**Linux系统管理和服务项目教程**

图片包含 游戏机

描述已自动生成

目录

[项目一 Linux简介 4](#_Toc42851273)

[项目二 安装系统 17](#_Toc42851274)

[项目三 命令行技术 30](#_Toc42851275)

[项目四 VI编辑器 58](#_Toc42851276)

[项目五 用户和组 67](#_Toc42851277)

[项目六 权限管理 84](#_Toc42851278)

[项目七 网络连接 99](#_Toc42851279)

[项目八 软件安装 120](#_Toc42851280)

[项目九 磁盘管理 124](#_Toc42851281)

[项目十 计划任务 133](#_Toc42851282)

[项目十一 状态检测 138](#_Toc42851283)

[项目十二 日志系统 148](#_Toc42851284)

[项目十三 SSH服务器 155](#_Toc42851285)

[项目十四 RSYNC服务器 164](#_Toc42851286)

[项目十五 NTP服务器 170](#_Toc42851287)

[项目十六 DNS服务器 173](#_Toc42851288)

[项目十七 APACHE服务器 180](#_Toc42851289)

[项目十八 NGINX服务器 188](#_Toc42851290)

[项目十九 SQUID服务器 191](#_Toc42851291)

[项目二十 NFS服务器 196](#_Toc42851292)

[项目二十一 SAMBA服务器 203](#_Toc42851293)

[项目二十二 FTP服务器 209](#_Toc42851294)

[项目二十三 IPTABLES防火墙 219](#_Toc42851295)

# Linux简介

**【任务描述】**

虽然我们大多数时间在桌面电脑上都使用windows进行工作学习，但是你知道吗，在企业服务器领域，Linux操作系统却被广泛应用。面对即将开展的工作，我们大部分的需求是基于Linux系统。现在让我们一起来认识和了解Linux系统。

**【学习目标】**

* 1. 了解Linux的行业应用，特别是注意在企业的核心技术应用。
  2. 了解Linux的基本结构和技术特点，理解Linux为什么会被广泛应用到生产生活的各个领域。
  3. 了解什么是开源软件，以及开源软件的优缺点。
  4. 了解Linux各大主流发行版各自的特点。

## 认识Linux系统

Linux（['linэks][ˈlɪnəks]）[[1]](#footnote-1)是一种自由和开放源代码的类UNIX操作系统。该操作系统的内核由林纳斯•托瓦兹（Linus Torvalds）在1991年10月5日首次发布。在加上用户空间的应用程序之后，成为Linux操作系统。Linux也是自由软件和开放源代码软件发展中最著名的例子。只要遵循GNU通用公共许可证，任何个人和机构都可以自由地使用Linux的所有底层源代码，也可以自由地修改和再发布。大多数Linux系统还包括了像提供GUI界面的X Window之类的程序。除了一部分专家之外，大多数人都是直接使用Linux发布版，而不是自己选择每一样组件或自行设置。

严格来讲，术语Linux只表示操作系统内核本身，但通常采用Linux内核来表达该意思。Linux则常用来指基于Linux内核的完整操作系统，包括GUI组件和许多其他实用工具。由于这些支持用户空间的系统工具和库主要由理查德•斯托曼（Richard M. Stallman）于1983年发起的GNU计划提供，自由软件基金会提议将该组合系统命名为GNU/Linux，但Linux本身不属于GNU计划。

Linux最初是作为支持英特尔x86架构的个人计算机的一个自由操作系统。目前Linux已经被移植到更多的计算机硬件平台，远远超出其他任何操作系统。Linux是一个领先的操作系统，可以运行在服务器和其他大型平台之上，如大型主机和超级计算机。世界上500个最快的超级计算机90％以上运行Linux发行版或变种，包括最快的前10名超级计算机运行的都是基于Linux内核的操作系统。Linux也广泛应用在嵌入式系统上，如手机，平板电脑，路由器，电视和电子游戏机等。在移动设备上广泛使用的Android操作系统就是创建在Linux内核之上。

通常情况下，Linux被打包成供个人计算机和服务器使用的Linux发行版，一些流行的主流Linux发布版，包括Debian（及其派生版本Ubuntu，Linux Mint），Fedora（及其相关版本Red Hat Enterprise Linux，CentOS）和openSUSE等。Linux发行版包含Linux内核和支撑内核的实用程序和库，通常还带有大量可以满足各类需求的应用程序。个人计算机使用的Linux发行版通常包X Window和一个相应的桌面环境，如GNOME或KDE。桌面Linux操作系统常用的应用程序，包括Firefox网页浏览器，LibreOffice办公软件，GIMP图像处理工具等。由于Linux是自由软件，任何人都可以创建一个符合自己需求的Linux发行版。

## 探索Linux系统发展史

1. **未完成的Multics**

早期的计算机并不像现在的微型PC，随处可见，它们只出现在军事、科研和教育等领域，并且为数不多的计算机不仅慢还很难使用。20世纪60年代初期，麻省理工大学（MIT）开发了“兼容分时系统（Compatible Time-Sharing System,CTSS）”,他可以让大型机通过多个终端（terminal）联机进入使用主机资源。1965年前后，由美国电话及电报公司(AT&T)贝尔实验室、麻省理工大学(MIT)及通用电气公司(GE)计划开发一个多用途（General-Purpose）、分时（Time-Sharing）及多用户（Multi-User）的操作系统，也就是这个Multics（MULTiplexed Information and Computing System），其被设计运行在GE-645大型主机上。不过，这个项目由于太过复杂，整个目标过于庞大，糅合了太多的特性，进展太慢，Multics虽然发布了一些产品，但是性能都很低，于是到了1969年2月，AT&T最终撤出了投入Multics项目的资源，中止这项合作计划。不可否认，Multics系统是一个优秀的设计，后面出来的Unix系统一定程度上受到它的启发。

1. **Unix和BSD**

我们通常说Linux是一个UNIX- Like（类UNIX）操作系统，继承了UNIX高效、稳定、安全的特性，并与UNIX保持着高度兼容性，我们常用的Linux系统整合着大量原本在UNIX下的工具与服务。

UNIX操作系统，是美国AT&T公司贝尔实验室于1969年实现的操作系统。最早由肯·汤普逊（Ken Thompson），丹尼斯·里奇（Dennis Ritchie），道格拉斯·麦克罗伊（Douglas McIlroy），和乔伊·欧桑纳（Joe Ossanna）于1969年在AT&T贝尔实验室开发。于1971年首次发布，最初是完全用汇编语言编写，这是当时的一种普遍的做法。后来，在1973年用一个重要的开拓性的方法，Unix被丹尼斯·里奇用编程语言C（内核和I/O例外）重新编写。高级语言编写的操作系统具有的可用性，允许移植到不同的计算机平台更容易。

Unix在学术机构和大型企业中得到了广泛的应用，当时的UNIX拥有者AT&T公司以低廉甚至免费的许可将Unix源码授权给学术机构做研究或教学之用，许多机构在此源码基础上加以扩充和改进，形成了所谓的“Unix变种”，这些变种反过来也促进了Unix的发展，其中最著名的变种之一是由加州大学柏克莱分校开发的柏克莱软件包(BSD)产品。

后来AT&T意识到了Unix的商业价值，不再将Unix源码授权给学术机构，并对之前的Unix及其变种声明了版权权利。而BSD在Unix的历史发展中具有相当大的影响力，被很多商业厂家采用，成为很多商用Unix的基础。由于版权问题，4.4BSD完全删除了来自于AT&T的代码。尽管后来，非商业版的Unix系统又经过了很多演变，但其中有不少最终都是创建在BSD版本上（Linux、Minix等系统除外）。所以从这个角度上，4.4 BSD 又是所有自由版本Unix的基础，它们和System V及Linux等共同构成Unix操作系统这片璀璨的星空。BSD在发展中也逐渐派生出3个主要的分支：FreeBSD、OpenBSD和NetBSD。

此后的几十年中，Unix仍在不断变化，其版权所有者不断变更，授权者的数量也在增加。Unix的版权曾经为AT&T所有，之后Novell拥有了Unix，再之后Novell又将版权出售给了圣克鲁兹作业。有很多大公司在取得了Unix的授权之后，开发了自己的Unix产品，比如IBM的AIX、惠普公司的HP-UX、SUN的Solaris和硅谷图形公司的IRIX。

Unix因为其安全可靠，高效强大的特点在服务器领域得到了广泛的应用。直到GNU/Linux流行开始前，Unix也是科学计算、大型机、超级计算机等所用操作系统的主流。即使现在其仍然被应用于一些对稳定性要求极高的数据中心之上。

操作系统家族简图如图 1‑1所示：



图 1‑1 操作系统家族

**NOTE**

值得一提的是，BSD UNIX最先实现了TCP/IP, 除此之外，柏克莱大学还开发了现代计算机领域广泛使用的DB和DNS，非常的了不起。

1. **GNU计划**

1983年，理查德·马修·斯托曼（Richard M. Stallman）创立了GNU计划。这个计划有一个目标，是为了发展一个完全自由的类Unix操作系统。自1984年发起这个计划以来，在1985年，理查德·马修·斯托曼发起自由软件基金会并且在1989年撰写了GPL协议（开源软件最重要的版权协议之一）。1990年代早期，GNU开始大量的产生或收集各种系统所必备的组件，像是库、编译器、调试工具、文本编辑器、网页服务器，以及一个Unix的用户界面（Unix shell），但是像一些底层环境，如硬件驱动、守护进程运行内核（kernel）仍然不完整和陷于停顿，GNU计划中是在马赫微核（Mach microkernel）的架构之上开发系统内核，也就是所谓的GNU Hurd。但是这个基于Mach的设计异常复杂，发展进度则相对缓慢。林纳斯·托瓦兹曾说过如果GNU内核在1991年时可以用，他不会自己去写一个。

GNU计划是现代软件发展的重要力量，它倡导的开放、自由（Open source，Free software），吸引了大量的企业和个人开发者参与其中，为各个开源软件项目贡献代码，使得开源软件蓬勃发展，这也Linux迅速壮大并逐渐流行的基础。

1. **Minix**

MINIX是一个轻量的小型类Unix操作系统，是为在计算机科学用作教学而设计的，作者是安德鲁·斯图尔特·塔能鲍姆（Andrew Stuart Tanenbaum）。从第三版开始，MINIX是自由软件，而且被重新设计。

因为AT&T的政策改变，在Version 7 Unix推出之后，发布新的使用条款，将UNIX源代码私有化，在大学中不再能使用UNIX源代码。塔能鲍姆教授为了能在课堂上教授学生操作系统运作的细节，决定在不使用任何AT&T的源代码前提下，自行开发与UNIX兼容的操作系统，以避免版权上的争议。他以小型UNIX（mini-UNIX）之意，将它称为MINIX。

1. **Linux诞生**

1991年，芬兰人林纳斯.托瓦兹在赫尔辛基大学上学，对操作系统很好奇，并且对MINIX只在教育学术上使用的设计很不满意，于是他决定写一个更加实用的操作系统，这就是后来的Linux内核。

林纳斯·托瓦兹开始在MINIX上开发Linux内核，为MINIX写的软件也可以在Linux内核上使用。后来Linux成熟了，可以在自己上面开发自己了。为了让Linux可以在商业上使用，林纳斯·托瓦兹决定改变他原来的协议（这个协议会限制商业使用），使用GNU GPL协议来代替。采用GPL协议发布的Linux受到全世界开发者的广泛关注和参与，开发者致力于融合GNU元素到Linux中，做出一个有完整功能的、自由的操作系统。Linux诞生路线如图 1‑2所示。

1994年3月，Linux1.0版正式发布，Marc Ewing成立了Red Hat软件公司，成为最著名的Linux经销商之一。



图 1‑2 Linux诞生

**NOTE**

1991年10月，赫尔辛基大学学生林纳斯.托瓦兹在comp.os.minix上发布一则信息：

“Hello everybody out there using minix- I'm doing a (free) operation system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones.”

## 理解开源软件运动

1. **开源软件和开源软件运动**

开放源代码软件就是在GNU通用公共许可证（GPL）或其他开源软件协议下发布的软件，以保障软件用户自由使用及接触源代码的权利。这同时也保障了用户自行修改、复制以及再分发的权利。简而言之：所有公布软件源代码的程序，都可以称为开放源代码软件。开放源代码有时不仅仅指开放软件的源代码，它同时也是一种软件开发模式的名称。使用开放源代码开发模式的软件代表就有Linux操作系统。

开放源代码软件运动是一个主要由程序工程师及其它计算机用户参与的声势浩大的运动。它是自由软件运动的一个分支，但两者的差别并不明显。一般而言，自由软件运动是基于政治及哲学思想（有时被称为所谓黑客文化）的理想主义运动，而开放源代码运动则主要注重程序本身的质量提升。

1. **Linux的标志**

Linux的标志和吉祥物是一只名字叫做Tux的企鹅，如图 1‑4所示。标志的由来是因为Linus在澳洲时曾被一只动物园里的企鹅咬了一口，印象深刻，便选择了企鹅作为Linux的标志。更容易被接受的引申的说法是：企鹅代表南极，而南极又是全世界所共有的一块陆地。这也就代表Linux是所有人的Linux。

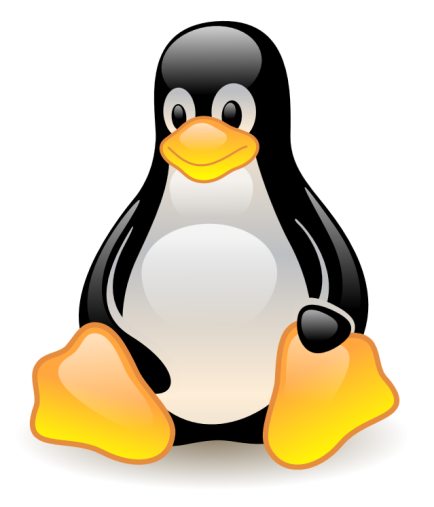


图 1-3

## 了解Linux系统特性和应用场景

1. **了解Linux系统的基本结构**

基于Linux的系统是一个模块化的类Unix操作系统。Linux操作系统的大部分设计思想来源于20世纪70年代到80年代的Unix操作系统所创建的基本设计思想。Linux系统使用单内核，由Linux内核负责处理进程控制、网络，以及外围设备和文件系统的访问。在系统运行的时候，设备驱动程序要么与内核直接集成，要么以加载模块形式添加。

Linux具有设备独立性，它内核具有高度适应能力，从而给系统提供了更高级的功能。GNU用户界面组件是大多数Linux操作系统的重要组成部分，提供常用的C函数库，shell，还有许多常见的Unix实用工具，可以完成许多基本的操作系统任务。大多数Linux系统使用的图形用户界面创建在X窗口系统之上，由X窗口系统通过软件工具及架构协议来创建操作系统所用的图形用户界面。

已安装Linux操作系统包含的一些组件：

1. 启动程序。例如GRUB或LILO。该程序在计算机开机启动的时候运行，并将Linux内核加载到内存中。
2. init程序。init是由Linux内核创建的第一个进程，称为根进程，所有的系统进程都是它的子进程，即所有的进程都是通过init启动。init启动的进程如系统服务和登录提示（图形或终端模式的选择）。
3. 软件库包含代码，可以通过运行的进程在Linux系统上使用ELF格式来执行文件，负责管理库使用的动态链接器是“ld-linux.so”。Linux系统上最常用的软件库是GNU C库。
4. 用户界面程序，如命令行shell或窗口环境。

### 了解Linux系统的应用方向

经过20多年的发展，Linux已成为最流行的操作系统之一，广泛应用于教育、科研、军事、企业以及个人计算机领域。因为良好的移植性，硬件兼容性，稳定高效，使它可以方便并可靠的部署在超级计算机、工作站、数据存储、网络服务器、嵌入式设备之上。

