文件类型

c 配置文件（config file）

d 普通文档（documentation）

g “鬼”文件（ghost file），很少见，就是该文件不应该被这个RPM包包含

l 授权文件（license file）

r 描述文件（read me）

#### RPM包中文件提取

[root@localhost ~]# rpm2cpio 包全名 | cpio -idv .文件绝对路径

rpm2cpio

#将rpm包转换为cpio格式的命令

cpio

#是一个标准工具，它用于创建软件档案文件和从档案文件中提取文件

[root@localhost ~]# cpio 选项 < [文件|设备]

选项：

-i：copy-in模式，还原

-d：还原时自动新建目录

-v：显示还原过程

[root@localhost ~]# rpm -qf /bin/ls

#查询ls命令属于哪个软件包

[root@localhost ~]# mv /bin/ls /tmp/

#造成ls命令误删除假象

[root@localhost ~]# rpm2cpio /mnt/cdrom/Packages/coreutils-8.4-19.el6.i686.rpm | cpio -idv ./bin/ls

#提取RPM包中ls命令到当前目录的/bin/ls下

[root@localhost ~]# cp /root/bin/ls /bin/

#把ls命令复制会/bin/目录，修复文件丢失

## 6.3 RPM包管理-yum在线管理

### 6.3.1 IP地址配置和网络yum源

#### IP地址配置

[root@localhost ~]# setup

#使用setup工具

[root@localhost ~]# vi /etc/sysconfig/network-scripts/ifcfg-eth0

把ONBOOT="no" 改为

ONBOOT="yes"

#启动网卡

[root@localhost ~]# service network restart

#重启网络服务

#### 网络yum源

[root@localhost yum.repos.d]# vim /etc/yum.repos.d/CentOS-Base.repo

[base] 容器名称，一定要放在[]中

name 容器说明，可以自己随便写

mirrorlist 镜像站点，这个可以注释掉

baseurl 我们的yum源服务器的地址。默认是CentOS官方的yum源服务 器，是可以使用的，如果你觉得慢可以改成你喜欢的yum源地 址

enabled 此容器是否生效，如果不写或写成enable=1都是生效，写成 enable=0就是不生效

gpgcheck 如果是1是指RPM的数字证书生效，如果是0则不生效

gpgkey 数字证书的公钥文件保存位置。不用修改

### 6.3.2 yum命令

#### 常用yum命令

（1）查询

[root@localhost yum.repos.d]# yum list

#查询所有可用软件包列表

[root@localhost yum.repos.d]# yum search 关键字

#搜索服务器上所有和关键字相关的包

（2）安装

[root@localhost yum.repos.d]# yum –y install 包名

选项：

install 安装

-y 自动回答yes

（3）升级

[root@localhost yum.repos.d]# yum -y update 包名

选项：

update 升级

-y 自动回答yes

（4）卸载

[root@localhost yum.repos.d]# yum -y remove 包名

选项：

remove 卸载

-y 自动回答yes

使用yum卸载会出现很多问题，它会把一些相应的依赖的包也相应的卸载，这样这些依赖包还有可能被其他软件包依赖，包括系统软件，这样有可能导致系统无法启动！！！！！所以，使用yum卸载是一件相当不安全的事情！

#### YUM软件组管理命令

[root@localhost ~]# yum grouplist

#列出所有可用的软件组列表

[root@localhost ~]# yum groupinstall 软件组名

#安装指定软件组，组名可以由grouplist查询出来

[root@localhost ~]# yum groupremove 软件组名

#卸载指定软件组

### 6.3.3 光盘yum源搭建

#### 光盘yum源搭建步骤

（1）挂载光盘

[root@localhost ~]# mount /dev/cdrom /mnt/cdrom/

（2）让网络yum源文件失效

[root@localhost ~]# cd /etc/yum.repos.d/

[root@localhost yum.repos.d]# mv CentOS-Base.repo CentOS-Base.repo.bak

[root@localhost yum.repos.d]# mv CentOS-Debuginfo.repo CentOS-Debuginfo.repo.bak

[root@localhost yum.repos.d]# mv CentOS-Vault.repo CentOS-Vault.repo.bak

（3）修改光盘yum源文件

[root@localhost yum.repos.d]# vim CentOS-Media.repo

[c6-media]

name=CentOS-$releasever - Media

baseurl=file:///mnt/cdrom

#地址为你自己的光盘挂载地址

# file:///media/cdrom/

# file:///media/cdrecorder/

#注释这两个不存在的地址

gpgcheck=1

enabled=1

#把enabled=0改为enabled=1，让这个yum源配置文件生效

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

注意：在linux配置文件中的注释一定要顶格写，有些地方不加空格千万不能加空格。

## 6.4源码包管理

### 6.4.1 源码包和RPM包的区别

#### 区别

（1）安装之前的区别：概念上的区别

（2）安装之后的区别：安装位置不同

#### RPM包安装位置

是安装在默认位置中

|  |  |
| --- | --- |
| RPM包默认安装路径 | |
| /etc/ | 配置文件安装目录 |
| /usr/bin/ | 可执行的命令安装目录 |
| /usr/lib/ | 程序所使用的函数库保存位置 |
| /usr/share/doc/ | 基本的软件使用手册保存位置 |
| /usr/share/man/ | 帮助文件保存位置 |

#### 源码包安装位置

安装在指定位置当中，一般是 /usr/local/软件名/

#### 安装位置不同带来的影响

RPM包安装的服务可以使用系统服务管理命令（service）来管理，例如RPM包安装的apache的启动方法是：

/etc/rc.d/init.d/httpd start

service httpd start

而源码包安装的服务则不能被服务管理命令管理，因为没有安装到默认路径中。所以只能用绝对路径进行服务的管理，如：

/usr/local/apache2/bin/apachectl start

### 6.4.2 源码包安装过程

#### 安装准备

（1）安装C语言编译器

（2）下载源码包 <http://mirror.bit.edu.cn/apache/httpd/>

#### 安装注意事项

（1）源代码保存位置：/usr/local/src/

（2）软件安装位置：/usr/local/

（3）如何确定安装过程报错：

1）安装过程停止

2）并出现error、warning或no的提示

#### 源码包安装过程

（1）下载源码包

（2）解压缩下载的源码包

（3）进入解压缩目录（非常重要）

（4）./configure 软件配置与检查

（5）定义需要的功能选项。

（6）检测系统环境是否符合安装要求。

（7）把定义好的功能选项和检测系统环境的信息都写入Makefile文件，用于后续的编辑。

（8）make 编译

make clean清除编译

（9）make install 编译安装

#### 源码包的卸载

不需要卸载命令，直接删除安装目录即可。不会遗留任何垃圾文件。

## 6.5脚本安装包

#### 脚本安装包

脚本安装包并不是独立的软件包类型，常见安装的是源码包。

是人为把安装过程写成了自动安装的脚本，只要执行脚本，定义简单的参数，就可以完成安装。

非常类似于Windows下软件的安装方式。

#### Webmin的作用

Webmin 是一个基于 Web 的 Linux 系统管理界面。您就可以通过图形化的方式设置用户帐号、Apache、DNS、文件共享等服务。

#### Webmin安装过程

（1）下载软件：<http://sourceforge.net/projects/webadmin/files/webmin/>

（2）解压缩，并进入加压缩目录

（3）执行安装脚本

# 第七章 用户和用户组管理

## 7.1 用户配置文件

### 7.1.1 用户信息文件/etc/passwd

#### 用户管理简介

越是对服务器安全性要求高的服务器，越需要建立合理的用户权限等级制度和服务器操作规范。在Linux中主要是通过用户配置文件来查看和修改用户信息

#### /etc/passwd

第1字段：用户名称

第2字段：密码标志

第3字段：UID（用户ID）

0： 超级用户

1-499： 系统用户（伪用户）

500-65535： 普通用户

第4字段：GID（用户初始组ID）

第5字段：用户说明

第6字段：家目录

普通用户：/home/用户名/

超级用户：/root/

第7字段：登录之后的Shell

#### 初始组和附加组

**初始组**：就是指用户一登录就立刻拥有这个用户组的相关权限，每个用户的初始组只能有一个，一般就是和这个用户的用户名相同的组名作为这个用户的初始组。

**附加组**：指用户可以加入多个其他的用户组，并拥有这些组的权限，附加组可以有多个。

#### Shell是什么

Shell就是Linux的命令解释器。

在/etc/passwd当中，除了标准Shell是/bin/bash之外，还可以写如/sbin/nologin，/usr/bin/passwd等。

### 7.1.2 影子文件/etc/shadow

#### 影子文件/etc/shadow

第1字段：用户名

第2字段：加密密码

加密算法升级为SHA512散列加密算法

如果密码位是“!!”或“\*”代表没有密码，不能登录

第3字段：密码最后一次修改日期

使用1970年1月1日作为标准时间，每过一天时间戳加1

第4字段：两次密码的修改间隔时间（和第3字段相比）

第5字段：密码有效期（和第3字段相比）

第6字段：密码修改到期前的警告天数（和第5字段相比）

第7字段：密码过期后的宽限天数（和第5字段相比）

0：代表密码过期后立即失效

-1：则代表密码永远不会失效。

第8字段：账号失效时间

要用时间戳表示

第9字段：保留

#### 时间戳换算

（1）把时间戳换算为日期

date -d "1970-01-01 16066 days"