Linux系统典型的应用包括：

* + 超级计算机。在TOP500中，有485台运行Linux系统，超过97%（2014.6）。
  + 服务器。Linux发行版一直被用来作为服务器的操作系统，并且已经在该领域中占据重要地位。Linux发行版是构成LAMP（Linux操作系统，Apache，MySQL，Perl / PHP / Python）的重要部分，LAMP是一个常见的网站托管平台，在开发者中已经得到普及。
  + 工作站。《泰坦尼克号》、《我是传奇》、《指环王》、《星球大战》、《哈利波特》、《怪物史莱克》、《2012》、《阿凡达》等特效制作依赖于Linux的集群系统完成。
  + 个人计算机。随着Xwindow的加入，桌面环境发展和应用软件的极大丰富，Linux在图形界面易用性上也取得了长足的进步，产生了诸如Ubuntu、Fedora等优秀的桌面系统。
  + 嵌入式设备。Linux的低成本、强大的定制功能以及良好的移植性能，使得Linux在嵌入式系统方面也得到广泛应用。比如数字视频系统、音频系统、车载系统、光源系统、智能家居系统采用了定制的Linux；在网络防火墙和路由器也大多使用了定制的Linux。
  + 在智能手机、平板电脑等移动设备方面，基于Linux内核的操作系统也成为最广泛的操作系统。比如Android、Sailfish、Firefox OS 、Ubuntu Touch等。
  + 云计算。全球最大的云计算服务商Amazon EC2 云完全构建于Linux架构之上；流行的Openstack云计算解决方案基于Linux系统部署。

### Linux的特性

* + Linux是一种UNIX Like 操作系统，它遵循POSIX 标准,运行在UNIX下的软件很容易移植的Linux下，这使得Linux立刻拥有了大量优秀的软件。同时，Linux与UNIX非常相似，而它的开发人员大都拥有UNIX的背景。
  + 使用Linux，包括对它的拷贝，修改，再发布，在遵循GPL的协议下，你不会有任何版权问题的担心，对于企业部署可以极大的降低成本。而正因为支持Linux平台不会依赖于任何一家私有软件公司，所有各大软硬件厂商都支持并发展Linux，如REDHAT，IBM，INTEL，DELL，ORACLE，VMWARE，GOOGLE等。
  + 由于Linux的开发是基于internet由社区开发，并有众多的支持者进行测试和BUG提交，所以使得Linux拥有更快的更新速度，更透明的漏洞修补和功能迭进。
  + Linux继承了UNIX多用户多任务的设计理念，允许多人同时上线工作，并合理分配资源。严格的用户权限管理使得不同的使用者之间保持高度的保密性和安全性。
  + Linux系统使用相对较少的硬件资源，你甚至可以找出一台你已经丢弃的古董计算机，在上面安装上Linux，在上面跑一些网络服务，你会惊讶的发现它竟然非常流畅，一般情况下，你不会担心它会越来越慢。
  + Linux独特的内核设计决定的它的网络性能出色，不少的网络设备厂商直接基于Linux开发网络路由、防火墙设备。
  + 正如前面提到，Linux得到来自各大软硬件厂商的支持，特别是企业级应用。

Linux的资源丰富，本身的工具和软件已经自带了详细使用文档和大量帮助信息。而且互联网上亦有众多乐于分享和帮助的Linux Fans，如果碰到问题，你甚至可以直接咨询软件的开发者。

## 收集Linux系统不同版本

1. **系统发行版**

Linux发行版指的就是通常所说的“Linux操作系统”，它可能是由一个组织，公司或者个人发布的。Linux主要作为Linux发布版（通常被称为"distro"）的一部分而使用。通常来讲，一个Linux发布版包括Linux内核，将整个软件安装到计算机上的一套安装工具，各种GNU软件，其他的一些自由软件，在一些特定的Linux发布版中也有一些专有软件。发行版为许多不同的目的而制作，包括对不同计算机硬件结构的支持，对一个具体区域或语言的本地化，实时应用，和嵌入式系统。目前，数百个Linux发行版被积极的开发，被广泛应用的发行版有：

* + **Debian**

Debian是完全开放，一个强烈信奉自由软件的系统，由Debian计划组织维护，其背后没有任何营利组织的支持，开发人员完全来自于全世界各地的志愿者。Debian基于Deb的包管理方式，apt的在线软件安装更新非常方便且快速。提供超过18000个软件包的支持，受到研究机构开发人员的极大欢迎。

* + **Ubuntu**

基于Debian开发，采用相同的deb和apt。通过精挑细选，保证软件质量，致力于开发一个简单易用的Linux系统。由Canonical支持，坚持每6个月发布一个版本，分别提供6个月和3年（LTS）的技术支持。由于其易用性和遍布世界各地的镜像源服务器，使得它近年来变得非常流行。

* + **Red Hat Enterprise Linux**

RHEL是Red Hat（红帽）公司的企业版Linux系统，因其稳定强大，各大厂商认证和良好的技术支持，在Linux服务器上市场占领超过50%份额。采用RPM的包管理方式，很多发行版都或多或少的受到它的影响。

* + **CentOS**

由社区开发并维护，基于RHEL，并与RHEL版本号保持一致。致力于提供一个自由使用且稳定的RHEL。开发者直接修改RHEL的源代码，去除了红帽的商标和商业服务组件，修复了很多存在的Bug。其拥有自己的软件仓库，提供免费的在线更新程序。

* + **Fedora**

主要由RedHat主持的社区Linux项目，采用同样的RPM包管理，致力于最新技术的开发和引入。经过测试稳定且有价值的技术将被RHEL吸纳。坚持每半年发布一个版本。

* + **SUSE Linux**

在欧洲非常流行的Linux发行版，以界面华丽和简单易用著称。2004年被NOVELL收购。NOVELL提供企业级的SUSE Linux Enterprise Server｜Desktop软件和商业技术支持服务，企业市场占有率较高。OpenSUSE是基于企业版的社区提供的免费SUSE Linux。

* + **还有一些非常有特色且流行的Linux发行版：**

Arch Linux，一个基于KISS（Keep It Simple and Stupid）的滚动更新的操作系统。

Gentoo，一个面向高级用户的发行版，所有软件的源代码需要自行编译。

Elementary OS：基于Ubuntu，界面酷似Mac OS X。

Linux Mint，从Ubuntu派生并与Ubuntu兼容的系统。

下图 1‑4所示是流行的Linux操作系统的图标。



图 1-4 流行的Linux操作系统

1. **Linux的版本**

Linux系统设计对软硬件的版本要求极高，不同的硬件平台、内核版本、系统版本、软件版本一般不能通用。对于Linux版本的理解，主要是了解以下两个方面：

**Kernel的版本**

Kernel版本即Linux内核版本。Linux内核有以下不同的命名方案。

早期版本：

第一个版本的内核是0.01。其次是0.02,0.03,0.10,0.11,0.12（第一个GPL版本),0.95,0.96,0.97,0.98,0.99及1.0。

旧计划（1.0和2.6版之间)，版本的格式为A.B.C，比如2.6.18。其中A,B,C代表：

A大幅度转变的内核。这是很少发生变化，只有当的代码和核心发生重大变化才会改变。在历史上曾改变两次的内核：1994年的1.0及1996年的2.0。

B是指一些重大修改的内核。在2.x时代，如果x为奇数指开发版本，偶数指稳定版本。

C是指轻微修订的内核。这个数字当有安全补丁,bug修复,新的功能或驱动程序时,便会有变化。

自2.6.0(2003年12月)发布后,人们认识到，更短的发布周期将是有益的。自那时起，版本的格式为A.B.C.D，其中A,B,C,D代表：

A和B是无关紧要的

C是内核的版本

D是安全补丁

自3.0(2011年7月)发布后，版本的格式为3.A.B，其中A,B代表：

A是内核的版本

B是安全补丁

我们可以去到[www.kernel.org](http://www.kernel.org)查看或下载最新的内核版本。

**操作系统的版本**

我们在生产和学习中使用的一般是完整Linux操作系统发行版。每个不同的发行版都有自己的版本号命名规则。以Debian为例：

目前最新的版本是9.x 。其中9表示主版本号，x表示次版本号。

Debian的系统映像的文件名如下：

debian-9.4.0-amd64-DLBD-1.iso

其中amd64表示支持64位CPU平台。

Linux是一种Unix Like操作系统。严格来说，Linux只是一个操作系统内核。大多数Linux发行版是由操作系统内核加上GNU的软件或工具形成完整的操作系统。

# 安装系统

**【任务描述】**

新的项目要求基础环境使用Linux系统，需要在服务器上部署Debian9的系统。并要求在安装部署的过程中，合理选择硬件配置，按需定制安装软件套件。

**【学习目标】**

* 1. 掌握Debian9图形界面安装方法；
  2. 掌握Debian9字符界面最小化安装精简版系统。

## 使用图形界面安装Debian9

### 启动安装程序

要使用Debian 9 Linux DVD或者最小引导介质启动安装程序，请按照以下步骤执行：

1. 断开所有与您安装不需要的外部固件或者 USB 磁盘的连接。

打开您的计算机系统。

在计算机中插入该介质。

关闭计算机并将引导介质留在里面。

打开您的计算机系统。

您需要按具体的按键或者组合键从该介质引导。在大多数计算机中会在开机后很短的时间内在屏幕中出现一个信息。通常类似“Press F12 to select boot device”，但不同的计算机中的具体文字以及您要按的按键有很大不同。查看您计算机或者主板的文档或者从硬件生产商或者零售商处寻求支持。

如果您的计算机不允许您在启动时选择引导设备，您可能需要将您系统的基本输入/输出系统（BIOS）配置为使用该介质引导。

**NOTE**

UEFI和BIOS的引导配置有很大差别。因此安装的系统必须使用安装时所用的同一固件引导。您不能在使用BIOS的系统中安装操作系统后，再在使用UEFI的系统中引导这个安装。

### 安装过程

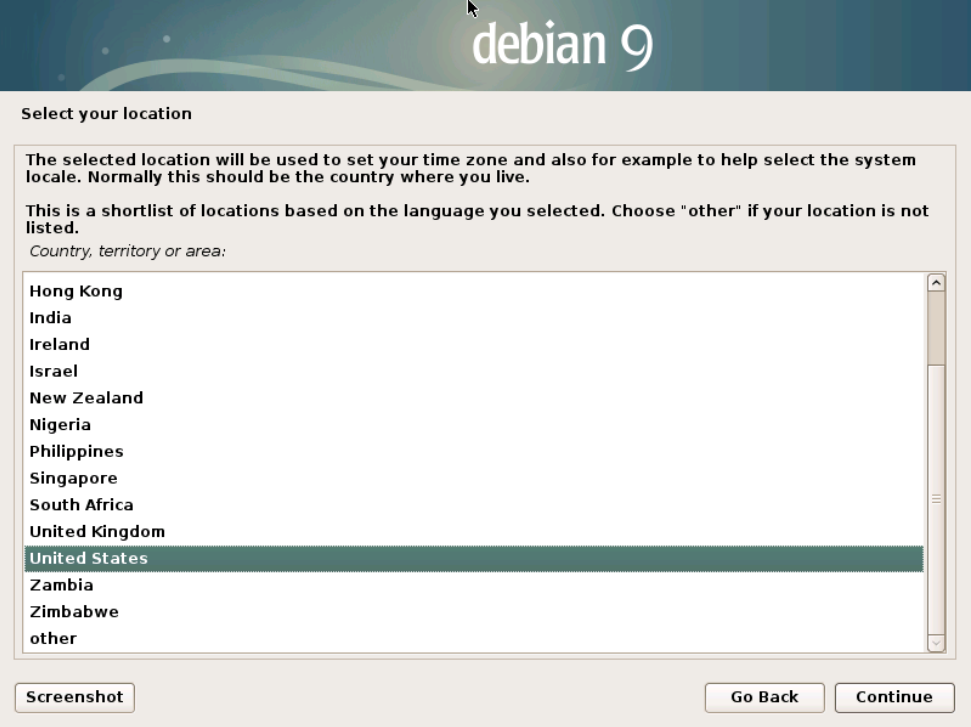
1. 进入安装启动界面。



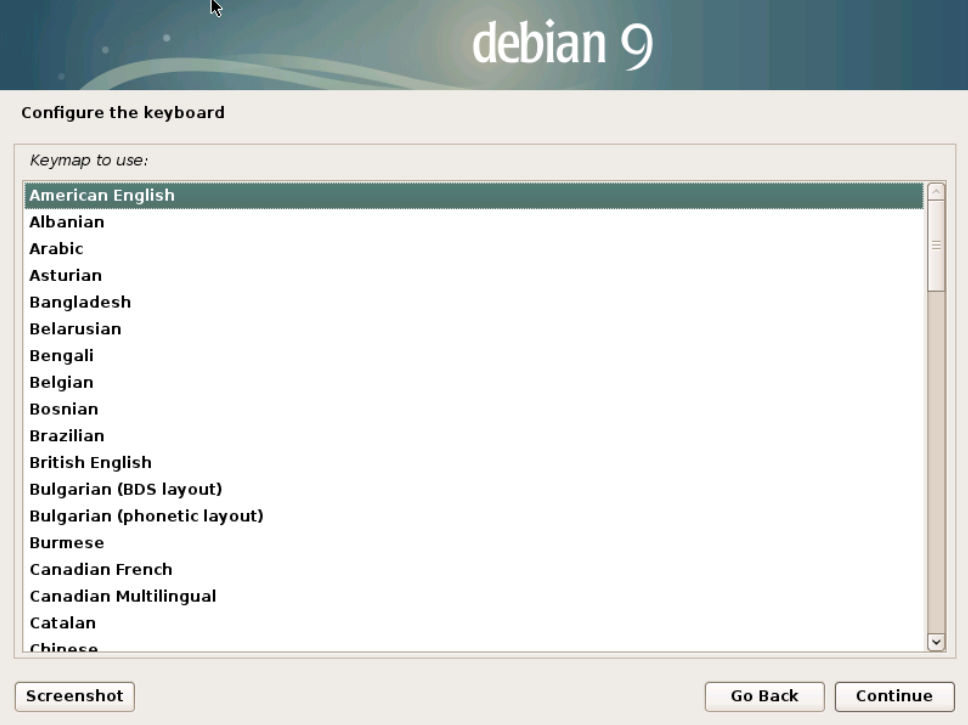
1. 配置语言



1. 配置区域



1. 配置键盘



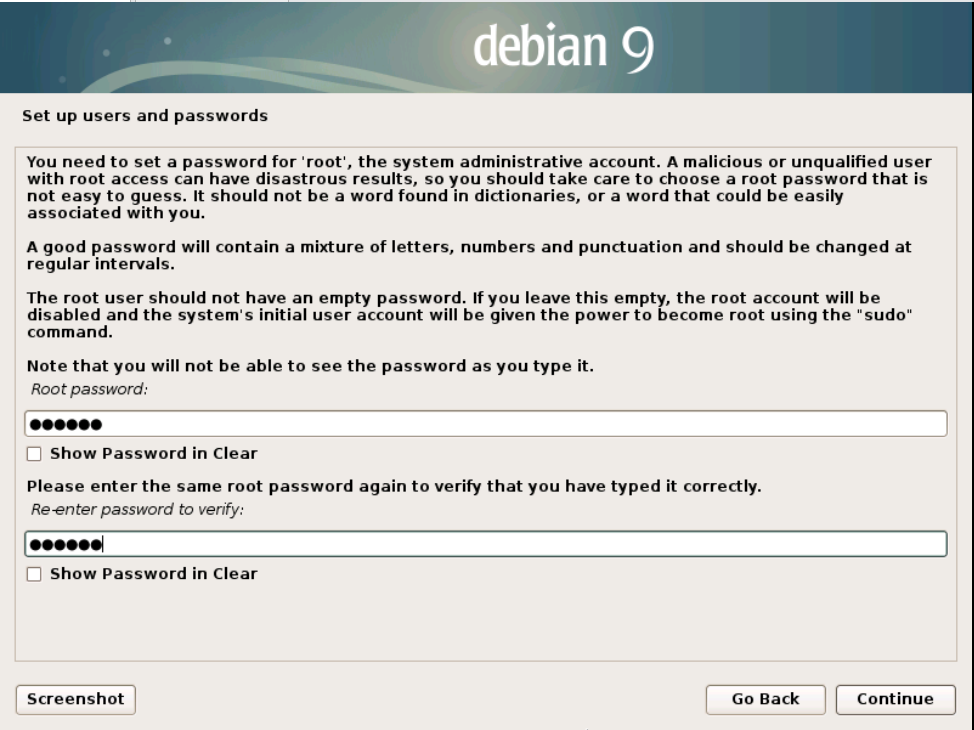
1. 扫描光盘



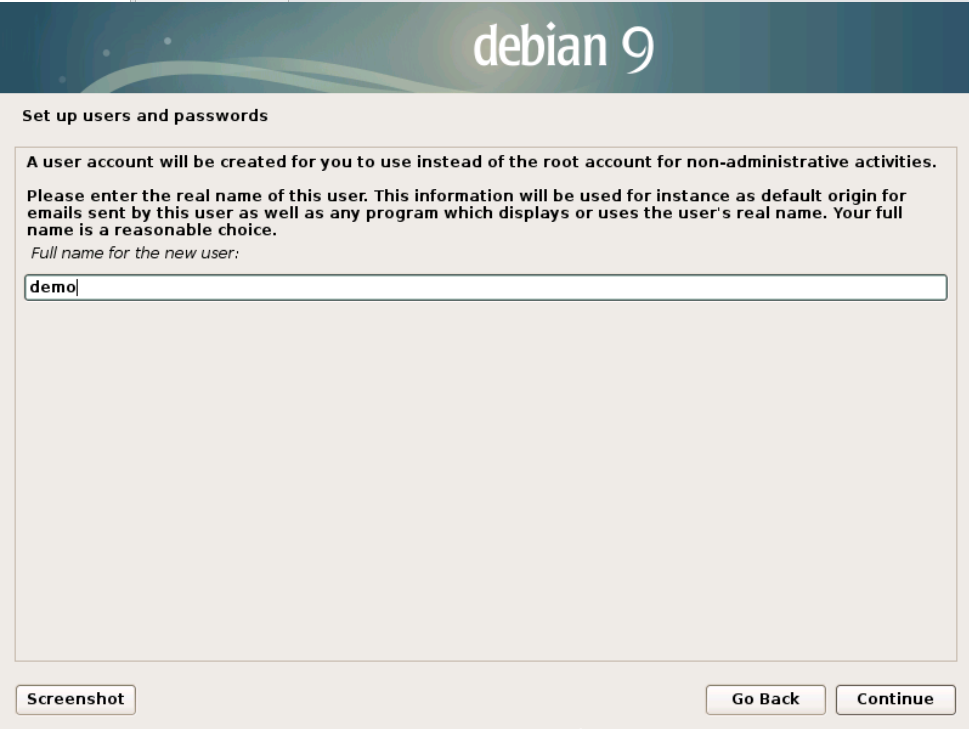
1. 配置网络



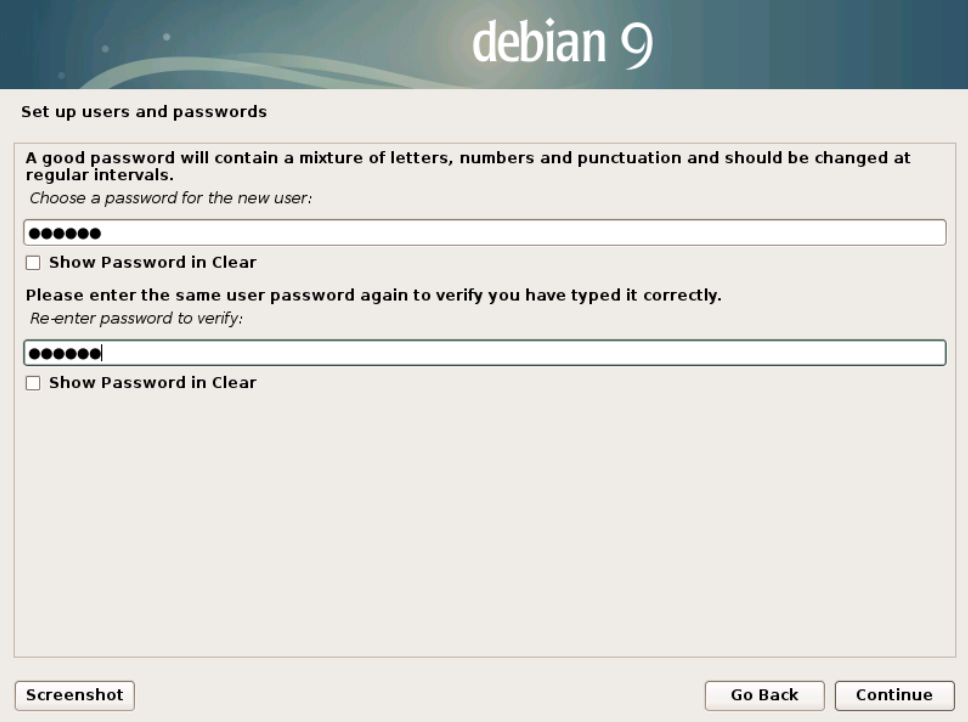
1. 配置管理员密码



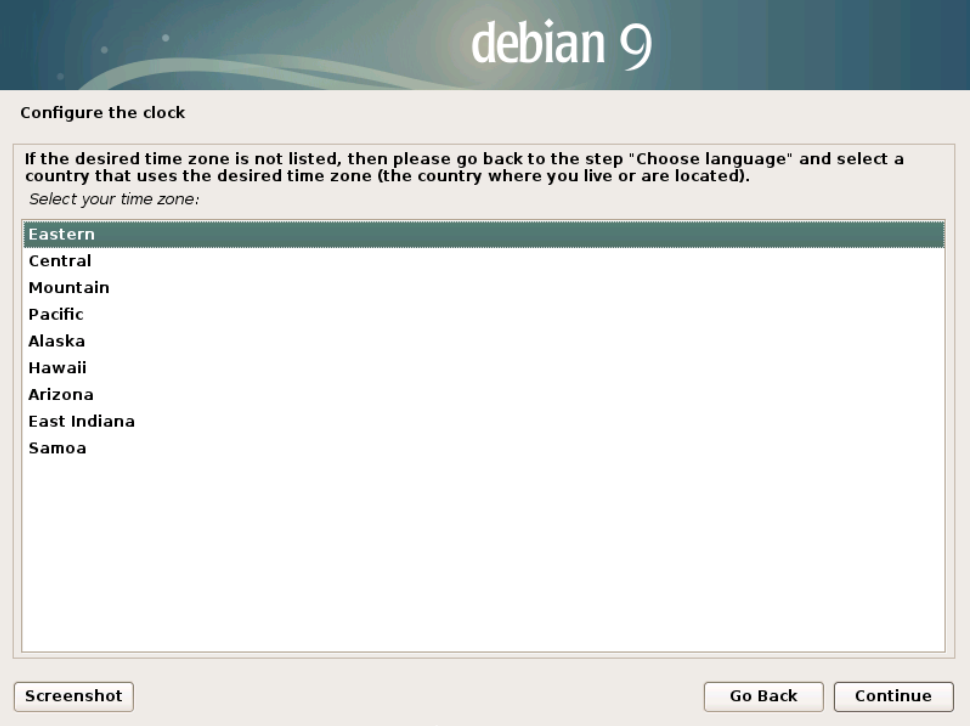
1. 配置普通用户



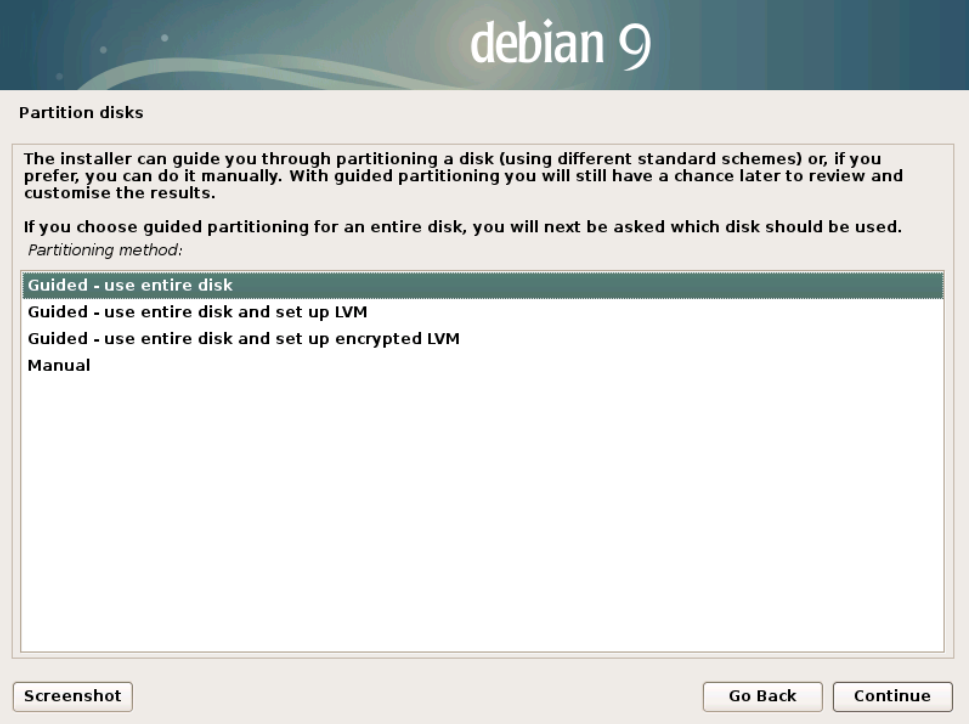
1. 配置普通用户密码



1. 配置时钟

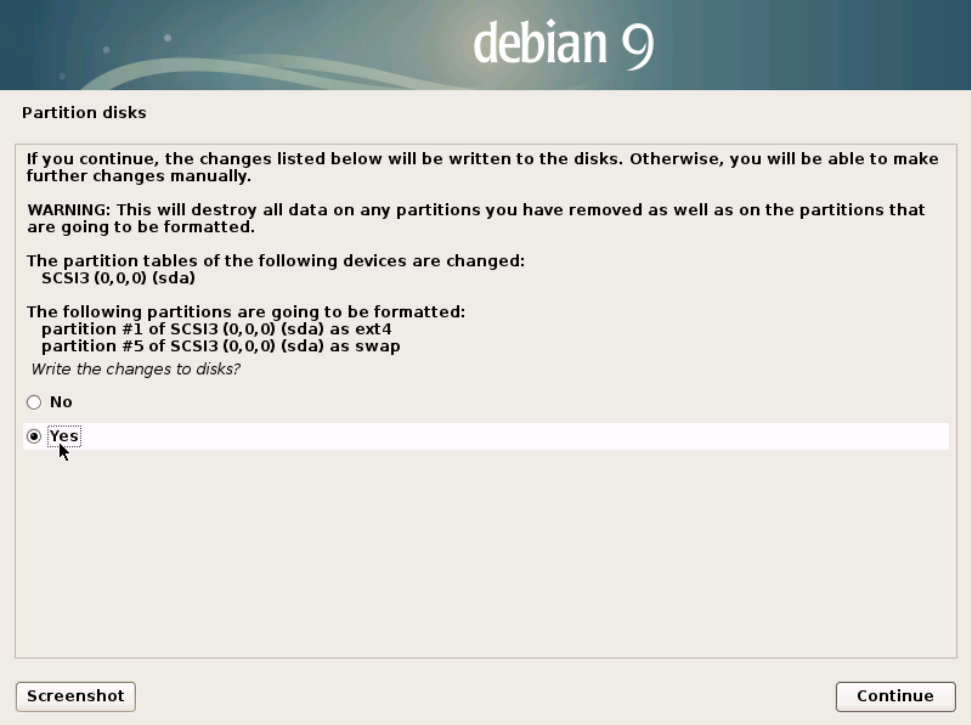


1. 磁盘分区向导

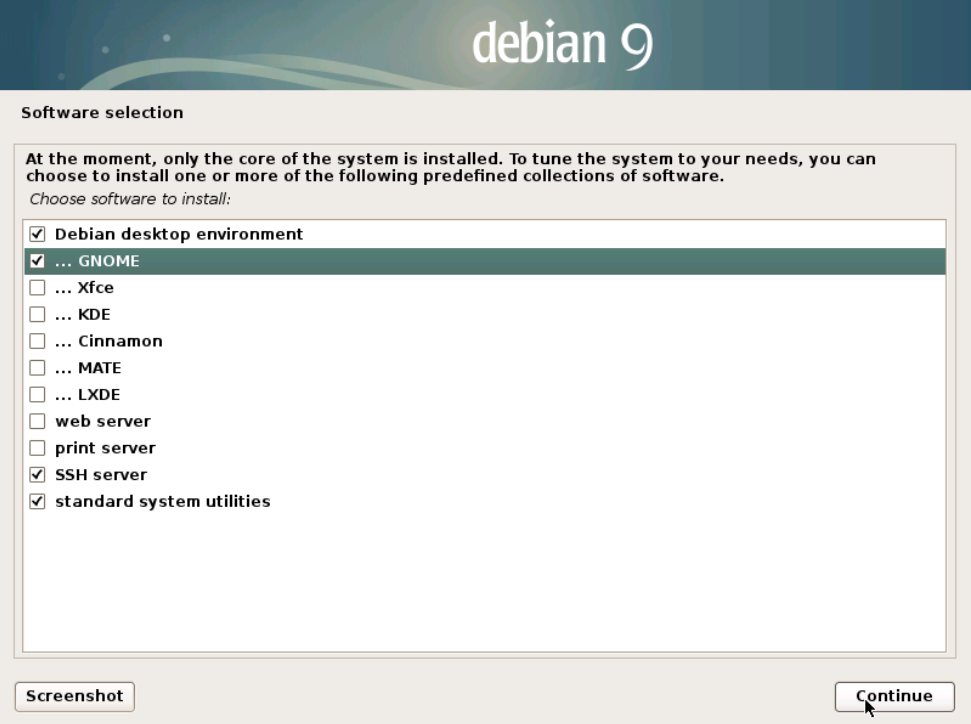








1. 安装软件包选择



1. 开始安装系统



1. 安装系统引导器





1. 安装完成



1. 重启系统



1. 登录系统



安装系统是部署一个业务应用的起点，也是应用架构的基础。所以提前合理的规划，对于安装细节的理解会使用整个工作得心应手，减少以后系统使用中碰到不必要的麻烦。而能应对不同的机房环境，服务器设备，安装介质，安装要求，并能解决安装过程中出现的问题，则需要在以后的学习，对Linux系统体系结构更加深入了解后，通过反复实践和经验的积累进行加强。

随着学习的深入和工作实践，以后在部署Linux系统时，要尽量考虑到安全性，稳定性，可扩展性，最大发挥系统软硬件性能，以及最优的性价比等多方面的因素。

# 命令行技术

**【任务描述】**

在学习和使用Linux系统的过程中，shell命令行是非常重要的组成部分。虽然图形桌面环境如GNOME提供了一个友好的操作界面，但在实践中，shell命令行提供了更多的功能，更好的灵活性，以及自动化和批量处理的能力，可以简化或实现那些使用图形工具难以有效完成的操作。shell环境还提供了在Linux服务器无法使用图形界面交互的可能，例如在系统非正常启动的情况下，或基于字符界面的远程连接管理。

**【学习目标】**

* 1. 理解命令行的格式及用法；
  2. 熟练使用--help和man获得命令的帮助信息；
  3. 理解命令行中工作路径的意义和使用。

## 开始使用shell命令行

Linux（or Unix） shell也叫做命令行界面，它是Linux/Unix操作系统下传统的用户和计算机的交互界面。用户直接输入命令来执行各种各样的任务。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低层的细节。Linux操作系统下的shell既是用户交互的界面，也是控制系统的脚本语言。

在Linux系统中被广泛使用的shell是Bash，在1987年由布莱恩·福克斯（Brian J. Fox）为了GNU计划而编写。1989年发布第一个正式版本，原先是计划用在GNU操作系统上，但能运行于大多数类Unix系统的操作系统之上，包括Linux与Mac OS X v10.4都将它作为默认shell。在Novell NetWare与Android在上也有移植。