（2）把日期换算为时间戳

echo $(($(date --date="2014/01/06" +%s)/86400+1))

### 7.1.3 组信息文件/etc/group和组密码文件/etc/gshadow

#### 组信息文件/etc/group

第一字段：组名

第二字段：组密码标志

第三字段：GID

第四字段：组中附加用户

#### 组密码文件/etc/gshadow

第一字段：组名

第二字段：组密码

第三字段：组管理员用户名

第四字段：组中附加用户

## 7.2 用户管理相关文件

#### 用户的家目录

普通用户：/home/用户名/，所有者和所属组都是此用户，权限是700

超级用户：/root/，所有者和所属组都是root用户，权限是550

#### 用户的邮箱

/var/spool/mail/用户名/

#### 用户模板目录

/etc/skel/

## 7.3 用户管理命令

### 7.3.1 用户添加命令useradd

#### useradd命令格式

[root@localhost ~]#useradd [选项] 用户名

选项：

-u UID： 手工指定用户的UID号

-d 家目录： 手工指定用户的家目录

-c 用户说明： 手工指定用户的说明

-g 组名： 手工指定用户的初始组

-G 组名： 指定用户的附加组

-s shell： 手工指定用户的登录shell。默认是/bin/bash

#### 添加默认用户

[root@localhost ~]# useradd lamp

[root@localhost ~]# grep "lamp" /etc/passwd

[root@localhost ~]# grep "lamp" /etc/shadow

[root@localhost ~]# grep "lamp" /etc/group

[root@localhost ~]# grep "lamp" /etc/gshadow

[root@localhost ~]# ll -d /home/lamp/

[root@localhost ~]# ll /var/spool/mail/lamp

#### 指定选项添加用户

groupadd lamp1

useradd -u 550 -g lamp1 -G root -d /home/lamp1 -c "test user" -s /bin/bash lamp1

#### 用户默认值文件

/etc/default/useradd

GROUP=100 #用户默认组

HOME=/home #用户家目录

INACTIVE=-1 #密码过期宽限天数（7）

EXPIRE= #密码失效时间（8）

SHELL=/bin/bash #默认shell

SKEL=/etc/skel #模板目录

CREATE\_MAIL\_SPOOL=yes #是否建立邮箱

/etc/login.defs

PASS\_MAX\_DAYS 99999 #密码有效期（5）

PASS\_MIN\_DAYS 0 #密码修改间隔（4）

PASS\_MIN\_LEN 5 #密码最小5位（PAM）

PASS\_WARN\_AGE 7 #密码到期警告（6）

UID\_MIN 500 #最小和最大UID范围

GID\_MAX 60000

ENCRYPT\_METHOD SHA512 #加密模式

### 7.3.2 修改用户密码passwd

#### passwd命令格式

[root@localhost ~]#passwd [选项] 用户名

选项：

-S 查询用户密码的密码状态。仅root用户可用。

-l 暂时锁定用户。仅root用户可用

-u 解锁用户。仅root用户可用

--stdin 可以通过管道符输出的数据作为用户的密码。

#### 查看密码状态

[root@localhost ~]# passwd -S lamp

lamp PS 2013-01-06 0 99999 7 -1

#用户名密码设定时间（2013-01-06） 密码修改间隔时间（0）

#密码有效期(99999) 警告时间（7） 密码不失效（-1）

#### 锁定用户和解锁用户

[root@localhost ~]# passwd -l lamp

[root@localhost ~]# passwd -u lamp

#### 使用字符串作为用户的密码

[root@localhost ~]# echo "123" | passwd --stdin lamp

### 7.3.3 修改用户信息usermod修改用户密码状态chage

#### 修改用户信息usermod

[root@localhost ~]# usermod [选项] 用户名

选项：

-u UID： 修改用户的UID号

-c 用户说明： 修改用户的说明信息

-G 组名： 修改用户的附加组

-L： 临时锁定用户（Lock）

-U： 解锁用户锁定（Unlock）

[root@localhost ~]# usermod -c "test user" lamp

#修改用户的说明

[root@localhost ~]# usermod -G root lamp

#把lamp用户加入root组

[root@localhost ~]# usermod -L lamp

#锁定用户

[root@localhost ~]# usermod -U lamp

#解锁用户

#### 修改用户密码状态chage

[root@localhost ~]# chage [选项] 用户名

选项：

-l： 列出用户的详细密码状态

-d 日期： 修改密码最后一次更改日期（shadow3字段）

-m 天数： 两次密码修改间隔（4字段）

-M 天数： 密码有效期（5字段）

-W 天数： 密码过期前警告天数（6字段）

-I 天数： 密码过后宽限天数（7字段）

-E 日期： 账号失效时间（8字段）

[root@localhost ~]# chage -d 0 lamp

#这个命令其实是把密码修改日期归0了（shadow第3字段）#这样用户一登陆就要修改密码

### 7.3.4 删除用户userdel用户切换命令su

#### 删除用户userdel

[root@localhost ~]# userdel [-r] 用户名

选项：

-r 删除用户的同时删除用户家目录

[手工删除用户](不建议)

[root@localhost ~]# vi /etc/passwd

[root@localhost ~]# vi /etc/shadow

[root@localhost ~]# vi /etc/group

[root@localhost ~]# vi /etc/gshadow

[root@localhost ~]# rm -rf /var/spool/mail/lamp

[root@localhost ~]# rm -rf /home/lamp/

#### 查看用户ID

[root@localhost ~]# id 用户名

#### 切换用户身份su

[root@localhost ~]# su [选项] 用户名

选项：

-： 选项只使用“-”代表连带用户的环境 变量一起切换

-c 命令： 仅执行一次命令，而不切换用户身份

**切换用户的时候，一定要添加减号，这样会连带着环境变量一起切换！！很重要，一定要连着用户的环境变量一起切换，否则会出现各种奇怪的问题。**

$ su - root

[lamp@localhost ~]$ su – root

#切换成root

[lamp@localhost ~]$ su - root -c "useradd user3"

#不切换成root，但是执行useradd命令添加user1用户

## 7.4 用户组管理命令

#### 添加用户组

[root@localhost ~]# groupadd [选项] 组名

选项：

-g GID: 指定组ID

#### 修改用户组(不建议，不如删掉重新建立)

[root@localhost ~]# groupmod [选项] 组名

选项：

-g GID: 修改组ID

-n 新组名: 修改组名

[root@localhost ~]# groupmod -n testgrp group1

#把组名group1修改为testgrp

#### 删除用户组

[root@localhost ~]# groupdel 组名

**注意：万一组中有用户这个组是否能删呢？！这是要分情况的，用户和组来讲是有两种不同的身份的，一种是初始用户，另一种是附加用户。初始用户和组的关系是相互依存的，如果没有初始组那么这个初始用户就不存在，那么这种情况下（组中有初始用户存在）这个组是不可以删除的，因为一旦删了，这个用户就没有组了,这样会报错；如果组里面有的不是初始用户而是附加用户，那么不影响组的删除。**

例子：

[root@localhost ~]# groupadd tg

[root@localhost ~]# useradd -g tg lamp

[root@localhost ~]# useradd -G tg lamp1

[root@localhost ~]# groupdel tg

groupdel: cannot remove the primary group of user 'lamp'

[root@localhost ~]# userdel -r lamp

[root@localhost ~]# groupdel tg

[root@localhost ~]#

#### 把用户添加入组或从组中删除

[root@localhost ~]# gpasswd 选项 组名

选项：

-a 用户名： 把用户加入组(操作的是附加用户)