除此之外，Shell还有ash、dash、ksh、zsh、csh、tcsh等。

Debian中的默认Shell是Bash。

### 使用shell命令行

默认DEBIAN9提供了7个虚拟控制台，可以使用[ctrl]+alt+F1~F7来进行切换，其中控制台1（或7）是图形界面，其余为命令行控制台，如图 5所示。



图 5 虚拟控制台

在桌面环境下，可以使用虚拟终端来运行命令。GNOME中的默认终端模拟器是gnome-terminal。可以通过如下方式启动：

* + 从顶部面板依次选择单击“应用程序”🡢“系统工具”🡢“终端”打开；
  + 在桌面空白区域或Nautilus窗口空白区域右击，选择“在终端中打开”；
  + 按快捷键alt+F2，在运行应用程序的对话框中输入“gnome-terminal”，回车运行，如图 6所示。

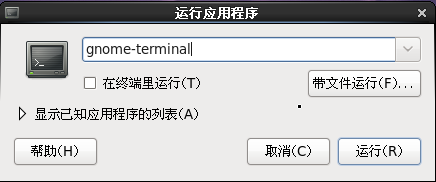


图 6 运行应用程序

通过以上任一种方式打开虚拟终端窗口，如图 7所示。



图 7 虚拟终端窗口

在shell提示符终端输入命令。如图 7所示，标准提示符列出了当前用户的登录名称、计算机的较短主机名、当前目录的名称（用方括号括起来）、后跟$提示符。如果以超级用户（root）身份运行shell（参见下文示例），$将替换为#，以便更明显地说明这是超级用户在工作。

用户可以从命令控制台，虚拟终端，远程客户端登录系统并执行命令。

### 在命令行中启动图形工具

就像其他任何其他程序一样，您可以从命令行启动包含图形界面的程序。例如，您可以在一个图形虚拟终端窗口中的shell提示符下键入firefox的命令启动图形的web浏览器。

**示例：**在命令行中启动firefox浏览器。

1. [demo@365linux ~]$ firefox

但是，这种做法的缺点是只要图形程序仍然在运行，用于启动的shell提示符就会被占用而一直不可用，这种情况称为程序在前台运行。为了避免这种不便，可以在提示符下的命令行末尾处添加一个&，以在后台启动程序。

**示例：**使用后台运行的方式在命令行中启动firefox浏览器。

1. [demo@365linux ~]$ firefox &

Bash还提供了通过shell提示符更改进程的运行方式（前台、后台）：

* 在Bash中可以使用Ctrl+c快捷键终止前台进程；
* 使用Ctrl+z快捷键暂停前台进程并返回shell提示符；
* 在终端中执行jobs命令列出该与该shell相关联的在后台运行或已停止的进程；
* 在终端中使用fg命令可以向前台发送作业；
* 在终端中使用bg命令可以运行后台暂停的进程。

**示例：**进程的前后台切换：

1. 在前台打开gedit文件编辑器。
2. [demo@365linux ~]$ gedit

按Ctrl+z组合键将进程暂停。

1. ^Z
2. [1]+ Stopped gedit

查看该shell终端的后台或已暂停的进程。

1. [demo@365linux ~]$ jobs
2. [1]+ Stopped gedit

使用bg命令将进程在后台恢复运行。

1. [demo@365linux ~]$ bg 1
2. [1]+ gedit &

使用fg命令将进程发送至前台，并使用Ctrl+c结束进程。

1. [demo@365linux ~]$ fg 1
2. gedit
3. ^C

**工作技巧**

通常在shell终端提示符中运行的作业（非服务进程）是和该shell相关联的，当该shell终端被关闭时，运行的作业也会停止，这时可以使用nohup的命令来脱离这种关联性。

如果出于某些原因您需要以root用户身份运行图形程序，而PolicyKit（普通用户特权获取机制管理）不支持以普通用户身份运行该图形程序时在必要时进行root用户的授权（比如Nautilus文件管理器），那么则需要在命令行下切换到root用户，然后在命令行中以root的身份打开该图形程序。

**示例：**以root用户身份打开nautilus文件管理器。

1. [demo@365linux ~]$ su -
2. 密码：
3. [root@365linux ~]# nautilus &
4. [root@365linux ~]# exit

**工作技巧**

在命令后中使用su – 命令切换用户后，想要回到之前的登录用户，要使用exit命令退出当前用户，则退回到之前的用户，而不能使用su – 来回反复切换。

### 命令的格式

在shell提示符下面输入的命令由三部分组成：

**命令 选项 参数**

command [-options] [parameter1] [parameter2] ……

**示例：**执行ls命令，-l –r是短选项，--size是长选项 /boot是命令执行的参数。

1. [demo@365linux ~]$ ls -l --size -r /boot

Bash命令至少有如下特点：

* 一条命令必须是以可执行的命令开头，以空格隔开；选项是可选的，但在大多数情况下被使用以满足用户的功能定制要求；参数一般是命令要操作的对象，一条命令的参数可以是一个或多个，没有参数则取命令参数默认值。输入完命令后按“Enter”键执行。
* “-”后面接简写的选项（字母），可以把多个简写的选项串在一起，不过有时要注意顺序；长选项（由单词或单词缩写组成）用“--”分隔。选项也可以有自己的参数，如“--width=40”。
* 命令中间的空格不论几格，Bash都视为一格，命令太长时用“\”转义“Enter”，使用命令可以换行继续输入。
* Linux命令严格区分大小写。
* 在大部分情况下，对于命令执行的结果，“No news is good news!”。
* Tab键补全命令和路径。
* Bash具有历史命令功能（history）。
* 命令可以使用别名的形式（alias）。
* 有大量的快捷键可以使用；
* 有完善的帮助文档。

以下对Bash的部分重要功能做详细的解释：

**命令行Tab键补全功能**

Tab补全允许您在提示符下键入足够的内容以使其唯一后快速补全命令和文件名。如果键入的字符匹配到的命令或文件名不唯一，则按Tab键两次，以显示所有以键入的字符开头的命令或文件名的情况。Tab补全的功能可以提高命令的输入速度，且可以判断输入的命令或文件名的正确性。

**示例：**使用两次Tab键显示出以pas开头的命令。

1. [demo@365linux ~]$ pas<Tab><Tab>
2. passwd paste pasuspender

**示例：**使用Tab键自动补全命令passwd。

1. [demo@365linux ~]$ pass<Tab>
2. [demo@365linux ~]$ passwd

**示例：**使用Tab键自动补全路径。

1. [demo@365linux ~]$ ls /etc/sysco<Tab>/ipt<Tab>
2. [demo@365linux ~]$ ls /etc/sysconfig/iptables

**Bash历史记录功能**

Shell历史记录允许您查看之前运行过的命令并对其进行编辑或再次执行。使用history命令查看所有之前命令，或者使用上箭头和下箭头一次滚动浏览一个历史记录命令。历史记录命令的输出包含数字值。在感叹号（!）后面使用该数字可以再次运行该命令。在感叹号后面使用非数字值则运行最后一个以这些字符开头的命令。

**示例：**使用history命令显示当前会话所有的历史命令。

1. [demo@365linux ~]$ history
2. 1 su -
3. 2 cd
4. 3 firefox &
5. 4 su -
6. 5 history

**示例：**执行编号为3的历史命令firefox &。

1. [demo@365linux ~]$ !3

**使用命令别名**

对于一些较长的命令执行格式或者命令组合，而又经常使用的，可以使用别名的方式进行定义，减少反复较长的输入。使用alias命令可以显示和定义别名，使用unalias取消命令别名。除非将别名的定义写到用户的配置文件，否则别名只在当前会话中有效。

**示例：**显示当前会话所有别名。

1. [demo@365linux ~]$ alias
2. alias l.='ls -d .\* --color=auto'
3. alias ll='ls -l --color=auto'
4. alias ls='ls --color=auto'
5. alias vi='vim'
6. alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'

**示例：**自定义别名p2。

1. [demo@365linux ~]$ alias p2='ping -c 3 192.168.148.2'
2. [demo@365linux ~]$ p2
3. PING 192.168.148.2 (192.168.148.2) 56(84) bytes of data.
4. 64 bytes from 192.168.148.2: icmp\_seq=1 ttl=128 time=0.112 ms
5. 64 bytes from 192.168.148.2: icmp\_seq=2 ttl=128 time=0.386 ms
6. 64 bytes from 192.168.148.2: icmp\_seq=3 ttl=128 time=0.141 ms
7. --- 192.168.148.2 ping statistics ---
8. 3 packets transmitted, 3 received, 0% packet loss, time 2000ms
9. rtt min/avg/max/mdev = 0.112/0.213/0.386/0.122 ms

**示例：**取消自定义的别名p2。

1. [demo@365linux ~]$ unalias p2
2. [demo@365linux ~]$ p2
3. -bash: p2: command not found

**工作技巧**

用户常用的命令别名定义命令应该写到用户家目录下的.bashrc文件中，以保证每次用户登录都能使用该别名。

**命令的物理位置**

Linux命令分为两种类型：

内部命令：Bash在本身内建的命令，这些命令在shell启动时被加载到内存；

外部命令：内建命令之外的可执行程序，通常是由系统的组件或者应用程序安装提供。

通过which命令可以定位命令在系统中的真实路径。

**示例：**使用which命令定位ls命令的路径。

1. [demo@365linux ~]$ which ls
2. alias ls='ls --color=auto'
3. /bin/ls

**示例：**使用which命令定位history命令，结果显示history并不存在于系统命令路径中，可见history命令是Bash内建的命令。

1. [demo@365linux ~]$ which history
2. /usr/bin/which: no history in (/usr/lib/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin: \
3. /usr/local/sbin:/usr/sbin:/sbin:/home/demo/bin)

**示例：**显示Bash在系统中查找命令路径的环境变量。

1. [demo@365linux ~]$ echo $PATH
2. /usr/lib/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/demo/bin

**Bash命令行的快捷键**

RHEL系统Bash命令行环境有大量的快捷键可以使用，方便操作。这里仅例出常用的快捷键。如下表 0‑1所示：

表 0‑1 Bash快捷键

|  |  |
| --- | --- |
| 快捷键 | 功能 |
| Ctrl + c | 非常规中断，中止前台进程  例如：中断命令 cat /dev/zero |
| Ctrl + d | 输入完成的正常信号       例如：命令 wc 和 at 的操作 |
| Ctrl + z | 挂起前台进程，用fg恢复 用bg恢复后台执行 |
| Ctrl + l | 清屏, 等同命令clear |
| Ctrl + u | 删除当前行 |
| Ctrl + h | 删除光标前一个字符 |
| Ctrl + w | 清除光标之前的字符串 |
| Ctrl + k | 清除光标之后的字符串 |
| Ctrl + a | 光标移动到命令行的行首 |
| Ctrl + e | 光标移动到命令行的行尾 |
| Ctrl + r | 从历史命令中找含有键入字符的命令 |
| Alt + b | 光标往前移动一个字符串 |
| Alt + f | 光标往后移动一个字符串 |
| Ctrl + y | 恢复Ctrl＋w或Ctrl＋k 清除的内容 |
| Ctrl + B | 光标往前移动一个字符 |
| Ctrl + F | 光标往后移动一个字符 |
| Ctrl + X | 在光标所在位置与行首切换 |
| Alt+ . | 补全之前输入过的参数 |
| Alt + U | 换成大写 |
| Ctrl + S | 锁住终端输出 |
| Ctrl + Q | 解锁终端 |

**工作技巧**

在某些情况下（比如使用cat命令查看了不能直接打开的文件），会导致终端显示乱码并无法正常输入，则可以盲打reset命令进行终端复原。

### 获得命令的帮助信息

只了解命令单一的作用是不够的。为了有效地使用命令，还需要了解每个命令可以接受哪些选项和参数，以及命令希望如何排列这些选项和参数（命令的语法）。一个完整的系统里包含了数千个命令，而每个命令都有自己众多不同的选项和参数，这样庞大繁多的命令用法极大的体现了Linux命令行的灵活性，同时也增加了使用者的学习和应用难度。

然而几乎所有的命令和配置文件的语法格式和用法都有帮助文档。我们可以通过以下几种方式获得帮中信息：

* 使用命令 -h (--help)或者 -?的选项来获得命令使用的规范和选项、参数的信息；
* 使用man来获得命令的使用手册；
* 使用pinfo读取文档；
* 查看/usr/share/doc中的文档。

**--help帮助输出**

大多数命令都有-h（有的命令-h有其他特定的功能，则只能使用--help）的帮助选项，执行命令的该选项时会在终端输出简洁的帮助信息。

**示例：**使用--help选项获取ls命令的帮助信息。

1. [demo@365linux ~]$ ls --help
2. Usage: ls [OPTION]... [FILE]...
3. List information about the FILEs (the current directory by default).
4. Sort entries alphabetically if none of -cftuvSUX nor --sort.
5. Mandatory arguments to long options are mandatory for short options too.
6. -a, --all do not ignore entries starting with .
7. ……

关于帮助输出的几个基本惯例：

* 方括号（[ ]）中的任何内容都为可选；
* 省略号（…）表示此字符串可以任意长度列表；
* 以竖线（|）分隔的多个选项，这表示可以选择其中任意一项；
* 尖括号（< >）中的文本表示必须出现变量数据。因此<filename>表示“在此插入要使用的文件名”。

**使用man读取帮助文档**

Linux man手册提供了比help输出更为详尽的帮助文档，类似于一本分成许多章节的大型书籍。用户可以通过在终端执行man命令来相关命令或文件的帮助文档信息。终端以每次一个屏幕的形式显示内容，并可通过键盘命令来控制导航man手册。man导航按键如下表 0‑2所示。

表 0‑2导航man手册

|  |  |
| --- | --- |
| 按键/命令 | 结果 |
| 空格键 | 向前滚动一个屏幕 |
| 方向键下 或J | 向前滚动一行 |
| 方向键上 或k | 向后滚动一后 |
| /string | 在man手册中向前搜索string |
| n | 在man手册中重复之前的向前搜索 |
| N | 在man手册中重复之前的向后搜索 |
| q | 退出man并返回到终端提示 |

**示例：**查看ls命令的man手册。

1. [demo@365linux ~]$ man ls
2. LS(1) User Commands LS(1)
3. NAME
4. ls - list directory contents
5. SYNOPSIS
6. ls [OPTION]... [FILE]...
7. DESCRIPTION
8. ……

**man文件主要包括以下几个部分（各命令可能有区别）：**

1. NAME 程序名和简介
2. SYNOPSIS 命令的格式，显示所有的选项和参数
3. DESCRIPTION 命令功能的描述和选项的详解
4. OPTIONS 所有选项清单和描述
5. EXAMPLES 用法举例
6. AUTHOR 作者
7. REPORTING BUGS 报告BUG
8. CCOPYRIGHT 许可证
9. SEE ALSO 相关内容

man手册分为多个章节，所以可能会出现相同名称的多个帮助文档内容。为了区人不同的man手册，在编写对man手册的引用时，通常在man手册的名称后面添加手册的章节号并用括号括起来。比如，用户命令passwd的帮助文档是passwd（1），存储本地用户信息的配置文件/etc/passwd的帮助文档是passwd(5)。

关于man的章节，可以通过man man来查看相关信息，如下表 0‑3所示：

表 0‑3 man手册的章节

|  |  |
| --- | --- |
| 章节 | man手册类型 |
| 1 | 用户命令 |
| 2 | 内核系统调用（从用户空间到内核的进入点） |
| 3 | 库函数 |
| 4 | 特殊文件和设备 |
| 5 | 文件格式和规范 |
| 6 | 游戏 |
| 7 | 规范、标准和其他页面 |
| 8 | 系统管理命令 |
| 9 | Linux内核API(内核调用) |

man命令会按特定顺序搜索手册中的各节，并显示找的第一个匹配项；例如man passwd默认情况下将显示passwd(1)。要找到特定节中的man手册，则必须在命令行中以参数的形式指定节号：man 5 passwd 将显示passwd(5)。

**示例：**查看/etc/passwd配置文件的帮助文档。

1. [demo@365linux ~]$ man 5 passwd

可以使用man -k keyword 对man手册执行关键字搜索，这将产生一个相关man手册的列表，包括对应的章节。

1. [demo@365linux ~]$ man -k passwd
2. chpasswd (8) - update passwords in batch mode
3. fgetpwent\_r [getpwent\_r] (3) - get passwd file entry reentrantly
4. getpwent\_r (3) - get passwd file entry reentrantly
5. gpasswd (1) - administer /etc/group and /etc/gshadow
6. htpasswd (1) - Manage user files for basic authentication
7. kpasswd (1) - change a user's Kerberos password
8. lpasswd (1) - Change group or user password
9. lppasswd (1) - add, change, or delete digest passwords
10. pam\_localuser (8) - require users to be listed in /etc/passwd
11. pam\_passwdqc (8) - Password quality-control PAM module
12. passwd (1) - update user's authentication tokens
13. passwd2des [xcrypt] (3) - RFS password encryption
14. passwd (5) - password file
15. ……

**NOTE**

关键字搜索需要makewhatis更新数据库，通常系统每天会自动运行更新。

**使用pinfo读取文档**

GNU Project开发的软件使用info系统来提供其部分文档，info文档通常以书籍的形式提供，其由包含超链接的info节点组成。此格式比man手册更加灵活，允许对复杂命令和概念进行更加彻底的说明。在某些情况下，某个命令同时存在相应的man手册和info文档；大多数情况下，info文档的信息更加详细。

**示例：**比较tar的man手册和info文档。

1. [demo@365linux ~]$ man tar
2. [demo@365linux ~]$ pinfo tar

**/usr/share/doc中的文档**

对于在man手册，info手册，或者GNOME帮助实用程序都不能找到相关帮助文档，则可以在系统目录/usr/share/doc中查找。许多应用程序和系统命令的帮助文档位于该目录下以RPM软件包命令的子目录中。

**示例：**查看mdadm.conf配置文件的示例文件。

1. [demo@365linux ~]$ less /usr/share/doc/mdadm-3.2.6/mdadm.conf-example

## 使用命令行完成系统任务

### 从命令行管理文件

Linux文件系统具有层次结构，其组织方式采用“倒树”模型。顶级目录称为根目录（/目录），是整文件系统层次结构的起点，而根分区挂载到/目录。要在系统中指定文件的位置，可以指定该文件的绝对路径（从根目录到各级子目录到文件），或者使用相对路径（从当前工作目录到其下的各级子目录到文件）。

在命令行中文件的路径，如：/usr/share/doc，位于最前面的/表示根目录，即绝对路径的起点，之后的/则表示路径中目录的分隔符。

**NOTE**

在某些系统或说法中，经常将根目录（/）称作文件系统层次结构的root（这里的root表示根的意思），而系统中存在的/root目录是管理员用户root的家目录，容易造成混淆，一定要理解清楚。

**切换工作路径**

**示例：**使用pwd命令查看当前的工作目录。

1. [demo@365linux ~]$ pwd
2. /home/demo

**示例：**使用cd命令切换工作目录。

1. 使用绝对路径方式进入doc目录。在命令行中，绝对路径作为参数一定是从根目录（/）开始，依次连接各级子目录。切换到目标目录后，终端提示符会改变成为当前目录的简写。
2. [demo@365linux ~]$ cd /usr/share/doc/
3. [demo@365linux doc]$

使用相对路径方式进入当前doc目录下的zip-3.0目录。在命令行中，使用相对目录，即相对于当前的工作目录，使用相对目录时，要省略目录前的路径分隔符，否则会和绝对路径产生混淆。示例中的的zip-3.0等同于/usr/share/doc/zip-3.0。

1. [demo@365linux doc]$ cd zip-3.0/
2. [demo@365linux zip-3.0]$

返回上一级目录，参数“..”表示上一级目录；“.”表示当前目录。

1. [demo@365linux zip-3.0]$ cd ..
2. [demo@365linux doc]$

快速返回当前用户的家目录，参数“~”表示当前用户的家目录。“~zhangsan”则表示用户zhangsan的家目录。

1. [demo@365linux doc]$ cd ~
2. [demo@365linux ~]$ pwd
3. /home/demo

快速进入上一次工作目录。参数“-”表示切换到当前目录之前的目录。

1. [demo@365linux ~]$ cd -
2. /usr/share/doc
3. [demo@365linux doc]$

**查看目录文件列表**

在Linux系统中，一个基本原则是“一切皆文件”，包括硬件设备。这样，通过简单工具即可完成某些功能非常强大的操作。根据文件的特点，Linux系统将文件分为七种类型：

* + -：一般文件；
  + d：目录；
  + l：链接文件；
  + b：块设备文件；
  + c：字符设备文件；
  + s：套接字文件；
  + p：管道文件

符号或字母是在命令行中的标示符。

**示例：**用ls命令查看文件列表并显示文件属性（包括类型）。

1. [demo@365linux ~]$ ls -l /boot
2. 总用量 26149
3. -rw-r--r--. 1 root root 109953 11月 11 11:26 config-2.6.32-431.el6.i686
4. drwxr-xr-x. 3 root root 1024 3月 5 03:02 efi

ls –l 命令产生的效果等同于ll命令，ll是该命令用法别名。ll命令列出的文件属性包含七个字段，分别是，文件类型及文件权限、连接数、拥有者、所属组、文件大小、文件最近修改时间、文件名。

**示例：**在Bash命令中使用通配符“\*”来匹配目录或文件名的引用。

1. [demo@365linux ~]$ ls /etc/a\*.conf
2. /etc/asound.conf /etc/autofs\_ldap\_auth.conf

**查找系统文件**

**示例：**使用find命令查找系统文件。

1. [demo@365linux ~]$ find /home/demo -name \*bash\*
2. /home/demo/.bash\_history
3. /home/demo/.bash\_logout
4. /home/demo/.bashrc
5. /home/demo/.bash\_profile

**文件的基本操作**

**示例：**对于目录和文件的基本操作命令。

1. 在用户的家目录创建一个新的文件夹test03。
2. [demo@365linux ~]$ mkdir test03

进入刚刚创建的新目录test03，创建一个空文件hello.txt。

1. [demo@365linux ~]$ cd test03/
2. [demo@365linux test03]$ touch hello.txt

查看文件的类型。

1. [demo@365linux test03]$ ll hello.txt
2. -rw-rw-r--. 1 demo demo 0 3月 17 18:13 hello.txt

建立hello.txt的软链接文件。

1. [demo@365linux test03]$ ln -s hello.txt ln\_hello.txt
2. [demo@365linux test03]$ ll
3. 总用量 0
4. -rw-rw-r--. 1 demo demo 0 3月 18 08:34 hello.txt
5. lrwxrwxrwx. 1 demo demo 9 3月 18 08:35 ln\_hello.txt -> hello.txt

在Linux系统中，软链接文件即指向目标文件的快捷方式，但源文件被删除时，软链接则成为一个失效的文件。除了软链接文件，Linux系统还支持硬链接文件（同样使用ln命令创建，不使用-s选项）。

删除该文件和链接文件。

1. [demo@365linux test03]$ rm hello.txt
2. [demo@365linux test03]$ ll
3. 总用量 0
4. lrwxrwxrwx. 1 demo demo 9 3月 18 08:35 ln\_hello.txt -> hello.txt
5. [demo@365linux test03]$ rm ln\_hello.txt
6. [demo@365linux test03]$ ls

复制一个文件到当前工作目录。

1. [demo@365linux test03]$ cp /etc/man.config ./
2. [demo@365linux test03]$ ls
3. man.config

创建一个新的目录test04，并将当前目录中的man.config移动到test04目录中。

1. [demo@365linux test03]$ mkdir test04
2. [demo@365linux test03]$ mv man.config test04/
3. [demo@365linux test03]$ ls
4. test04
5. [demo@365linux test03]$ ls test04/
6. man.config

进入test04目录，查看当前绝对路径。

1. [demo@365linux test03]$ cd test04/
2. [demo@365linux test04]$ pwd
3. /home/demo/test03/test04

返回上一级目录，删除含有文件的目录test04。

1. [demo@365linux test04]$ cd ..
2. [demo@365linux test03]$ rm -rf test04
3. [demo@365linux test03]$ ls

回到上一级目录，并删除空目录test03。

1. [demo@365linux test03]$ cd ..
2. [demo@365linux ~]$ rmdir test03/

**工作技巧**

rm –rf命令会强制删除一切它的目标目录下的所有内容，所以要谨慎使用，特别是root用户使用时。

**NOTE**

关于Linux文件系统、文件类型、权限等概念会在后面的章节中提及。

### 设置系统时间和时区

**示例：**使用date命令查看当前的时间。

1. [demo@365linux test03]$ date
2. 2014年 03月 18日 星期二 08:42:51 CST

**示例：**使用date命令设置时间。

1. [root@365linux ~]# date -s "2014/03/18 08:45:00"
2. 2014年 03月 18日 星期二 08:45:00 CST

**示例：**修改系统的时区。

1. [root@365linux ~]# ln -sf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

另外，建议同时修改配置文件/etc/sysconfig/clock的内容为ZONE="Asia/Shanghai"。

**工作技巧**

使用date命令查看和修改的是系统时间。而在系统内部还有另外一个时间概念，硬件时钟，Linux系统重启后时间会与硬件时钟保持同步，所以修改系统时间，同时也要修改硬件时钟。在命令行中使用hwclock命令查看和设置硬件时钟。

### 退出、关闭系统

**示例：**使用poweroff命令关闭Linux系统。

1. [root@365linux ~]# poweroff

在Linux系统中，有多种方式实现关机或重启。可以使用poweroff、shutdown、halt、init命令关机。使用shutdown、halt、init、reboot可以实现系统重启。关机命令之间存在互相调用，且对于关闭系统和关闭电源在不同的系统版本上存在差异。一般情况下，建议使用poweroff关机，使用reboot重启。

## Bash命令行重要的高阶功能

### 管道

在Linux系统中，管道允许用户将标准输出信息从程序连接至一个程序的输入，这样可以将多个程序（命令）连接成一个管道，后一个程序的作用对象即为前一个程序的输出结果。

RHEL系统使用竖线“|”连接程序管道操作。

**示例：**ls命令列出/usr/lib目录下的所有文件，其结果并不直接显示到屏幕输出，而是通过管道发送到下一个命令grep，通过grep命令过滤出结果中文件名含有jpeg的文件。

1. [demo@365linux ~]$ ls /usr/lib/ |grep "jpeg"

**示例：**用find命令找到/var目录中大小超过1M的文件，将结果用grep过滤出文件路径中包含cache的文件，再将文件列出结果交给wc统计行数，最终得到/var/目录中大小超过1M且与cache有关的文件有多少个。

1. [root@365linux ~]# find /var -size +1M |grep "cache" |wc -l

### I/O重定向

Linux命令行I/O重定向允许用户将标准输出或错误输出从程序发送至文件，以进行保存或屏蔽在终端的输出显示。重定向还支持反过来将文件内容读取至命令行程序。关于I/O重定向的定义如表 0‑4所示。

表 0‑4标准输入输出

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 编号 | 默认 |
| STDIN | 标准输入 | 0 | 键盘 |
| STDOUT | 标准输出 | 1 | 终端 |
| STDERR | 标准错误 | 2 | 终端 |

**示例：**将date命令的标准输出重定向到文件，操作会覆盖文件原来的内容。

1. [demo@365linux ~]$ date > file
2. [demo@365linux ~]$ cat file
3. 2014年 03月 19日 星期三 12:21:43 CST
4. [demo@365linux ~]$ date > file
5. [demo@365linux ~]$ cat file
6. 2014年 03月 19日 星期三 12:21:56 CST

**示例：**使用重定向功能合并文件。

1. [demo@365linux ~]$ echo "1" > file1
2. [demo@365linux ~]$ echo "2" > file2
3. [demo@365linux ~]$ cat file1 file2 > file3
4. [demo@365linux ~]$ cat file3
5. 1
6. 2

**示例：**使用追加模式（不覆盖）将标准输出重定向到文件。

1. [demo@365linux ~]$ date >> test.txt
2. [demo@365linux ~]$ date >> test.txt
3. [demo@365linux ~]$ ls >> test.txt
4. [demo@365linux ~]$ cat test.txt
5. 2014年 03月 19日 星期三 12:27:00 CST
6. 2014年 03月 19日 星期三 12:27:07 CST
7. a
8. b
9. c
10. ……

**示例：**将标准错误输出重定向到文件（覆盖）。

1. [demo@365linux ~]$ ls /boot /root 2> file

**示例：**将标准错误输出重定向到文件（追加）。

1. [demo@365linux ~]$ ls /boot /root 2>> file

**示例：**将标准错误输出重定向到设备文件/dev/null，作用是丢弃错误。

1. [demo@365linux ~]$ ls /xyzabc 2> /dev/null

**示例：**将标准输出和标准错误输出组合，重定向到文件。

1. [demo@365linux ~]$ ls /boot /xyzabc >file 2>&1
2. 或者：
3. [demo@365linux ~]$ ls /boot /xyzabc &> file

**示例：**使用标准输入做为cat命令的处理对象（效果等同于使用参数方式）。

1. [demo@365linux ~]$ cat < file

**示例：** 将文件file的内容读取出来，重定向到test文件。

1. [demo@365linux ~]$ cat > test < file

**示例：** 将键盘输入的内容重定向到file03文件，直到用户输入EOF结束输入。

1. [demo@365linux ~]$ cat > file03 << EOF
2. > 1234
3. > abcd
4. > EOF
5. [demo@365linux ~]$ cat file03
6. 1234
7. abcd

### 正则表达式

正则表达式是用来搜索和匹配文本模式的特殊字符串，依赖于特定的Linux命令行工具工作，比如less、man、vim、locate、grep、sed、awk等。多用于脚本中批量处理。

**示例：**匹配文件中以b开头的行，这里的“^”是正则表达式，表示行首。

1. [demo@365linux ~]$ grep '^b' /etc/passwd

**NOTE**

正则表达式将在shell脚本编程中详细介绍。

### 编写Bash脚本

Linux Bash除了能够在终端中进行交互执行命令外，还可以通过编写shell脚本完成自动化、批处理的任务。Shell脚本实际上就是按语句序列执行的命令组合的文本文件，当然，在Bash提供了很多的编程结构，比如判断，循环，函数等。

**示例：**查看本网段哪些IP可以ping通。

1. [demo@365linux ~]$ vim ping.sh
2. #!/bin/bash
3. #Name : ping.sh
4. #Last revision date : 2014-01-07
5. #usage : ./ping.sh
6. #Evironment
7. Net=“192.168.1”
8. #script start
9. for i in {1..254}
10. do
11. (
12. ping -c 1 $Net.$i &>/dev/null
13. if [ $? -eq 0 ];then
14. echo "$i is up."
15. fi
16. )&
17. #wait
18. done

Linux具有强大的命令行功能，命令行具有简洁高效，功能全面，传输数据量小等特点，在系统管理中发挥巨大的作用。即使操作系统图形界面日益成熟，对于服务器系统而言，命令行功能一直被保留下来，并得到加强。

对于Linux命令行的学习，不能死记硬背命令的语法或用法示例，而要掌握Linux命令的书写规范，理解每条命令背后的逻辑、原理和命令执行后产生的效果。至于命令的用法，则要通过帮助文档，手册获得其相关的选项，参数的意义。对于常用的命令，则能熟能生巧，举一反三。

对于初学者而言，命令的使用往往成为入门的门槛，觉得生涩难懂而又难以记忆。其实不然，对于命令的学习，不要跳脱应用而单一的对于命令选项，参数，用法的去练习、记忆，而应该从实用的角度，当在具体的系统管理操作时，引入命令的学习。比如，我们知道了在图形见面下的文件管理、时间配置、退出登录关闭系统，那么在命令行下如何控制实现这些操作呢。比如，以后在用户管理中，使用命令行如何添加、删除用户等。自然而然完成了对于命令的学习。学习Linux系统管理不是学习命令，命令只是基本单元，只是工具，利用系统命令完成系统管理的任务才是目的。