-d 用户名： 把用户从组中删除

例子：

[root@localhost ~]# gpasswd -a user1 root

Adding user user1 to group root

[root@localhost ~]# gpasswd -a user3 root

Adding user user1 to group root

[root@localhost ~]#

# 第八章 权限管理

## 8.1 ACL权限

### 8.1.1 ACL权限简介与开启

#### ACL权限简介



#### 查看分区ACL权限是否开启

[root@localhost ~]# dumpe2fs -h /dev/sda3

#dumpe2fs命令是查询指定分区详细文件系统信息的命令

选项：

-h 仅显示超级块中信息，而不显示磁盘块组的详细信息

#### 临时开启分区ACL权限

[root@localhost ~]# mount -o remount,acl /

#重新挂载根分区，并挂载加入acl权限

#### 永久开启分区ACL权限

[root@localhost ~]# vi /etc/fstab

UUID=c2ca6f57-b15c-43ea-bca0-f239083d8bd2 / ext4 defaults,acl 1 1

#加入acl

[root@localhost ~]# mount -o remount /

#重新挂载文件系统或重启动系统，使修改生效

### 8.1.2 查看与设定ACL权限

#### 查看ACL命令

[root@localhost ~]# getfacle 文件名

#查看acl权限

#### 设定ACL权限的命令

[root@localhost ~]# setfacl 选项 文件名

选项： -m 设定ACL权限

-x 删除指定的ACL权限

-b 删除所有的ACL权限

-d 设定默认ACL权限。

-k 删除默认ACL权限

-R 递归设定ACL权限。

#### 给用户设定ACL权限



[root@localhost ~]# useradd zhangsan

[root@localhost ~]# useradd lisi

[root@localhost ~]# useradd st

[root@localhost ~]# groupadd tgroup

[root@localhost ~]# mkdir /project

[root@localhost ~]# chown root:tgroup /project/

[root@localhost ~]# chmod 770 /project/

[root@localhost ~]# setfacl -m u:st:rx /project/

#给用户st赋予r-x权限，使用“u:用户名:权限”格式

#### 给用户组设定ACL权限

[root@localhost /]# groupadd tgroup2

[root@localhost /]# setfacl -m g:tgroup2:rwx project/

#为组tgroup2分配ACl权限。使用“g:组名:权限”格式

### 8.1.3 最大有效权限与删除ACL权限

#### 最大有效权限mask

mask是用来指定最大有效权限的。如果我给用户赋予了ACL权限，是需要和mask的权限“相与”才能得到用户的真正权限

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **and** |
| r | r | r |
| r | - | - |
| - | r | - |
| - | - | - |

修改最大有效权限

[root@localhost /]# setfacl -m m:rx 文件名

#设定mask权限为r-x。使用“m:权限”格式

#### 删除ACL权限

[root@localhost /]# setfacl -x u:用户名 文件名

#删除指定用户的ACL权限

[root@localhost /]# setfacl -x g:组名 文件名

#删除指定用户组的ACL权限

[root@localhost /]# setfacl -b 文件名

#会删除文件的所有的ACL权限

### 8.1.4 默认ACL权限和递归ACL权限

#### 递归ACL权限

递归是父目录在设定ACL权限时，所有的子文件和子目录也会拥有相同的ACL权限。

setfacl -m u:用户名:权限 -R 文件名

#### 默认ACL权限

默认ACL权限的作用是如果给父目录设定了默认ACL权限，那么父目录中所有新建的子文件都会继承父目录的ACL权限。

setfacl -m d:u:用户名:权限 文件名

## 8.2 文件特殊权限

### 8.2.1 SetUID

#### SetUID的功能

（1）只有可以执行的二进制程序才能设定SUID权限

（2）命令执行者要对该程序拥有x（执行）权限

（3）命令执行者在执行该程序时获得该程序文件属主的身份（在执行程序的过程中灵魂附体为文件的属主）

（4）SetUID权限只在该程序执行过程中有效，也就是说身份改变只在程序执行过程中有效

（5）passwd命令拥有SetUID权限，所以普通可以修改自己的密码

[root@localhost ~]# ll /usr/bin/passwd

-rwsr-xr-x. 1 root root 25980 2月 22 2012 /usr/bin/passwd

（6）cat命令没有SetUID权限，所以普通用户不能查看/etc/shadow文件内容

[root@localhost ~]# ll /bin/cat

-rwxr-xr-x 1 root root 47976 6月 22 2012 /bin/cat



#### 设定SetUID的方法

4代表SUID

chmod 4755 文件名

chmod u+s 文件名

#### 取消SetUID的方法

chmod 755 文件名

chmod u-s 文件名

#### 危险的SetUID

（1）关键目录应严格控制写权限。比如“/”、“/usr”等

（2）用户的密码设置要严格遵守密码三原则

（3）对系统中默认应该具有SetUID权限的文件作一列表，定时检查有没有这之外的文件被设置了SetUID权限

### 8.2.2 SetGID

#### SetGID针对文件的作用

（1）只有可执行的二进制程序才能设置SGID权限

（2）命令执行者要对该程序拥有x（执行）权限

（3）命令执行在执行程序的时候，组身份升级为该程序文件的属组

（4）SetGID权限同样只在该程序执行过程中有效，也就是说组身份改变只在程序执行过程中有效

[root@localhost ~]# ll /usr/bin/locate

-rwx--s--x 1 root slocate 35612 8月 24 2010 /usr/bin/locate

[root@localhost ~]# ll /var/lib/mlocate/mlocate.db

-rw-r----- 1 root slocate 1838850 1月 20 04:29 /var/lib/mlocate/mlocate.db

/usr/bin/locate是可执行二进制程序，可以赋予SGID

执行用户lamp对/usr/bin/locate命令拥有执行权限

执行/usr/bin/locate命令时，组身份会升级为slocate组，而slocate组对/var/lib/mlocate/mlocate.db数据库拥有r权限，所以普通用户可以使用locate命令查询mlocate.db数据库

命令结束，lamp用户的组身份返回为lamp组

#### SetGID针对目录的作用

（1）普通用户必须对此目录拥有r和x权限，才能进入此目录

（2）普通用户在此目录中的有效组会变成此目录的属组

（3）若普通用户对此目录拥有w权限时，新建的文件的默认属组是这个目录的属组

[root@localhost ~]# cd /tmp/

[root@localhost tmp]# mkdir dtest

[root@localhost tmp]# chmod g+s dtest

[root@localhost tmp]# ll -d dtest/

[root@localhost tmp]# chmod 777 dtest/

[root@localhost tmp]# su – lamp

[lamp@localhost ~]$ cd /tmp/dtest/

[lamp@localhost dtest]$ touch abc

[lamp@localhost dtest]$ ll

#### 设定SetGID

2代表SGID

chmod 2755 文件名

chmod g+s 文件名

#### 取消SetGID

chmod 755 文件名

chmod g-s 文件名

### 8.3.3 Sticky BIT

#### SBIT粘着位作用

（1）粘着位目前只对目录有效

（2）普通用户对该目录拥有w和x权限，即普通用户可以在此目录拥有写入权限

（3）如果没有粘着位，因为普通用户拥有w权限，所以可以删除此目录下所有文件，包括其他用户建立的文件。一但赋予了粘着位，除了root可以删除所有文件，普通用户就算拥有w权限，也只能删除自己建立的文件，但是不能删除其他用户建立的文件

[root@localhost ~]# ll -d /tmp/

drwxrwxrwt. 3 root root 4096 12月 13 11:22 /tmp/

#### 设置与取消粘着位

（1）设置粘着位

chmod 1755 目录名

chmod o+t 目录名

（2）取消粘着位

 chmod 777 目录名

 chmod o-t 目录名

## 8.3 文件系统属性chattr权限

#### chattr命令格式

[root@localhost ~]# chattr [+-=] [选项] 文件或目录名

+: 增加权限

-: 删除权限

=: 等于某权限

选项：

i：如果对文件设置i属性，那么不允许对文件进行删除、改名，也不能添加和修改数据；如果对目录设置i属性，那么只能修改目录下文件的数据，但不允许建立和删除文件。

a：如果对文件设置a属性，那么只能在文件中增加数据，但是不能删除也不能修改数据；如果对目录设置a属性，那么只允许在目录中建立和修改文件，但是不允许删除

#### 查看文件系统属性

[root@localhost ~]# lsattr 选项 文件名

选项：

-a 显示所有文件和目录

-d 若目标是目录，仅列出目录本身的属 性，而不是子文件的

## 8.4 系统命令sudo权限

#### sudo权限

（1）root把本来只能超级用户执行的命令赋予普通用户执行。

（2）sudo的操作对象是系统命令

#### sudo使用

[root@localhost ~]# visudo

#实际修改的是/etc/sudoers文件

root ALL=(ALL) ALL

#用户名 被管理主机的地址=（可使用的身份） 授权命令（绝对路径）

# %wheel ALL=(ALL) ALL

#%组名 被管理主机的地址=（可使用的身份） 授权命令（绝对路径）

#### 授权sc用户可以重启服务器

[root@localhost ~]# visudo

sc ALL= /sbin/shutdown –r now

#### 普通用户执行sudo赋予的命令

[root@localhost ~]# su - lixiaodan

[sc@localhost ~]$ sudo -l

#查看可用的sudo命令

[lamp@localhost ~]$ sudo /sbin/shutdown -r now

#普通用户执行sudo赋予的命令

**注意：对于赋予权限这块千万不能赋予普通用户类似操作vim这种工具类的软件的执行权限，这样，普通用户很可能以超级用户的身份去执行这样的软件，然后修改重要的配置文件，这是一件非常不安全的事情，非常非常不安全，出现这种情况，完全是root管理员的问题！！！千万要避免这样的情况发生！**