# VI编辑器

**【任务描述】**

编辑器是编写或修改文本文件的重要工具之一，在各种操作系统中，编辑器都是不可缺少的部件。Linux操作系统中，系统和应用的配置大多需要通过修改配置文件来进行设置，所以编辑器的使用频率高，更显得重要。熟练掌握Linux编辑器的用法，可以极大的提高工作效率。为方便各种用户在各个不同的环境中使用，提供一系列的编辑器，如vi，emacs，pico，nano，gedit等。每种编辑器都有各自的特性，在不同的工作环境中根据自己的需求加以选择。

vim（vi improved）是一种强大的文本编辑器，支持复杂的文本操作。相对图形界面的gedit编辑器，vim可以很方便的在命令行中使用，而且在任何Linux系统中始终可用。

vim是vi的高级版本，提供更多的功能，比如自动格式，语法高亮等。当系统中vim无法使用时，依然可以使用vi命令代替，用法相同。

**工作技巧**在大多数采用最小化安装的Linux系统上，vim默认是不被安装的，但有vi可以使用，你也可以额外安装vim编辑器。

**【学习目标】**

* 1. 使用vim高效的编辑文本文件；
  2. vim三种模式的切换和使用。

## 在命令行中使用vim编辑器

在命令行终端中使用vim命令启动vim编辑器。

**示例：**使用vim打开filename.txt文件，如果文件不存在，则保存后创建一个新文件。

1. [demo@365linux ~]$ vim filename

## vim的三种模式

**命令模式**

打开vim编辑器，即进入命令模式（也称一般模式）。通过键盘命令，对文档进行复制，粘贴，删除，替换，移动光标，继续查找等。该模式也是编辑模式和末行模式的切换的中间模式，可以通过Esc键返回到命令模式。

**编辑模式**

也称插入模式，用于对文档进行添加、删除、修改等操作，在编辑模式中，所有的键盘操作（除了退出编辑模式键）都是输入或删除的操作，所以在编辑模式下没有可用的键盘命令操作。

**末行模式**

进入末行模式，光标移动到屏幕的底部，输入内置的指令，可执行相关的操作，如文件的保存，退出，定位光标，查找，替换，设置行标等。

## 三种模式之间的切换

三种模式之间的切换如下图 8所示：



图 8 vim三种模式切换

三种模式切换按键详细说明如下：

1. 进入编辑模式的命令如表 0‑1所示：

表 0‑1 进入编辑模式的命令

|  |  |
| --- | --- |
| 按键 | 功能 |
| i | 从光标所在位置前面开始插入文本，同insert键； |
| I | 从光标所在行的行首开始插入文本； |
| a | 从光标所在位置后面开始插入文本； |
| A | 从光标所在行的行尾开始插入文本； |
| o | 在光标所在行下方新增一行插入文本； |
| O | 在光标所在行上方新增一行插入文本； |
| s | 删除光标所在字符并开始插入文本； |
| S | 删除光标所在行并开始插入文本。 |

1. 进入末行模式的命令如表 0‑2所示：

表 0‑2 进入末行模式的命令

|  |  |
| --- | --- |
| 按键 | 功能 |
| ： | 在后面接要执行的命令； |
| / | 在后面接要搜索的字符串，从光标位置开始向下搜索，按n重复前一个搜索动作，按N反向重复前一个搜索动作； |
| ? | 在后面接要搜索的字符串，从光标位置开始向上搜索，按n重复前一个搜索动作，按N反向重复前一个搜索动作。 |

**NOTE**

不能从编辑模式直接进入末行模式，也不能从末行模式进入编辑模式，两者之间的转换必须先按Esc键退出到命令模式，然后再通过相关的命令按键切换。

## 退出vim编辑器

退出vim需要进入末后模式执行退出的命令。在命令模式下，按“：”键进入末后模式，在末行模式下输入相关的命令，如表 0‑3所示。

表 0‑3 退出vim编辑器的命令

|  |  |
| --- | --- |
| 命令 | 功能 |
| q | 没有对文档做过修改，退出； |
| q! | 对文档做过修改，强制不保存退出； |
| wq或x | 保存退出；可以添加！表示强制保存退出； |
| ZZ | 若文档没有修改，则不保存退出；若文档已经修改，则保存后退出。 |

## vim命令模式下的常用操作

在vim编辑器命令模式下，有着大量方便快捷的键盘命令，用来控制光标，操作文本。使用vim的键盘命令，可以使用户的双手不离开主键盘区域，不使用鼠标，实现光标移动、复制、粘贴、删除等操作，熟练使用，极大的提高工作效率。

以下列出命令模式下常用的操作命令：

**光标移动**

常用光标移动命令见表 0‑4。

表 0‑4 光标移动命令

|  |  |
| --- | --- |
| 命令（n表示数字） | 功能 |
| h/j/k/l | 光标向左/下/上/右移动一个字符 |
| nj | 向下移动n行（可以是nh/nk/nl） |
| Ctrl+f/b/d/u | 屏幕向下/上/移动一页（半页） |
| n<space键> | 光标向后移动n字符 |
| n<enter键> | 光标向下移动n行 |
| H/M/L | 光标移动到屏幕上方/中央/下方 |
| +/- | 光标移动到非空格符的下/上一行 |
| 0或者^ | 光标移动到行首 |
| $ | 光标移动到行尾 |
| gg | 光标移动到文件第一行 |
| G | 光标移动到文件最后一行 |
| nG | 光标移动到文件的第n行 |

**删除、复制与粘贴**

常用的删除、复制与粘贴的命令见表 0‑5。

表 0‑5 删除、复制与粘贴的命令

|  |  |
| --- | --- |
| 命令 | 功能 |
| x/X/nx | 向后/前删除一（n）个字符 |
| dd/ndd | 删除光标所在的行/向下删除n行 |
| d1G/dgg | 删除光标位置到第一行所有数据 |
| dＧ | 删除光标位置到最后一行所有数据 |
| d0/d$ | 删除光标位置到该行行首/尾 |
| cw/ncw | 更改光标位置的一（n）个字符串 |
| yy/nyy | 复制光标所在一（向下n）行 |
| y1G/ygg | 复制光标位置到第1行所有行 |
| yG | 复制光标位置到最后一行数据 |
| y0/y$ | 复制光标位置到该行行首/尾 |
| yw/nyw | 复制光标位置1（n）个字符串 |
| p/P | 粘贴到光标位置下/上一行 |

**替换**

替换命令见表 0‑6。

表 0‑6 替换的命令

|  |  |
| --- | --- |
| 命令 | 功能 |
| r | 仅替换一次光标所在的字符 |
| R | 一直替换光标所在字符，直到按Esc |

**其他**

其他命令模式下常用的命令见表 0‑7。

表 0‑7 其他末命令模式常用命令

|  |  |
| --- | --- |
| 命令 | 功能 |
| u | 撤消前一个操作 |
| U | 撤消一行内的所有改动 |
| Ctrl+r | 重做上一个操作 |
| J | 合并光标所在行与下一行 |

## vim末行模式下的常用操作

在vim末行模式，除了最常用的保存、退出等命令外，有更多的命令组合用于完成复杂的文本操作。

以下列出末行模式下常用的操作命令：

**搜索替换**

搜索替换的命令见表表 0‑8。

表 0‑8 搜索替换的命令

|  |  |
| --- | --- |
| 命令 | 功能 |
| n1,n2s/word1/word2/g | 将从n1行到n2行之间的word1替换为word2，如无g则只替换第一个匹配 |
| 1,$s/word1/word2/gc | 将从第一行到最后一行之间的word1替换为word2，c表示每次替换确认 |
| %s/^/word2/g | 在整个文件的每行行首插入word2 |
| %s/$/word2/g | 在整个文件的每行行尾插入word2 |
| %/var/char-&/g | 在整个文件中匹配到var后替换为char-var,&指代匹配的结果，可能为正则匹配的多种结果。 |

**其他**

其他末行模式常用命令见表 0‑9。

表 0‑9 其他末行模式常用命令

|  |  |
| --- | --- |
| 命令 | 功能 |
| w/w! | 保存文件/强制保存文件 |
| w filename | 另存为filename |
| n1,n2 w filename | 将文件的第n1行到第n2行另存到filename |
| r filename | 读取另外的文件到正在编辑的文件 |
| ！command | 暂时离开vi执行命令 |
| r !command | 把命令的输出插入到当前 |
| sh | 转动shell，输入exit返回 |
| e! | 将文件还原 |
| set nu/set nonu | 设置行号/取消行号 |
| set autoindent | 设置自动对齐格式，(取消 set noautoindent) |
| set ruler | 设置在屏幕底部显示光标所在的行列位置 |
| set ignorecase | 忽略正则表达示中大小写 |
| nohlsearch | 取消搜索到的关键字的高亮显示 |

**工作技巧**在末行模式下输入命令是可以用Tab键自动补全，也可以使用：help获得相关帮助。

## vim的其他操作

**块操作**

命令模式键入v则进入块操作：移动光标选定操作块；按y键复制；按c键剪切；按p键粘贴。

**分隔窗口**

可以在一个vim当前窗口中并排打开多个文件。

**示例：**水平分隔窗口命令，同时水平排列打开文件file1.txt和file2.txt。

1. [demo@365linux ~]$ vim -o file1.txt file2.txt

**示例：**垂直分隔空口命令，同时垂直排列打开文件file1.txt和file2.txt。

1. [demo@365linux ~]$ vi m -O file1.txt file2.txt

窗口移动快捷键：ctrl+w

**vim文件恢复**

使用vim编辑一个文件test.txt时，会在文件所在目录产生一个临时文件，文件名为.test.txt.swp，这是一个隐藏文件。我们在vi中所做的操作会暂时存在该文件内。

如果在文件编辑过程中vi非正常关闭，那么重新打开vi test.txt时，系统会提示发现交换文件.test.txt.swp，可能的原因是：

1. 有另一个程序也在编辑同一个文件；

上次编辑此文件时崩溃。

这时可以按O只读打开，或R进行修复，或E直接编辑，或Q退出。

手动删除.test.txt.swp后，则不会再出现该提示。

**工作技巧**在对系统的关键性配置文件进行编辑修改时，强烈建议先做好原始文件的备份，因为vim编辑器对文件完成修改并保存后是无法恢复的。

vim编辑器是Linux系统中最常用的命令行工具。看似繁多的操作命令其实有规律可循，记住几个基础指令，其他的技巧多为基础指令的组合，可以在此基础上举一反三。

# 用户和组

**【任务描述】**

技术部来了一位新的同事，系统管理员需要在服务器中为该员工添加一个新的用户账号。方便该员工使用自己的账号登录到服务器，进行文档存取或开发相关的工作。

**【学习目标】**

* 1. 了解用户的基本概念，LINUX系统用户类型；
  2. 使用命令行工具查看、添加、修改、删除用户；
  3. 理解添加用户时，系统的实现过程；
  4. 了解组的基本概念，添加和删除组。

## 使用命令行工具添加用户

充分理解Linux系统用户账号体系，建立账号时考虑与工作对应的标准，规范，及安全性要求。

Linux系统是一个真正意义上的多用户多任务的系统，这就意味着在系统的机制上存在着多种不同的用户，对应相应不同的权限和功能。系统中的用户可以是一个对应真实物理用户的账号，也可以是特定应用程序使用的身份账号。Linux系统通过定义不同的用户，来控制用户在系统中的权限。系统的每个文件都有设计成属于相应的用户和组，不同的用户则决定了其对系统内哪些文件是否可以访问，写入或执行。

在Linux中，有三种不同的用户类型，root（也称管理员账户，超级用户或根用户），普通用户和系统用户。

root用户是系统内置的管理员用户，拥有系统最大权限，可以完全访问和操作系统的所有文件。正因如此，在通常系统使用时，不建议使用root用户登录系统，只有在需要管理员权限时才由普通用户切换到管理员。

如果在安装系统时安装了图形界面，那么在第一次启动的时候会被要求创建一个普通用户，这个用户是被用来登录和使用系统。普通用户默认情况下只能对自己的家目录下的文件和文件夹进行修改或删除的操作，最大程度的控制了对系统的损害。如果在安装系统时没有安装图形环境，也应在进入系统行手动创建一个普通用户，用作平常登录使用。

系统用户类似于普通用户，不同之处在于系统用户通常没有自己的家目录，也不能登录到系统，仅用来控制应用程序的运行。系统用户一般由系统自带应用或其他服务程序的软件包安装时创建。

在LINUX的设计中，每个用户都有一个对应的组，组即是多个（含一个）成员用户为同一目的组成的组织，组内的成员对属于该组下的文件拥有相同的权限。默认情况下，LINUX用户拥有自己的私人组（usr private group, UPG），当一个新用户被创建时，同时会创建一个和用户名相同的用户私人组。

### useradd命令创建用户

**用法：**useradd [options] user\_name

**常用选项：**

-u UID：为新用户指定一个UID（不使用系统默认按顺序分配的），使用-r，强制建立系统账号（小于/etc/login.defs上UID\_MIN），使用-o，允许新用户使用不唯一的UID。

-g GROUP：为新用户指定一个组（指定的组必须存在）。

-G GROUPS：为新用户指定一个附加组。

-M：不创建用户的家目录（默认创建）。

-m：为新用户创建家目录。使用-k选项将skeleton\_dir内的档案复制到家目录下。

-c：为新用户进行说明注释（/etc/passwd的说明栏）。

-d：为新用户指定家目录。默认值为default\_home内的login名称。

-s：为新用户指定登录后使用的shell。

-e：为新用户指定账号的终止日期。日期的指定格式为MM/DD/YY。

-f：用户账号过期几日后永久失效。当值为0时账号则立刻失效，-1时关闭此功能。默认关闭。

**示例：**使用命令创建新用户。

1. [demo@365linux ~]$ su -
2. 密码：
3. [root@365linux ~]# useradd user01

创建用户的操作需要管理员权限，如果您是以普通用户登录系统，首先要切换到管理员用户。

**示例：**创建新用户，使用自定义选项。

1. [root@365linux ~]# useradd -u 1000 -c "ftp user" -s /sbin/nologin user02

**工作技巧**

作为一个好的习惯，你应该给每个账号添加注释说明（使用-c），否则时间长了，你可能会忘记每个用户的用途。

### passwd命令设置用户密码

**用法：**passwd [options] user\_name

**常用选项：**

-l：锁定指定的账号。

-u：解锁指定的被锁定的账号。

-n：指定密码最短时间。

-x：指定密码最长时间。

-w：指定密码过期前的警告天数。

-i：指定密码过期后，账号失效前的天数。

-S：报告指定用户密码的状态。

--stdin 从标准输入读入密码，常用于shell脚本。

**示例：**为用户设置密码。

1. [root@365linux ~]# passwd user01
2. 更改用户 user01 的密码。
3. 新的密码：
4. 重新输入新的密码：
5. passwd：所有的身份验证令牌已经成功更新。

NOTE

root可以修改任何用户的密码，普通用户仅可以修改自己的密码。

### 系统添加用户过程解释

使用useradd命令创建新用户和使用passwd命令设置用户密码，系统是如何保存用户信息。当命令执行时，系统发生了如下动作：

在系统用户信息配置文件中添加一行。可以使用cat命令查看：

1. [root@365linux ~]# cat /etc/passwd
2. root:x:0:0:root:/root:/bin/bash
3. bin:x:1:1:bin:/bin:/sbin/nologin
4. ……
5. user01:x:502:502::/home/user01:/bin/bash

在/etc/passwd文件中，每一行对应一个用户的属性信息。新创建的用户被追加在最后一行。每一行用“：”分割成七个字段，每个字段的意义是：

* + Account：用户名。
  + Password：用户密码，因为/etc/passwd所有人可读，出于安全考虑，该字段默认用x代替。真正的密码文件保存在/etc/shadow文件中。用户可以通过authconfig来设定是否使用shadow文件及md5加密。
  + UID：用户ID号。UID为0，表示是拥有最高权限的系统管理员（如root）。默认UID1-499为系统保留账号，普通用户UID一般500-60000。一般来说，用户UID号是唯一的。
  + GID：用户所属主组的组ID号。
  + GECOS：用户的注释说明（可选）。
  + Directory：用户的主目录。
  + Shell：用户所使用的shell。

useradd命令不使用任何选项，添加的新用户属性使用系统预设的默认值。新添加的user01用户UID和GID都是502，家目录是/home/user01，登录shell为/bin/bash。