# 第九章 文件系统管理

## 9.1 回顾分区和文件系统

#### 分区类型

（1）主分区：总共最多只能分四个

（2）扩展分区：只能有一个，也算作主分区的一种，也就是说主分区加扩展分区最多有四个。但是扩展分区不能存储数据和格式化，必须再划分成逻辑分区才能使用。

（3）逻辑分区：逻辑分区是在扩展分区中划分的，如果是IDE硬盘，Linux最多支持59个逻辑分区，如果是SCSI硬盘Linux最多支持11个逻辑分区

#### 分区表示方法



|  |  |
| --- | --- |
| **分区设备文件名** | |
| 主分区1 | /dev/sda1 |
| 主分区2 | /dev/sda2 |
| 主分区3 | /dev/sda3 |
| 扩展分区 | /dev/sda4 |
| 逻辑分区1 | /dev/sda5 |
| 逻辑分区2 | /dev/sda6 |
| 逻辑分区3 | /dev/sda7 |



|  |  |
| --- | --- |
| **分区的设备文件名** | |
| 主分区1 | /dev/sda1 |
| 扩展分区 | /dev/sda2 |
| 逻辑分区1 | /dev/sda5 |
| 逻辑分区2 | /dev/sda6 |
| 逻辑分区3 | /dev/sda7 |

#### 文件系统

（1）ext2：是ext文件系统的升级版本，Red Hat Linux7.2版本以前的系统默认都是ext2文件系统。1993年发布，最大支持16TB的分区和最大2TB的文件（1TB=1024GB=1024\*1024KB）

（2）ext3： ext3文件系统是ext2文件系统的升级版本，最大的区别就是带日志功能，以在系统突然停止时提高文件系统的可靠性。支持最大16TB的分区和最大2TB的文件

（3）ext4：它是ext3文件系统的升级版。ext4 在性能、伸缩性和可靠性方面进行了大量改进。EXT4的变化可以说是翻天覆地的，比如向下兼容EXT3、最大1EB文件系统和16TB文件、无限数量子目录、Extents连续数据块概念、多块分配、延迟分配、持久预分配、快速FSCK、日志校验、无日志模式、在线碎片整理、inode增强、默认启用barrier等。是CentOS 6.3的默认文件系统 （1EB=1024PB=1024\*1024TB）

## 9.2 文件系统常用命令

### 9.2.1 df命令、du命令、fsck命令和dump2fs命令

#### 文件系统查看命令df

[root@localhost ~]# df [选项] [挂载点]

选项：

-a 显示所有的文件系统信息，包括特殊文件系统，如 /proc、/sysfs

-h 使用习惯单位显示容量，如KB，MB或GB等

-T 显示文件系统类型

-m 以MB为单位显示容量

-k 以KB为单位显示容量。默认就是以KB为单位

#### 统计目录或文件大小

[root@localhost ~]# du [选项] [目录或文件名]

选项：

-a 显示每个子文件的磁盘占用量。默认只统计 子目录的磁盘占用量

-h 使用习惯单位显示磁盘占用量，如KB，MB 或GB等

-s 统计总占用量，而不列出子目录和子文件的 占用量

#### du命令和df命令的区别

df命令是从文件系统考虑的，不光要考虑文件占用的空间，还要统计被命令或程序占用的空间（最常见的就是文件已经删除，但是程序并没有释放空间）

du命令是面向文件的，只会计算文件或目录占用的空间

#### 文件系统修复命令fsck（尽量不要使用）

[root@localhost ~]# fsck [选项] 分区设备文件名

选项：

-a： 不用显示用户提示，自动修复文件系统

-y： 自动修复。和-a作用一致，不过有些文件系统只支 持-y

#### 显示磁盘状态命令dumpe2fs

[root@localhost ~]# dumpe2fs 分区设备文件名

### 9.2.2 挂载命令

#### 查询与自动挂载

[root@localhost ~]# mount [-l]

#查询系统中已经挂载的设备，-l会显示卷标名称

[root@localhost ~]# mount –a

#依据配置文件/etc/fstab的内容，自动挂载

#### 挂载命令格式

[root@localhost ~]# mount [-t 文件系统] [-L 卷标名] [-o 特殊选项] 设备文件名 挂载点

选项：

-t 文件系统：加入文件系统类型来指定挂载的类型，可以ext3、ext4 、iso9660等文件系统

-L 卷标名： 挂载指定卷标的分区，而不是安装设备文件名挂载

-o 特殊选项：可以指定挂载的额外选项

|  |  |
| --- | --- |
| **参数** | **说明** |
| atime/noatime | |  | | --- | | 更新访问时间/不更新访问时间。访问分区文件时，是否更新文件的访问时间，默认为更新 | |
| async/sync | 异步/同步，默认为异步 |
| auto/noauto | |  | | --- | | 自动/手动，mount -a命令执行时，是否会自动安装/etc/fstab文件内容挂载，默认为自动 | |
| defaults | 定义默认值，相当于rw,suid,dev,exec,auto,nouser,async这七个选项 |
| exec/noexec | 执行/不执行，设定是否允许在文件系统中执行可执行文件，默认是exec允许 |
| remount | 重新挂载已经挂载的文件系统，一般用于指定修改特殊权限 |
| rw/ro | 读写/只读，文件系统挂载时，是否具有读写权限，默认是rw |
| suid/nosuid | 具有/不具有SUID权限，设定文件系统是否具有SUID和SGID的权限，默认是具有 |
| user/nouser | 允许/不允许普通用户挂载，设定文件系统是否允许普通用户挂载，默认是不允许，只有root可以挂载分区 |
| usrquota | 写入代表文件系统支持用户磁盘配额，默认不支持 |
| grpquota | 写入代表文件系统支持组磁盘配额，默认不支持 |

[root@localhost ~]# mount -o remount,noexec /home

#重新挂载/boot分区，并使用noexec权限

[root@localhost sh]# cd /home

[root@localhost boot]# vi hello.sh

[root@localhost boot]# chmod 755 hello.sh

[root@localhost boot]# ./hello.sh

[root@localhost boot]# mount -o remount,exec /home

#记得改回来啊，要不会影响系统启动的

### 9.2.3 挂载光盘与U盘

#### 挂载光盘

[root@localhost ~]# mkdir /mnt/cdrom/

#建立挂载点

[root@localhost ~]# mount -t iso9660 /dev/cdrom /mnt/cdrom/

#挂载光盘

[root@localhost ~]# mount /dev/sr0 /mnt/cdrom/

#### 卸载命令

[root@localhost ~]# umount 设备文件名或挂载点

[root@localhost ~]# umount /mnt/cdrom

#### 挂载U盘

[root@localhost ~]# fdisk –l

#查看U盘设备文件名

[root@localhost ~]# mount -t vfat /dev/sdb1 /mnt/usb/

注意：Linux默认是不支持NTFS文件系统的

### 9.2.4 支持NTFS文件系统

#### 下载NTFS-3G插件

<http://www.tuxera.com/community/ntfs-3g-download/>

#### 安装NTFS-3G

[root@localhost ~]# tar -zxvf ntfs-3g\_ntfsprogs-2013.1.13.tgz

#解压

[root@localhost ~]# cd ntfs-3g\_ntfsprogs-2013.1.13

#进入解压目录

[root@localhost ntfs-3g\_ntfsprogs-2013.1.13]# ./configure

#编译器准备。没有指定安装目录，安装到默认位置中

[root@localhost ntfs-3g\_ntfsprogs-2013.1.13]# make

#编译

[root@localhost ntfs-3g\_ntfsprogs-2013.1.13]# make install

#编译安装

#### 使用

[root@localhost ~]# mount -t ntfs-3g 分区设备文件名 挂载点

## 9.3 fdisk分区

### 9.3.1 fdisk命令分区过程

#### 添加新硬盘

这里采用虚拟机的方法添加新硬盘

#### 查看新硬盘

[root@localhost ~]# fdisk -l

#### 使用fdisk命令分区

[root@localhost ~]# fdisk /dev/sdb

|  |  |
| --- | --- |
| **fdisk交互指令说明** | |
| **命令** | **说明** |
| a | 设置可引导标记 |
| b | 编辑bsd磁盘标签 |
| c | 设置DOS操作系统兼容标记 |
| d | 删除一个分区 |
| l | 显示已知的文件系统类型。82为Linux swap分区，83为Linux分区 |
| m | 显示帮助菜单 |
| n | 新建分区 |
| o | 建立空白DOS分区表 |
| p | 显示分区列表 |
| q | 不保存退出 |
| s | 新建空白SUN磁盘标签 |
| t | 改变一个分区的系统ID |
| u | 改变显示记录单位 |
| v | 验证分区表 |
| w | 保存退出 |
| x | 附加功能（仅专家） |

#### 重新读取分区表信息

[root@localhost ~]# partprobe

#### 格式化分区

[root@localhost ~]# mkfs -t ext4 /dev/sdb1

#### 建立挂载点并挂载

[root@localhost ~]# mkdir /disk1

[root@localhost ~]# mount /dev/sdb1 /disk1/

### 9.3.2 分区自动挂载与fstab文件修复

#### /etc/fstab文件

第一字段：分区设备文件名或UUID（硬盘通用唯一识别码）

第二字段：挂载点

第三字段：文件系统名称

第四字段：挂载参数

第五字段：指定分区是否被dump备份，0代表不备份，1代表每天备份，2代表不定期备份

第六字段：指定分区是否被fsck检测，0代表不检测，其他数字代表检测的优先级，那么当然1的优先级比2高

#### 分区自动挂载

[root@localhost ~]# vi /etc/fstab

…省略部分输出…

/dev/sdb5 /disk5 ext4 defaults 1 2

#### /etc/fstab文件修复

[root@localhost ~]# mount -o remount,rw /

## 9.4 分配swap分区

#### 新建swap分区

[root@localhost ~]# fdisk /dev/sdb

别忘记把分区ID改为82

#### 格式化

[root@localhost ~]# mkswap /dev/sdb1

#### 加入swap分区

[root@localhost ~]# swapon /dev/sdb1

#加入swap分区

[root@localhost ~]# swapoff /dev/sdb1

#取消swap分区

#### swap分区开机自动挂载

[root@localhost ~]# vi /etc/fstab

/dev/sdb1 swap swap defaults 0 0

#### free命令

[root@localhost ~]# free #查看内存与swap分区使用状况

cached（缓存）：是指把读取出来的数据保存在内存当中，当再次读取时，不用读取硬盘而直接从内存当中读取，加速了数据的读取过程

buffer（缓冲）：是指在写入数据时，先把分散的写入操作保存到内存当中，当达到一定程度再集中写入硬盘，减少了磁盘碎片和硬盘的反复寻道，加速了数据的写入过程

# 第十章 Shell基础

## 10.1 Shell概述

#### Shell是什么

Shell是一个命令行解释器，它为用户提供了一个向Linux内核发送请求以便运行程序的界面系统级程序，用户可以用Shell来启动、挂起、停止甚至是编写一些程序。



Shell还是一个功能相当强大的编程语言，易编写，易调试，灵活性较强。Shell是解释执行的脚本语言，在Shell中可以直接调用Linux系统命令。

#### Shell的分类

（1）Bourne Shell：从1979起Unix就开始使用Bourne Shell，Bourne Shell的主文件名为sh。

（2）C Shell： C Shell主要在BSD版的Unix系统中使用，其语法和C语言相类似而得名。

Shell的两种主要语法类型有Bourne和C，这两种语法彼此不兼容。Bourne家族主要包括sh、ksh、Bash、psh、zsh；C家族主要包括：csh、tcsh

Bash： Bash与sh兼容，现在使用的Linux就是使用Bash作为用户的基本Shell。

#### Linux支持的Shell

/etc/shells

## 10.2 Shell脚本的执行方式

#### echo输出命令

[root@localhost ~]# echo [选项] [输出内容]

选项：

-e： 支持反斜线控制的字符转换

|  |  |
| --- | --- |
| **控制字符** | **作 用** |
| \\ | 输出\本身 |
| \a | 输出警告音 |
| \b | 退格键，也就是向左删除键 |
| \c | 取消输出行末的换行符。和“-n”选项一致 |
| \e | ESCAPE键 |
| \f | 换页符 |
| \n | 换行符 |
| \r | 回车键 |
| \t | 制表符，也就是Tab键 |
| \v | 垂直制表符 |
| \0nnn | 按照八进制ASCII码表输出字符。其中0为数字零，nnn是三位八进制数 |
| \xhh | 按照十六进制ASCII码表输出字符。其中hh是两位十六进制数 |

[root@localhost ~]# echo -e "ab\bc"

#删除左侧字符

[root@localhost ~]# echo -e "a\tb\tc\nd\te\tf"

#制表符与换行符

[root@localhost ~]# echo -e \

"\x61\t\x62\t\x63\n\x64\t\x65\t\x66"

#按照十六进制ASCII码也同样可以输出

[root@localhost ~]# echo -e "\e[1;31m abcd \e[0m"

#输出颜色

#30m=黑色，31m=红色，32m=绿色，33m=黄色

#34m=蓝色，35m=洋红，36m=青色，37m=白色

#### 第一个脚本

[root@localhost sh]# vi hello.sh

#!/bin/Bash

#The first program

# Author: XuZhenkang （E-mail: xuzhenkang@hotmail.com）

echo -e "XuZhenkang is the honest man in the world."

#### 脚本执行

赋予执行权限，直接运行

chmod 755 hello.sh

./hello.sh

通过Bash调用执行脚本

bash hello.sh

## 10.3 Bash的基本功能

### 10.3.1 历史命令与命令补全

#### 历史命令

[root@localhost ~]# history [选项] [历史命令保存文件]

选项：

-c： 清空历史命令

-w： 把缓存中的历史命令写入历史命令保存文件

~/.bash\_history

历史命令默认会保存1000条,可以在环境变量配置文件/etc/profile中进行修改

#### 历史命令的调用

使用上、下箭头调用以前的历史命令

使用“!n”重复执行第n条历史命令

使用“!!”重复执行上一条命令

使用“!字串”重复执行最后一条以该字串开头的命令

#### 命令与文件补全

在Bash中，命令与文件补全是非常方便与常用的功能，我们只要在输入命令或文件时，按“Tab”键就会自动进行补全

### 10.3.2 命令别名与常用快捷键

#### 命令别名

[root@localhost ~]# alias 别名='原命令'

#设定命令别名

[root@localhost ~]# alias

#查询命令别名

#### 命令执行时顺序

1 第一顺位执行用绝对路径或相对路径执行的命令。

2 第二顺位执行别名。

3 第三顺位执行Bash的内部命令。

4 第四顺位执行按照$PATH环境变量定义的目录查找顺序找到的第一个命令。

#### 让别名永久生效

[root@localhost ~]# vi /root/.bashrc

#### 删除别名

[root@localhost ~]# unalias 别名

#### Bash常用快捷键

|  |  |
| --- | --- |
| **快捷键** | **作用** |
| ctrl+A | 把光标移动到命令行开头。如果我们输入的命令过长，想要把光标移动到命令行开头时使用。 |
| ctrl+E | 把光标移动到命令行结尾。 |
| ctrl+C | 强制终止当前的命令。 |
| ctrl+L | 清屏，相当于clear命令。 |
| ctrl+U | 删除或剪切光标之前的命令。我输入了一行很长的命令，不用使用退格键一个一个字符的删除，使用这个快捷键会更加方便 |
| ctrl+K | 删除或剪切光标之后的内容。 |
| ctrl+Y | 粘贴ctrl+U或ctrl+K剪切的内容。 |
| ctrl+R | 在历史命令中搜索，按下ctrl+R之后，就会出现搜索界面，只要输入搜索内容，就会从历史命令中搜索。 |
| ctrl+D | 退出当前终端。 |
| ctrl+Z | 暂停，并放入后台。这个快捷键牵扯工作管理的内容，我们在系统管理章节详细介绍。 |
| ctrl+S | 暂停屏幕输出。 |
| ctrl+Q | 恢复屏幕输出。 |

### 10.3.3 输入输出重定向

#### 标准输入输出

|  |  |  |  |
| --- | --- | --- | --- |
| **设备** | **设备文件名** | **文件描述符** | **类型** |
| 键盘 | /dev/stdin | 0 | 标准输入 |
| 显示器 | /dev/sdtout | 1 | 标准输出 |
| 显示器 | /dev/sdterr | 2 | 标准错误输出 |