在系统用户密码存储文件/etc/shadow中添加新行用户密码信息。

1. [root@365linux ~]# cat /etc/shadow
2. ……
3. user01:$6$0yEadFvM$FBkTS2Tog3U9EM6Ga5Y3BdneOBTEnbGG7oGB2qVa9dpo8EyHcN.LHj4XFQ8zluNqeBLaUghzJlRJcmIX6lAjz1:16153:0:99999:7:::

/etc/shadow文件用于存放用户密码相关的信息。该文件的权限值为000（任何人没有任何权限），只有root用户可以突破这一限制。该文件同样用“：”将文件分割成9个字段，每一个字段的意义如下：

* + 登录名
  + 经md5加密的用户密码（前面有\*或！，则账号被锁定无法使用）
  + 密码上一次被更改的日期（格式：从1970年1月1日起的天数）
  + 密码不可更改的天数（密码最短时间，密码要过多少天才能被修改）
  + 密码过期时间（密码最长时间，密码过多少天后必须被更改）
  + 密码过期前警告时间
  + 密码过期几天后账号失效（在此时间段内要求用户修改密码）
  + 账号失效日期（格式：从1970年1月1日起的天数）
  + 保留，目前未定义

默认情况下，用户被设置为永不过期。

在组账号信息配置文件/etc/group中添加相关私人组新行。

1. user01:x:502:

组账号信息配置文件三个字段分别表示：组名；密码标示（加密密码存储在/etc/gshadow中）；组ID号，GID应该与/etc/passwd文件中对应用户的GID一致。

在组的密码存储文件/etc/gshadow中添加新行。

1. user01:!::

在系统/home目录下创建用户的同名家目录。

1. [root@365linux ~]# ls -l /home
2. drwx------. 4 user01 user01 4096 3月 24 17:42 user01

用户家目录（也称主目录或主文件夹）属于user01用户和user01组，只有user01用户对该文件夹有读、写、执行的权限，所有其他的权限被拒绝。关于文件权限将在下一节详细说明。

将/etc/skel目录下默认的用户配置文件复制到用户的家目录。

1. [root@365linux ~]# ls -la /home/user01
2. 总用量 28
3. drwx------. 4 user01 user01 4096 3月 24 17:42 .
4. drwxr-xr-x. 5 root root 4096 3月 24 18:01 ..
5. -rw-r--r--. 1 user01 user01 18 7月 9 2013 .bash\_logout
6. -rw-r--r--. 1 user01 user01 176 7月 9 2013 .bash\_profile
7. -rw-r--r--. 1 user01 user01 124 7月 9 2013 .bashrc
8. drwxr-xr-x. 2 user01 user01 4096 7月 14 2010 .gnome2
9. drwxr-xr-x. 4 user01 user01 4096 3月 5 02:54 .mozilla

在系统邮件存储目录创建用户邮箱文件（可选）。

1. [root@365linux ~]# ll /var/mail/
2. -rw-rw----. 1 user01 mail 0 3月 24 17:42 user01

### usermod命令修改用户账号信息

**用法：**usermod [option] user\_name

**常用选项：**

-L：锁定账号（在/etc/shadow中密码部分前加一个！），-U解锁。

-l：改变用户的登录名。

除以上选项，usermod使用与useradd相似的选项和参数指定修改用户账号属性信息。

**示例：**修改用户账号信息的注释说明。

1. [root@365linux ~]# usermod -c "web user" user01
2. [root@365linux ~]# cat /etc/passwd |grep user01
3. user01:x:502:502:web user:/home/user01:/bin/bash

**示例：**使用usermod锁定和解锁账号。

1. 使用用-L选项锁定用户，限制用户登录。
2. [root@365linux ~]# usermod -L user01

查看用户的密码文件，此时在密码字段前添加了“！”号，用户不能登录。

1. [root@365linux ~]# grep user01 /etc/shadow
2. user01:!$6$0yEadFvM$FBkTS2Tog3U9EM6Ga5Y3BdneOBTEnbGG7oGB2qVa9dpo8EyHcN.LHj4XFQ8zluNqeBLaUghzJlRJcmIX6lAjz1:16153:0:99999:7:::

使用-U的选项解锁被锁定的用户。

1. [root@365linux ~]# usermod -U user01
2. [root@365linux ~]# grep user01 /etc/shadow
3. user01:$6$0yEadFvM$FBkTS2Tog3U9EM6Ga5Y3BdneOBTEnbGG7oGB2qVa9dpo8EyHcN.LHj4XFQ8zluNqeBLaUghzJlRJcmIX6lAjz1:16153:0:99999:7:::

### userdel命令删除用户

**用法：**userdel [options] user\_name

**常用选项：**

-f：强制删除用户即使该用户仍在登录；

-r：删除用户的同时删除该用户的家目录和邮件。

**示例：**删除一个用户。

1. [root@365linux ~]# userdel user01
2. [root@365linux ~]# ls /home
3. demo user01 user02

没有使用-r的选项，删除了用户，而用户的家目录被保留。

**NOTE**

如果用户同名组没有其他成员，则连同删除，反之保留。

**工作技巧**

注意在系统中新建一个用户时，系统可能将一个已经删除的旧用户的UID重新分配给新用户，如果在删除旧用户时没有删除该用户的文件（在家目录或散落在其他位置），新用户就获得旧用户原来在系统中的文件的所有权，这种情况将导致信息泄漏和其他安全问题。解决的方案是使用find命令指出这些文件进行删除或备份（改变权限）。

## 使用命令行管理工作组群

使用命令行工具添加组群，和设置组群密码。并深刻理解系统如何保存组群信息和相关文件。掌握使用命令行添加组群，并设置组群密码的方法；掌握使用命令行修改组群，设置组群管理员，添加，删除组成员的管理方法；深刻理解系统如何保存组群信息的过程。

### groupadd命令新建组群

**用法：**groupadd [options] group\_name

**常用选项：**

-g：指定新建组的GID。使用-o的选项，可以使用重复的GID。使用-r的选项来建立系统账号。

**示例：**新建一个新组。

1. [root@365linux ~]# groupadd sales

### 组群管理命令示例

**示例：**给组群添加密码，使知道密码的用户能加入临时到该组。

1. 修改组sales的密码。没有设密码的组是不允许用户申请加入的。
2. [root@365linux ~]# gpasswd sales

添加用户zhangsan，切换到用户，查看用户当前所属的组群。

1. [root@365linux ~]# useradd zhangsan
2. [root@365linux ~]# su – zhangsan
3. [zhangsan@365linux ~]$ groups
4. zhangsan

用户主动申请加入组群sales。

1. [zhangsan@365linux ~]$ newgrp sales
2. 密码：
3. [zhangsan@365linux ~]$ groups
4. sales zhangsan

用户退出组群sales。

1. [zhangsan@365linux ~]$ exit
2. [zhangsan@365linux ~]$ groups
3. zhangsan

删除组sales的密码，禁止以后用户再主动加入。

1. [root@365linux ~]$ gpasswd -r sales

**示例：**设置组群管理员，用来管理组成员。

1. 添加用户lisi，将用户zhangsan加入到sales组中。
2. [root@365linux ~]# useradd lisi
3. [root@365linux ~]# gpasswd -a lisi sales

将用户lisi设置为组sales的管理员

1. [root@365linux ~]# gpasswd -A lisi sales

切换到用户lisi，行使管理权限，将用户zhangsan加入组sales。

1. [root@365linux ~]# su - lisi
2. [lisi@365linux ~]$ gpasswd -a zhangsan sales
3. Adding user zhangsan to group sales

此时用户zhangsan成为组sales的成员，切换组时不需要组密码。

1. [zhangsan@365linux ~]$ groups
2. zhangsan sales
3. [zhangsan@365linux ~]$ newgrp sales
4. [zhangsan@365linux ~]$ groups
5. sales zhangsan

组管理员lisi将用户zhangsan从sales组中删除。

1. [lisi@365linux ~]$ gpasswd -d zhangsan sales
2. Removing user zhangsan from group sales

**示例**：修改组名，将组sales的组名称改为xiaoshou。

1. [root@365linux ~]# groupmod -n sales xiaoshou

**示例**：删除组xiaoshou。

1. [root@365linux ~]# groupdel xiaoshou

### 其他有用的命令

#### id命令输出用户简要信息

**示例：**查看用户user02的uid、gid、组。

1. [root@365linux ~]# id user02
2. uid=1000(user02) gid=1000(user02) 组=1000(user02)

#### 使用chage查看或修改用户的账号和密码信息：

**示例：**查看用户的账号和密码信息。

1. [root@365linux ~]# chage -l user02
2. Last password change : Mar 24, 2014
3. Password expires : never
4. Password inactive : never
5. Account expires : never
6. Minimum number of days between password change : 0
7. Maximum number of days between password change : 99999
8. Number of days of warning before password expires : 7

**示例：**设置用户user02账号过期时间是2014-12-28。

1. [root@365linux ~]# chage -E "2014-12-28" user02

**工作技巧**

在创建用户时或之后为用户设置有效期，对于一些雇用人员临时开启的账号是非常有用的。账号在临时人员离开后自动失效，可以避免一些安全上的隐患。

#### 使用pwck对用户进行一致性检查

**示例：**检查系统内用户账号信息的完整性。

1. [root@365linux ~]# pwck
2. user 'adm': directory '/var/adm' does not exist
3. ……
4. pwck：无改变

输出中系统账号的家目录信息的缺失是正常的。

## 切换用户获得系统管理权限

使用su和sudo切换系统账号，获得管理员权限，比较两种切换方式的区别。使用su切换到管理员；配置使用sudo方式切换到管理员。

## 切换账号

在命令行中，可以很方便的切换到不同的用户。普通用户要想切换到任意别的用户，必须输入目标用户的密码；管理可以随时切换到其他的用户而不需要用户密码。切换用户可以使用以下命令：

**su**  切换到其他的用户，即以目标用户的身份登录系统继续工作。

**sudo** 并不真的切换用户，而是以管理员的身份执行命令。

## 使用su命令切换账号

**示例：**用户user02切换到user03，前提是uesr03设置了密码可以登录。

1. [user02@365linux ~]$ su - user03
2. 密码：
3. [user03@365linux ~]$

**示例：**用户user03退出登录，返回user02。

1. [user03@365linux ~]$ exit
2. logout
3. [user02@365linux ~]$

**示例：**用户user02切换到管理员用户root，用户名可以省略。

1. [user02@365linux ~]$ su -
2. 密码：
3. [root@365linux ~]#

**工作技巧**

在切换用户时使用su命令，在su和用户名之间的选项使用“-”或不使用，结果存在差异。不使用时，切换用户但不会切换到目标用户的目录和shell环境变量。而使用时任何环境因素都会初始化，效果相当于目标用户登录后。

## 使用sudo命令获得root权限

在生产环境中，特别是多用户进行项目协作的场景下，需要管理员权限时，管理员不能直接告诉用户root账户的密码，而应该使用sudo方式授予普通用户部分管理员权限。这在权限控制和问责机制中非常重要。

在LINUX系统中，普通用户要使用sudo方式临时获得管理员的权限执行命令，需要管理员提前将这个普通用户添加到sudo的配置文件中的授权用户列表中。默认情况下，所以的普通用户均不可以使用sudo。

**示例：**授权普通用户可以使用sudo临时获得root权限。

1. 使用visudo命令编辑sudo配置文件/etc/sudoers。
2. [root@365linux ~]# visudo
3. demo ALL=(ALL) ALL

在文件的最后加入demo用户，这一行配置的含义是允许demo用户使用sudo在任何地方运行任何命令，即授予最大权限。

切换到普通用户demo，演示使用sudo方式可以实现创建新用户。

1. [root@365linux ~]# su - demo
2. [demo@365linux ~]$ useradd user05
3. -bash: /usr/sbin/useradd: 权限不够
4. [demo@365linux ~]$ sudo useradd user05
5. [sudo] password for demo:

使用sudo时需要确认用户的身份，此时输入的demo用户自身的密码，而不是root用户的。这样即可以给普通用户授予管理系统的权限，又避免了泄漏root用户的密码。而且在必要时，容易根据不同的用户记录追踪用户的操作记录。

## 用户账号初始化

**用户特定配置文件**

当用户被创建时，系统从/etc/skel/目录下复制用户的配置文件到用户的家目录。这些配置文件用来定义用户的工作环境，比如PATH路径、命令别名等。这些文件位于每个用户的家目录内，所以只对当前用户有效。如下：

~/.bashrc 定义函数和别名

~/.bash\_profile 设置环境变量

~/.bash\_logout 定义用户退出时执行的命令

**工作技巧**

对于要在系统中经常执行的复杂的命令可以使用别名的方式将其输入简化，为了保证下一次启动依然有效，可将别名定义在bashrc文件中。

**全局用户配置文件**

顾名思义，这些是对所有用户都生效的设置。如下：

/etc/bashrc 定义函数和别名

/etc/profile 设置环境变量

/etc/profile.d 目录下的脚本被/etc/profile引用

**系统预设的值**

用户的属性信息的默认值在文件/etc/login.defs中被定义。

**示例：**查看/etc/login.defs，使用grep命令过滤掉文件中的以#开头的注释行。

1. [root@365linux ~]# grep -v '^#' /etc/login.defs
2. MAIL\_DIR /var/spool/mail
3. PASS\_MAX\_DAYS 99999
4. PASS\_MIN\_DAYS 0
5. PASS\_MIN\_LEN 5
6. PASS\_WARN\_AGE 7
7. UID\_MIN 500
8. UID\_MAX 60000
9. GID\_MIN 500
10. GID\_MAX 60000
11. CREATE\_HOME yes
12. UMASK 077
13. USERGROUPS\_ENAB yes
14. ENCRYPT\_METHOD SHA512

创建新用户的默认属性信息在配置文件/etc/default/useradd中被定义。

**示例：**查看/etc/default/useradd文件中的配置项。

1. [root@365linux ~]# cat /etc/default/useradd
2. # useradd defaults file
3. GROUP=100
4. HOME=/home
5. INACTIVE=-1
6. EXPIRE=
7. SHELL=/bin/bash
8. SKEL=/etc/skel
9. CREATE\_MAIL\_SPOOL=yes

**示例：**使用uesradd命令列出系统预设的添加用户信息的默认值。

1. [root@365linux ~]# useradd -D
2. GROUP=100
3. HOME=/home
4. INACTIVE=-1
5. EXPIRE=
6. SHELL=/bin/bash
7. SKEL=/etc/skel
8. CREATE\_MAIL\_SPOOL=yes

命令输出结果分别表示新创建用户的组群ID初始值；用户家目录的上一级目录；账号过期后失效时间；账号过期日期；默认的登录shell；用户默认配置文件的源目录（从指定的目录中拷贝配置文件到用户的家目录）；默认创建用户邮箱文件。

可以通过useradd –D 加上相应的选项更新这些预设值。

**useradd -D 可使用的选项：**

-b：定义用户家目录的上一级目录；

-e：用户账号的过期日期；

-f：用户账号过期几日后失效；

-g：新建用户的起始群组或ID；

-s：新建用户登录后使用的shell。

主要了解了Linux用户和组的管理。用户和组是Linux系统的基本概念，是实现多用户协作，权限分配，文件共享等任务的基础。理解普通用户和root用户的特点，在需要时切换到root用户或者获取管理权限。

在实际工作中，管理员要对系统全局环境或者用户的特定文件进行设置，以满足项目对于用户需求。比如设置所有新建用户自动属于某个特定的组。

当需要一次性重复创建很多账号时，可以编辑shell脚本批量处理。

不管实际应用如何变化，用户和组的基本原则是不变的。

# 权限管理

**【任务描述】**

通过文件的权限控制用户对文件的访问。Linux文件权限系统简单而又灵活，易于理解和应用，又可以轻松地处理最常见的权限情况。文件只具有三种应用权限的用户类别和三种基础的控制权限。