#### 输出重定向

|  |  |  |
| --- | --- | --- |
| **类别** | **符号** | **作用** |
| 标准输出重定向 | 命令 > 文件 | 以覆盖的方式，把命令的正确输出输出到指定的文件或设备当中。 |
| 命令 >> 文件 | 以追加的方式，把命令的正确输出输出到指定的文件或设备当中。 |
| 标准错误输出重定向 | 错误命令 2>文件 | 以覆盖的方式，把命令的错误输出输出到指定的文件或设备当中。 |
| 错误命令 2>>文件 | 以追加的方式，把命令的错误输出输出到指定的文件或设备当中。 |
| 正确输出和错误输出同时保存 | 命令 > 文件 2>&1 | 以覆盖的方式，把正确输出和错误输出都保存到同一个文件当中。 |
| 命令 >> 文件 2>&1 | 以追加的方式，把正确输出和错误输出都保存到同一个文件当中。 |
| 命令 &>文件 | 以覆盖的方式，把正确输出和错误输出都保存到同一个文件当中。 |
| 命令 &>>文件 | 以追加的方式，把正确输出和错误输出都保存到同一个文件当中。 |
| 命令>>文件1 2>>文件2 | 把正确的输出追加到文件1中，把错误的输出追加到文件2中。 |

#### 输入重定向

[root@localhost ~]# wc [选项] [文件名]

选项：

-c 统计字节数

-w 统计单词数

-l 统计行数

命令<文件 把文件作为命令的输入

命令<< 标识符 … 标识符 把标识符之间内容作为 命令的输入

### 10.3.4 多命令顺序执行与管道符

#### 多命令顺序执行

|  |  |  |
| --- | --- | --- |
| **多命令执行符** | **格式** | **作用** |
| ; | 命令1;命令2 | 多个命令顺序执行，命令之间没有任何逻辑联系 |
| && | 命令1&&命令2 | 逻辑与  当命令1正确执行，则命令2才会执行  当命令1执行不正确，则命令2不会执行 |
| || | 命令1||命令2 | 逻辑或  当命令1执行不正确，则命令2才会执行  当命令1正确执行，则命令2不会执行 |

例子：

[root@localhost ~]# ls ; date ; cd /user ; pwd

[root@localhost ~]# dd if=输入文件 of=输出文件 bs=字节数 count=个数

选项：

if=输入文件 指定源文件或源设备

of=输出文件 指定目标文件或目标设备

bs=字节数 指定一次输入/输出多少字节，即把这些字节看做 一个数据块

count=个数 指定输入/输出多少个数据块

例子：

[root@localhost ~]# date ; dd if=/dev/zero of=/root/testfile bs=1k count=100000 ; date

[root@localhost ~]# ls anaconda-ks.cfg && echo yes

[root@localhost ~]# ls /root/test || echo "no

[root@localhost ~]# 命令 && echo yes || echo no

#### 管道符

命令格式：

[root@localhost ~]# 命令1 | 命令2

#命令1的正确输出作为命令2的操作对象

颜色显示

例子：

[root@localhost ~]# ll -a /etc/ | more

[root@localhost ~]# netstat -an | grep "ESTABLISHED"

[root@localhost ~]# grep [选项] "搜索内容" 文件名

选项：

-i： 忽略大小写

-n： 输出行号

-v： 反向查找

--color=auto 搜索出的关键字用颜色显示

### 10.3.5 通配符与其他特殊符号

#### 通配符

|  |  |
| --- | --- |
| **通配符** | **作用** |
| ? | 匹配一个任意字符 |
| \* | 匹配0个或任意多个任意字符，也就是可以匹配任何内容 |
| [] | 匹配中括号中任意一个字符。例如：[abc]代表一定匹配一个字符，或者是a，或者是b，或者是c。 |
| [-] | 匹配中括号中任意一个字符，-代表一个范围。例如：[a-z]代表匹配一个小写字母。 |
| [^] | 逻辑非，表示匹配不是中括号内的一个字符。例如：[^0-9]代表匹配一个不是数字的字符。 |

[root@localhost ~]# cd /tmp/

[root@localhost tmp]# rm -rf \*

[root@localhost tmp]# touch abc

[root@localhost tmp]# touch abcd

[root@localhost tmp]# touch 012

[root@localhost tmp]# touch 0abc

[root@localhost tmp]# ls ?abc

[root@localhost tmp]# ls [0-9]\*

[root@localhost tmp]# ls [^0-9]\*

#### Bash中其他特殊符号

|  |  |
| --- | --- |
| **符号** | **作用** |
| '' | 单引号。在单引号中所有的特殊符号，如“$”和“`”(反引号)都没有特殊含义。 |
| "" | 双引号。在双引号中特殊符号都没有特殊含义，但是“$”、“`”和“\”是例外，拥有“调用变量的值”、“引用命令”和“转义符”的特殊含义。 |
| `` | 反引号。反引号括起来的内容是系统命令，在Bash中会先执行它。和$()作用一样，不过推荐使用$()，因为反引号非常容易看错。 |
| $() | 和反引号作用一样，用来引用系统命令。 |
| # | 在Shell脚本中，#开头的行代表注释。 |
| $ | 用于调用变量的值，如需要调用变量name的值时，需要用$name的方式得到变量的值。 |
| \ | 转义符，跟在\之后的特殊符号将失去特殊含义，变为普通字符。如\$将输出“$”符号，而不当做是变量引用。 |

#### 反引号与$()

[root@localhost ~]# echo `ls`

[root@localhost ~]# echo $(date)

#### 单引号与双引号

[root@localhost ~]# name=xzk

[root@localhost ~]# echo '$name'

[root@localhost ~]# echo "$name"

[root@localhost ~]# echo ‘$(date)'

[root@localhost ~]# echo “$(date)"

## 10.4 Bash的变量

### 10.4.1 用户自定义变量

#### 什么是变量

变量是计算机内存的单元，其中存放的值可以改变。当Shell脚本需要保存一些信息时，如一个文件名或是一个数字，就把它存放在一个变量中。每个变量有一个名字，所以很容易引用它。使用变量可以保存有用信息，使系统获知用户相关设置，变量也可以用于保存暂时信息。

#### 变量设置规则

（1）变量名称可以由字母、数字和下划线组成，但是不能以数字开头。如果变量名是“2name”则是错误的。

（2）在Bash中，变量的默认类型都是字符串型，如果要进行数值运算，则必修指定变量类型为数值型。

（3）变量用等号连接值，等号左右两侧不能有空格。

（4）变量的值如果有空格，需要使用单引号或双引号包括。

（5）在变量的值中，可以使用“\”转义符。

（6）如果需要增加变量的值，那么可以进行变量值的叠加。不过变量需要用双引号包含“$变量名”或用${变量名}包含。

（7）如果是把命令的结果作为变量值赋予变量，则需要使用反引号或$()包含命令。

（8）环境变量名建议大写，便于区分。

#### 变量分类

（1）用户自定义变量

（2）环境变量：这种变量中主要保存的是和系统操作环境相关的数据。

（3）位置参数变量：这种变量主要是用来向脚本当中传递参数或数据的，变量名不能自定义，变量作用是固定的。

（4）预定义变量：是Bash中已经定义好的变量，变量名不能自定义，变量作用也是固定的。

#### 本地变量

（1）变量定义

[root@localhost ~]# name="xuzhenkang"

（2）变量叠加

[root@localhost ~]# aa=123

[root@localhost ~]# aa="$aa"456

[root@localhost ~]# aa=${aa}789

（3）变量调用

[root@localhost ~]# echo $name

（4）变量查看

[root@localhost ~]# set

（5）变量删除

[root@localhost ~]# unset name

### 10.4.2 环境变量

#### 环境变量是什么

用户自定义变量只在当前的Shell中生效，而环境变量会在当前Shell和这个Shell的所有子Shell当中生效。如果把环境变量写入相应的配置文件，那么这个环境变量就会在所有的Shell中生效

#### 设置环境变量

export 变量名=变量值

#申明变量

env

#查询变量

unset 变量名

#删除变量

#### 系统常见环境变量

PATH：系统查找命令的路径

[root@localhost ~]# echo $PATH

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

PATH="$PATH":/root/sh #PATH变量叠加

PS1：定义系统提示符的变量

\d：显示日期，格式为“星期 月 日”

\h：显示简写主机名。如默认主机名“localhost”

\t：显示24小时制时间，格式为“HH:MM:SS”

\T：显示12小时制时间，格式为“HH:MM:SS”

\A：显示24小时制时间，格式为“HH:MM”

\u：显示当前用户名

\w：显示当前所在目录的完整名称

\W：显示当前所在目录的最后一个目录

\#：执行的第几个命令

\$：提示符。如果是root用户会显示提示符为“#”，如果是普通用户会显示提示符为“$”

举例：

[root@localhost ~]# PS1='[\u@\t \w]\$ '

[root@04:50:08 /usr/local/src]#PS1='[\u@\@ \h \# \W]\$‘

[root@04:53 上午 localhost 31 src]#PS1='[\u@\h \W]\$ '

### 10.4.3 位置参数变量

#### 位置参数变量

|  |  |
| --- | --- |
| **位置参数变量** | **作用** |
| $n | n为数字，$0代表命令本身，$1-$9代表第一到第九个参数，十以上的参数需要用大括号包含，如${10}. |
| $\* | 这个变量代表命令行中所有的参数，$\*把所有的参数看成一个整体 |
| $@ | 这个变量也代表命令行中所有的参数，不过$@把每个参数区分对待 |
| $# | 这个变量代表命令行中所有参数的个数 |

例子1：

#!/bin/bash

num1=$1

num2=$2

sum=$(( $num1 + $num2))

#变量sum的和是num1加num2

echo $sum

#打印变量sum的值

例子2：

#!/bin/bash

echo "A total of $# parameters"

#使用$#代表所有参数的个数

echo "The parameters is: $\*"

#使用$\*代表所有的参数

echo "The parameters is: $@"

#使用$@也代表所有参数

例子3：$\*与$@的区别

#!/bin/bash

for i in "$\*"

#$\*中的所有参数看成是一个整体，所以这个for循环只会循环一次

do

echo "The parameters is: $i"

done

x=1

for y in "$@"

#$@中的每个参数都看成是独立的，所以“$@”中有几个参数，就会循环几次

do

echo "The parameter$x is: $y"

x=$(( $x +1 ))

done

### 10.4.4 预定义变量

#### 预定义变量

|  |  |
| --- | --- |
| **预定义变量** | **作用** |
| $？ | 最后一次执行的命令的返回状态。如果这个变量的值为0，证明上一个命令正确执行；如果这个变量的值为非0（具体是哪个数，由命令自己来决定），则证明上一个命令执行不正确了。 |
| $$ | 当前进程的进程号（PID） |
| $! | 后台运行的最后一个进程的进程号（PID） |

#!/bin/bash

# Author: Xu Zhenkang （E-mail: Xuzhenkang@hotmail.com）

echo "The current process is $$"

#输出当前进程的PID。

#这个PID就是variable.sh这个脚本执行时，生成的进程的PID

find /root -name hello.sh &

#使用find命令在root目录下查找hello.sh文件

#符号&的意思是把命令放入后台执行，工作管理我们在系统管理章节会详细介绍

echo "The last one Daemon process is $!"

#### 接收键盘输入

[root@localhost ~]# read [选项] [变量名]

选项：

-p “提示信息”：在等待read输入时，输出提示信息

-t 秒数： read命令会一直等待用户输入，使用 此选项可以指定等待时间

-n 字符数： read命令只接受指定的字符数，就会 执行

-s： 隐藏

#!/bin/bash

# Author: Xu Zhenkang （E-mail: Xuzhenkang@hotmail.com）

read -t 30 -p "Please input your name: " name

#提示“请输入姓名”并等待30秒，把用户的输入保存入变量name中

echo "Name is $name "

read -s -t 30 -p "Please enter your age: " age

#年龄是隐私，所以我们用“-s”选项隐藏输入

echo -e "\n"

echo "Age is $age "

read -n 1 -t 30 -p "Please select your gender[M/F]: " gender

#使用“-n 1”选项只接收一个输入字符就会执行（都不用输入回车）

echo -e "\n"

echo "Sex is $gender"

## 10.5 Bash的运算符

### 10.5.1 数值运算与运算符

#### declare声明变量类型

[root@localhost ~]# declare [+/-][选项] 变量名

选项：

-： 给变量设定类型属性

+： 取消变量的类型属性

-i： 将变量声明为整数型（integer）

-x： 将变量声明为环境变量

-p： 显示指定变量的被声明的类型

#### 数值运算

**方法1**

[root@localhost ~]# aa=11

[root@localhost ~]# bb=22

#给变量aa和bb赋值

[root@localhost ~]# declare -i cc=$aa+$bb

**方法2：expr或let数值运算工具**

[root@localhost ~]# aa=11

[root@localhost ~]# bb=22

#给变量aa和变量bb赋值

[root@localhost ~]# dd=$(expr $aa + $bb)

#dd的值是aa和bb的和。注意“+”号左右两侧必须有空格

**方法3：“$((运算式))”或“$[运算式]”**

[root@localhost ~]# aa=11

[root@localhost ~]# bb=22

[root@localhost ~]# ff=$(( $aa+$bb ))

[root@localhost ~]# gg=$[ $aa+$bb ]

#### 运算符

|  |  |  |
| --- | --- | --- |
| **优先级** | **运算符** | **说明** |
| 13 | -,+ | 单目负、单目正 |
| 12 | !,~ | 逻辑非、按位取反或补码 |
| 11 | \*,/,% | 乘、除、取模 |
| 10 | +,- | 加、减 |
| 9 | <<,>> | 按位左移、按位右移 |
| 8 | <=,>=,<,> | 小于或等于、大于或等于、小于、大于 |
| 7 | ==,!= | 等于、不等于 |
| 6 | & | 按位与 |
| 5 | ^ | 按位异或 |
| 4 | | | 按位或 |
| 3 | && | 逻辑与 |
| 2 | || | 逻辑或 |
| 1 | =,+=,-=,\*=,/=,%=,&=,^=,|=,<<=,>>= | 赋值、运算且赋值 |

[root@localhost ~]# aa=$(( (11+3)\*3/2 ))

#虽然乘和除的优先级高于加，但是通过小括号可以调整运算优先级

[root@localhost ~]# bb=$(( 14%3 ))

#14不能被3整除，余数是2

[root@localhost ~]# cc=$(( 1 && 0 ))

#逻辑与运算只有想与的两边都是1，与的结果才是1，否则与的结果是0

### 10.5.2 变量测试与内容替换

#### 变量测试与内容替换

|  |  |  |  |
| --- | --- | --- | --- |
| **变量置换方式** | **变量y没有设置** | **变量y为空值** | **变量y设置值** |
| x=${y-新值} | x=新值 | x为空 | x=$y |
| x=${y:-新值} | x=新值 | x=新值 | x=$y |
| x=${y+新值} | x为空 | x=新值 | x=新值 |
| x=${y:+新值} | x为空 | x为空 | x=新值 |
| x=${y=新值} | x=新值  y=新值 | x为空  y值不变 | x=$y  y值不变 |
| x=${y:=新值} | x=新值  y=新值 | x=新值  y=新值 | x=$y  y值不变 |
| x=${y?新值} | 新值输出到标准错误输出（就是屏幕） | x为空 | x=$y |
| x=${y:?新值} | 新值输出到标准错误输出 | 新值输出到标准错误输出 | x=$y |

例子1：测试x=${y-新值}

[root@localhost ~]# unset y

#删除变量y

[root@localhost ~]# x=${y-new}

#进行测试

[root@localhost ~]# echo $x

new

#因为变量y不存在，所以x=new

[root@localhost ~]# y=""

#给变量y赋值为空

[root@localhost ~]# x=${y-new}

#进行测试

[root@localhost ~]# echo $x

[root@localhost ~]# y=old

#给变量y赋值

[root@localhost ~]# x=${y-new}

#进行测试

[root@localhost ~]# echo $x

old

## 10.6 环境变量配置文件

### 10.6.1 环境变量配置文件简介

#### source命令

[root@localhost ~]# source 配置文件

或

[root@localhost ~]# . 配置文件

#### 环境变量配置文件简介

环境变量配置文件中主要是定义对系统的操作环境生效的系统默认环境变量，比如PATH、HISTSIZE、PS1、HOSTNAME等默认环境变量。

/etc/profile

/etc/profile.d/\*.sh

~/.bash\_profile

~/.bashrc

/etc/bashrc

### 10.6.2 环境变量配置文件作用

#### 环境变量配置文件

/etc/profile

/etc/profile.d/\*.sh

~/.bash\_profile

~/.bashrc

/etc/bashrc



#### /etc/profile的作用：

USER变量：

LOGNAME变量：

MAIL变量：

PATH变量：

HOSTNAME变量：

HISTSIZE变量：

umask：

调用/etc/profile.d/\*.sh文件

#### ~/.bash\_profile的作用

调用了~/.bashrc文件。

在PATH变量后面加入了“:$HOME/bin”这个目录

#### ~/.bashrc的作用

定义默认别名

调用/etc/bashrc

#### /etc/bashrc的作用

PS1变量

umask

PATH变量

调用/etc/profile.d/\*.sh文件

### 10.6.3 其他配置文件和登录信息

#### 注销时生效的环境变量配置文件

~/.bash\_logout

#### 其他配置文件

~/bash\_history

#### Shell登录信息

#### 本地终端欢迎信息： /etc/issue

|  |  |
| --- | --- |
| **转义符** | **作用** |
| \d | 显示当前系统日期 |
| \s | 显示操作系统名称 |
| \l | 显示登录的终端号，这个比较常用。 |
| \m | 显示硬件体系结构，如i386、i686等 |
| \n | 显示主机名 |
| \o | 显示域名 |
| \r | 显示内核版本 |
| \t | 显示当前系统时间 |
| \u | 显示当前登录用户的序列号 |

#### 远程终端欢迎信息： /etc/issue.net

转义符在/etc/issue.net文件中不能使用

是否显示此欢迎信息，由ssh的配置文件/etc/ssh/sshd\_config决定，加入“Banner /etc/issue.net”行才能显示（记得重启SSH服务）

#### 登陆后欢迎信息：/etc/motd

不管是本地登录，还是远程登录，都可以显示此欢迎信息

# 第十一章 Shell编程

## 11.1 基础正则表达式

#### 正则表达式与通配符

正则表达式用来在文件中匹配符合条件的字符串，正则是包含匹配。grep、awk、sed等命令可以支持正则表达式。

通配符用来匹配符合条件的文件名，通配符是完全匹配。ls、find、cp这些命令不支持正则表达式，所以只能使用shell自己的通配符来进行匹配了。

#### 基础正则表达式

|  |  |
| --- | --- |
| **元字符** | **作用** |
| \* | 前一个字符匹配0次或任意多次。 |
| . | 匹配除了换行符外任意一个字符。 |
| ^ | 匹配行首。例如：^hello会匹配以hello开头的行。 |
| $ | 匹配行尾。例如：hello&会匹配以hello结尾的行。 |
| [] | 匹配中括号中指定的任意一个字符，只匹配一个字符。  例如：[aoeiu]匹配任意一个元音字母，[0-9]匹配任意一位数字，[a-z][0-9]匹配小写字和一位数字构成的两位字符。 |
| [^] | 匹配除中括号的字符以外的任意一个字符。例如：[^0-9]匹配任意一位非数字字符，[^a-z]表示任意一位非小写字母。 |
| \ | 转义符。用于取消讲特殊符号的含义取消。 |
| \{n\} | 表示其前面的字符恰好出现n次。例如：[0-9]\{4\}匹配4位数字，[1][3-8][0-9]\{9\}匹配手机号码。 |
| \{n,\} | 表示其前面的字符出现不小于n次。例如：[0-9]\{2,\}表示两位及以上的数字。 |
| \{n,m\} | 表示其前面的字符至少出现n次，最多出现m次。例如：[a-z]\{6,8\}匹配6到8位的小写字母。 |

#### “\*”前一个字符匹配0次，或任意多次

grep "a\*" test\_rule.txt

#匹配所有内容，包括空白行

grep "aa\*" test\_rule.txt

#匹配至少包含有一个a的行

grep "aaa\*" test\_rule.txt

#匹配最少包含两个连续a的字符串

grep "aaaaa\*" test\_rule.txt

#则会匹配最少包含四个个连续a的字符串

#### “.” 匹配除了换行符外任意一个字符

grep "s..d" test\_rule.txt

#“s..d”会匹配在s和d这两个字母之间一定有两个字符的单词

grep "s.\*d" test\_rule.txt

#匹配在s和d字母之间有任意字符

grep ".\*" test\_rule.txt

#匹配所有内容

#### “^”匹配行首，“$”匹配行尾

grep "^M" test\_rule.txt

#匹配以大写“M”开头的行

grep "n$" test\_rule.txt

#匹配以小写“n”结尾的行

grep -n "^$" test\_rule.txt

#会匹配空白行

#### “[]” 匹配中括号中指定的任意一个字符，只匹配一个字符

grep "s[ao]id" test\_rule.txt

#匹配s和i字母中，要不是a、要不是o

grep "[0-9]" test\_rule.txt

#匹配任意一个数字

grep "^[a-z]" test\_rule.txt

#匹配用小写字母开头的行

#### “[^]” 匹配除中括号的字符以外的任意一个字符

grep "^[^a-z]" test\_rule.txt

#匹配不用小写字母开头的行

grep "^[^a-zA-Z]" test\_rule.txt

#匹配不用字母开头的行

“\” 转义符

grep "\.$" test\_rule.txt

#匹配使用“.”结尾的行

#### “\{n\}”表示其前面的字符恰好出现n次

grep "a\{3\}" test\_rule.txt

#匹配a字母连续出现三次的字符串

grep "[0-9]\{3\}" test\_rule.txt

#匹配包含连续的三个数字的字符串

#### “\{n,\}”表示其前面的字符出现不小于n次

grep "^[0-9]\{3,\}[a-z]" test\_rule.txt

#匹配最少用连续三个数字开头的行

#### “\{n,m\}”匹配其前面的字符至少出现n次，最多出现m次

grep "sa\{1,3\}i" test\_rule.txt

#匹配在字母s和字母i之间有最少一个a，最多三个a

## 11.2 字符截取命令

### 11.2.1 cut字段提取命令

#### cut字段提取

[root@localhost ~]# cut [选项] 文件名

选项：

-f 列号： 提取第几列

-d 分隔符： 按照指定分隔符分割列

[root@localhost ~]# vi student.txt

ID Name gender Mark

1 XZK M 86

2 LXD F 90

3 LXC M 83

**注意：此文件每列使用tab键分割**

[root@localhost ~]# cut -f 2 student.txt

[root@localhost ~]# cut -f 2,3 student.txt

[root@localhost ~]# cut -d ":" -f 1,3 /etc/passwd

#### cut命令的局限

[root@localhost ~]# df -h | cut -d " " -f 1,3

**多个空格作为分隔符无法使用cut进行分割**

### 11.2.2 printf命令

printf ‘输出类型输出格式’ 输出内容

输出类型：

%ns： 输出字符串。n是数字指代输出几个字符

%ni： 输出整数。n是数字指代输出几个数字

%m.nf： 输出浮点数。m和n是数字，指代输出的整数 位数和小数位数。如%8.2f代表共输出8位数， 其中2位是小数，6位是整数。

输出格式：

\a: 输出警告声音

\b: 输出退格键，也就是Backspace键

\f: 清除屏幕

\n: 换行

\r: 回车，也就是Enter键

\t: 水平输出退格键，也就是Tab键

\v: 垂直输出退格键，也就是Tab键

[root@localhost ~]# printf %s 1 2 3 4 5 6

[root@localhost ~]# printf %s %s %s 1 2 3 4 5 6

[root@localhost ~]# printf '%s %s %s' 1 2 3 4 5 6

[root@localhost ~]# printf '%s %s %s\n' 1 2 3 4 5 6

[root@localhost ~]# vi student.txt

ID Name PHP Linux MySQL Average

1 Lixiaodan 82 95 86 87.66

2 XZK 74 96 87 85.66

3 Zhang 99 83 93 91.66

printf '%s' $(cat student.txt)

#不调整输出格式

printf '%s\t %s\t %s\t %s\t %s\t %s\t \n' $(cat student.txt)

#调整格式输出

在awk命令的输出中支持print和printf命令

print：print会在每个输出之后自动加入一个换行符（Linux默认没有print命令）

printf：printf是标准格式输出命令，并不会自动加入换行符，如果需要换行，需要手工加入换行符

### 11.2.3 awk命令

# awk ‘条件1{动作1} 条件2{动作2}…’ 文件名

条件（Pattern）：

一般使用关系表达式作为条件

x > 10 判断变量 x是否大于10

x>=10 大于等于

x<=10 小于等于

动作（Action）：

格式化输出

流程控制语句

[root@localhost ~]# vi student.txt

ID Name PHP Linux MySQL Average

1 Liming 82 95 86 87.66

2 Sc 74 96 87 85.66

3 Gao 99 83 93 91.66

# awk '{printf $2 "\t" $6 "\n"}' student.txt

# df -h | awk '{print $1 "\t" $3}'

BEGIN

# awk 'BEGIN{printf "This is a transcript \n" }

{printf $2 "\t" $6 "\n"}' student.txt

END

# awk 'END{printf "The End \n" }

{printf $2 "\t" $6 "\n"}' student.txt

FS内置变量

# cat /etc/passwd | grep "/bin/bash" | \

awk 'BEGIN {FS=":"} {printf $1 "\t" $3 "\n"}'

关系运算符

# cat student.txt | grep -v Name | \

awk '$6 >= 87 {printf $2 "\n" }'

### 11.2.4 sed命令

## 11.3 字符处理命令

## 11.4 条件判断

## 11.5 流程控制