**【学习目标】**

* 1. 理解Linux系统文件权限设计；
  2. 使用命令行工具管理权限。

## Linux系统权限

应用权限的用户类别：

**文件拥有者**

文件归用户所有，通常是创建文件的用户。但可以更改。

**文件所属组**

文件归单个组所有，通常为创建该文件的用户的主要组。但可以更改。

**其他用户**

除了拥有者，所属组外的其他用户。

应用权限时，用户权限优先级高于组的权限，高于其他人的权限。

**NOTE**

每个用户都有自己所属的组，每个文件也有所属的组，这两个组的意义是不同的。文件所属组可以恰好是该文件拥有者的所属组，也可以是另一个不同的组。

三种基础的控制权限：读取、写入和执行。这些权限的作用如下表0‑1所示：

表0‑1文件和目录的权限

|  |  |  |
| --- | --- | --- |
| 权限 | 对文件的影响 | 对目录的影响 |
| r（读取） | 可以读取文件的内容 | 可以列出目录的内容（文件名） |
| w （写入） | 可以更改文件的内容 | 可以创建或删除目录中的任一文件 |
| x （执行） | 可以作为命令执行文件 | 可以访问目录的内容（还取决于目录中文件的权限） |

**NOTE**

访问并进入目录需要用户同时对目录有读取和执行的权限。如果用户仅对某目录有读取的访问权限，则可以列出其中文件的名称，但是其他信息（包括权限或时间戳）都不可用，也不可访问。如果用户仅对某目录具有执行的权限，则用户不能列出该目录中文件的名称，但是如果用户已经知道对其具有读取权限的文件的名称，那么可以通过明确指定文件名来访问该目录下指定文件的内容。

默认情况下，如果用户对某个目录具有写入的权限，那么他可以删除该目录下的任何文件，不论被删除文件的拥有者是谁，权限设置如何。

**示例：**在命令行中列出文件的属性。

1. [demo@365linux ~]$ ll /etc/man.config
2. -rw-r--r--. 1 root root 494011月 17 2012 /etc/man.config
3. 文件类型与权限链接数拥有者所属组文件大小文件被创建或修改日期文件名

除去文件类型的标识位，文件权限字段一共9个符号，每三个符号为一组，分别表示文件拥有者的权限，所属组的权限和其他人的权限。如示例中的文件，拥有者为root用户，他的权限为rw-（可读、可写、执行权限位上没有执行权限）；所属组为root组，所属组的权限为r--（仅有可读的权限）；其他人也是仅有可读的权限。

**示例：**在命令行中列出目录的属性

1. [demo@365linux ~]$ ll -d /home/demo/
2. drwx------. 29 demo demo 4096 3月 24 04:48 /home/demo/

该目录的权限为只有拥有者demo用户对它有读、写、执行的权限，所属组和其他人都没有任何权限。这是用户家目录的特点，只有用户本身才能访问自己的家目录。

## 在命令行中管理权限

### 使用chmod命令更改文件权限

**用法：**chmod [选项]... 模式[,模式]... 文件...

或：chmod [选项]... 八进制模式文件...

或：chmod [选项]... --reference=参考文件文件...

**常用选项：**

-R：以递归方式更改所有的文件及子目录。

符号模式： chmod WhoWhatWhich file|directory

简单理解：chmod [-R] [ugoa][+-=][rwx] 文件|目录

Who是指u、g、o、a （代表拥有者、组、其他、全部）

What是指+、-、= （代表添加、删除、精确设置）

Which是指r、w、x （代表读取、写入、可执行的权限）

八进制模式：chmod [-R] ### file|directory

###是指三位数字，每一个数字代表一个权限对象类别：拥有者、组、其他人。

#一位数字是指在该对象（比如拥有者）上的权限的总和，r=4、w=2、x=1。那么如果文件的拥有者的权限是rwx，用数字表示就是7（即4+2+1=7）。

**示例：**设置file文件的拥有者加上执行的权限，其他人去除读取的权限。

1. [demo@365linux ~]$ ll file
2. -rw-rw-r--. 1 demo demo 267 3月 19 12:45 file
3. [demo@365linux ~]$ chmod u+x,o-r file
4. [demo@365linux ~]$ ll file
5. -rwxrw----. 1 demo demo 267 3月 19 12:45 file

注意文件前后的权限变化，当文件有了可执行权限时，文件在终端中会颜色高亮。

**示例：**精确设置文件的权限。

1. [demo@365linux ~]$ chmod u=rw,g=r file

设置file文件的拥有者的权限为可读可写，组仅可读，其他人没有权限。

**示例：**统一设置目录及目录下所有文件的权限。

1. [demo@365linux ~]$ chmod -R +w targetdir/

给目录targetdir及该目录下的所有文件在所有用户类型权限位上添加可写的权限。

**示例：**使用数字方式设置文件的权限。

1. [demo@365linux ~]$ chmod 755 file

设置文件file的权限为rwxr-xr-x，即拥有者可读可写可执行，组和其他人都为可读可执行。

### 使用chown命令更改文件用户所有权

要更改文件或文件夹的用户或组的所有权，使用chown或chgrp命令。

**用法：**chown [选项]... [所有者][:[组]] 文件...

或：chown [选项]... --reference=参考文件文件...

**示例：**改变文件file的所有者。

1. [root@365linux demo]# ll file
2. -rwxr-xr-x. 1 demo demo 267 3月 19 12:45 file
3. [root@365linux demo]# chown zhangsan file
4. [root@365linux demo]# ll file
5. -rwxr-xr-x. 1 zhangsan demo 267 3月 19 12:45 file

**示例：**改变文件file的所属组。

1. [root@365linux demo]# chgrp mail file
2. [root@365linux demo]# ll file
3. -rwxr-xr-x. 1 zhangsan mail 267 3月 19 12:45 file

**示例：**同时改变用户和组的所有权，并使用-R选项递归目录和目录下的所有文件。

1. [root@365linux demo]# chown zhangsan:mail targetdir/

**NOTE**

在变更文件或目录的用户和组的所有权时，前提条件是指定的目标用户和组在系统中已经存在。

## 特殊权限

系统中有些地方的设计需要一些特殊的权限。比如passwd命令，普通用户在使用passwd命令修改自己的密码时，也需要更新/etc/shadow文件，而该文件只有管理员可以修改，这就意味着，普通用户成功修改了自己的密码，那么他在运行passwd命令时获得得管理员的权限或者说是使用root的身份在执行。这是怎么做的呢。

**示例：**查看passwd命令的权限。

1. [demo@365linux ~]$ ll `which passwd`
2. -rwsr-xr-x. 1 root root 25980 2月 17 2012 /usr/bin/passwd

看到在root用户的权限位上是rws，这表示在执行权限位上，除了x还有一个setuid的权限，这个就是特殊权限的一种。

特殊权限对文件和目录的影响如表0‑2所示：

表0‑2特殊权限对文件和目录的影响

|  |  |  |
| --- | --- | --- |
| 特殊权限 | 对文件的影响 | 对目录的影响 |
| u+s (setuid或者suid) | 以拥有文件的用户身份执行文件，而不是以运行文件的用户身份 | 无影响。 |
| g+s (setgid或者sgid) | 以拥有文件的组身份执行文件 | 在目录中最新创建的文件将其组所有者设置为与目录的组所有者相同 |
| o+t (sticky) | 无影响 | 对目录具有写入权限的用户仅可以删除其拥有的文件，而无法删除其他用户所拥有的文件 |

**示例：**含有sgid权限的文件。

1. [root@365linux ~]# ll /usr/bin/wall
2. -r-xr-sr-x. 1 root tty 10932 6月 18 2013 /usr/bin/wall

**示例：**含有sticky权限的目录。

1. [root@365linux ~]# ll -d /tmp
2. drwxrwxrwt. 30 root root 4096 3月 26 08:17 /tmp

**NOTE**

因为特殊权限和执行权限在同一个位置上，所以特殊权限的大小写表示了该位置上是否还有可执行权限。如果特殊权限为小写，则表示此处有x权限；如果特殊权限为大写，则表示此处没有x权限。不过从特殊权限的应用角度来看，其总是要伴随x权限一起使用才有实际意义。

**设置特殊权限**

用符号法设置：setuid = u+s；setgid = g+s；sticky = o+t

用数字法设置：setuid = 4；setgid = 2；sticky = 1

**示例：**给目录设置sgid权限。

1. [root@365linux ~]# chmod u+s directory
2. 或者
3. [root@365linux ~]# chmod 2775 directory

**工作技巧**

因为setuid可以让普通用户拥有root的身份和权限，所以为安全起见，您应该留意系统中含有setuid位的程序或脚本，是否异常。可以通过find命令从系统中找出它们：

[root@365linux ~]# find / -perm -4000

### 隐藏的扩展属性（权限）

为了极大的保证文件的安全，Linux文件系统还文件预留了一些扩展的属性，比如设置让root用户也无法删除文件的权限。这些属性被隐藏了起来，需要特定的命令才能查看和设置。

**示例：**查看文件的扩展属性。

1. [root@365linux ~]# lsattr testfile.txt
2. -------------e- testfile.txt

“e”就是一个扩展属性，表示使用磁盘块映射。这是一个默认的属性，不能去除。

**示例：**设置文件的扩展属性。

1. [root@365linux ~]# chattr +i testfile.txt
2. [root@365linux ~]# lsattr testfile.txt
3. ----i--------e- testfile.txt

给文件添加“i”的属性表示该文件不能被删除、不能修改、不能重命名、不能创建链接。更多的文件扩展属性设置选项可以通过“man chattr”来获得详细的说明。

### 访问控制列表ACL

Linux系统通常的权限管理只是针对文件或目录的拥有者、所属组，其他人进行读、写、执行的权限的划分。如果要对某个文件或目录进行除了上述三类用户的单一特定用户或组进行更细致的权限划分，比如，针对某个文件，除了拥有者和组之外，其他人都没有读取的权限，而唯独用户zhangsan需要读取的权限（zhangsan属于其他人的范围），这就需要借助ACL（Access Control List）进行。

ACL的使用需要文件系统的支持，目前绝大部分的Linux文件系统（如EXT2/3/4，JFS，XFS）支持ACL的功能。在LINUX中ACL是默认启动的。如果系统默认没有启动ACL的功能，则需要添加ACL属性并重新挂载文件系统以获取ACL的支持（后面的章节介绍文件系统挂载的操作）。

在确定某个进程是否能够访问某一文件时，权限的优先级如下：

1. 如果是以文件的拥有者身份运行该进程，那么就应用该文件的拥有者权限；

如果是以列于用户ACL条目中的用户运行该进程，那么就应用用户ACL（只要受mask允许）；

如果是以文件的所属组身份或具有明确组ACL条目的组身份运行该进程，如果权限是由任意匹配组授予的，则应用组的权限（只要mask许可）

否则，应用文件的其他权限。

其中mask称作具有ACL的文件的掩码，用于限制组成员和ACL补充用户和组成员的最大权限。

**ACL的命令**

**getfacl命令**

**描述：**查看文件或目录的ACL权限。

**用法：**getfacl [options] file …

**示例：**查看文件的ACL

1. [root@365linux test]# getfacl testfile
2. # file: testfile
3. # owner: root
4. # group: root
5. user::rw-
6. user:zhangsan:rwx
7. group::r--
8. mask::rwx
9. other::r--

文件testfile的用户和组都是root，它的权限是644，只有root用户是读写，其他人都是只读。而设置了ACL，ACL用户zhangsan则对该文件拥有读、写、执行的权限。

对于添加了ACL的文件，使用ls -l命令列出时，权限位的最后带有“+”号。如下：

1. [root@365linux test]# ll testfile
2. -rw-rwxr--+ 1 root root 75 3月 26 09:33 testfile

**setfacl命令**

**描述：**设置某个文件或目录的ACL权限

**用法：**setfacl [-bkndRLP] { -m|-M|-x|-X ... } file ...

**常用选项：**

-m：设置或修改文件的ACL权限；

-x：取消文件的一个ACL权限；

-b：删除文件所有的ACL权限；

-k：删除所有的默认的ACL权限；

--set：设置文件的ACL，替代当前的ACL；

--mask：重新计算有效的mask值；

-R：递归子目录；

-d：设置默认的ACL权限，仅能针对目录使用；

--restore：从文件恢复备份的ACL。

**查看和设置ACL**

**示例：**使用setfacl命令设置文件的ACL。

1. 针对用户zhangsan来设置权限为rwx。
2. [root@365linux test]# setfacl -m u:zhangsan:rwx acltest.file

针对组sales来设置权限为rw。

1. [root@365linux test]# setfacl -m g:sales:rw acltest.file

设置限制的权限为r

1. [root@365linux test]# setfacl -m m:r acltest.file

查看文件当前的ACL。

1. [root@365linux test]# getfacl acltest.file
2. # file: acltest.file
3. # owner: root
4. # group: root
5. user::rw-
6. user:zhangsan:rwx #effective:r--
7. group::r--
8. group:sales:rw- #effective:r--
9. mask::r--
10. other::r--

最终的权限：文件的拥有者root为rw-；ACL用户zhangsan的权限为rwx，但因为设置了限制权限mask为r--，两个权限相与后，最终zhangsan对该文件的权限为r--；同理文件所属组的权限为r--；ACL组zhangsan的权限受mask的影响也为r--；其他人的权限为r--。

设置ACL权限，如果同时设置多个权限，权限之间使用“，”分隔。

**示例：**同时设置多个ACL用户和组的权限。

1. [root@365linux test]# setfacl -m u:user03:rwx,u:user04:rwx,g:sales:rw testfile

**删除文件的ACL权限**

**示例：**删除ACL权限

1. [root@365linux test]# setfacl -x g:zhangsan acltest.file

示例中删除的ACL组的权限，删除ACL用户权限用户类似。删除后可执行getfacl命令查看结果。

**示例：**一次性删除所有的ACL权限。

1. [root@365linux test]# setfacl -b acltest.file

**设置目录默认的ACL**

如果希望在一个目录中新建的文件和子目录都使用同一个预定的ACL，那么我们可以使用默认(Default)ACL。在对一个目录设置了默认的ACL以后，每个在目录中创建的文件都会自动继承目录的默认ACL作为自己的ACL。

**示例：**设置目录默认的ACL。

1. 设置目录testdir的ACL指定组sales的权限为rwx。
2. [root@365linux test]# setfacl -d -m g:sales:rwx testdir/
3. 查看目录testdir当前的ACL权限。
4. [root@365linux test]# getfacl testdir/
5. # file: testdir/
6. # owner: root
7. # group: root
8. user::rwx
9. group::r-x
10. other::r-x
11. default:user::rwx
12. default:group::r-x
13. default:group:sales:rwx
14. default:mask::rwx
15. default:other::r-x

在目录testdir中创建一个文件file.txt。

1. [root@365linux test]# touch testdir/file.txt

查看该文件的权限，自动继承了上级目录的ACL（受文件默认的mask影响）。

1. [root@365linux test]# getfacl testdir/file.txt
2. # file: testdir/file.txt
3. # owner: root
4. # group: root
5. user::rw-
6. group::r-x #effective:r--
7. group:sales:rwx #effective:rw-
8. mask::rw-
9. other::r--

**备份和恢复ACL**

主要的文件操作命令cp和mv都支持备份时保留文件的ACL，cp命令需要加上-p参数。但是tar等常见的备份工具是不会保留目录和文件的ACL信息的。这种情况下，如果备份和恢复带有ACL的文件和目录，那么可以先把文件的ACL权限信息备份到一个文件里。以后用--restore选项来回复这个文件中保存的ACL信息。

**示例：**文件的ACL信息备份和恢复

1. 查看目录及其所有子目录和文件当前的ACL信息
2. [root@365linux test]# getfacl -R testdir/
3. # file: testdir/
4. # owner: root
5. # group: root
6. user::rwx
7. group::r-x
8. other::r-x
9. default:user::rwx
10. default:group::r-x
11. default:group:sales:rwx
12. default:mask::rwx
13. default:other::r-x
14. # file: testdir//file.txt
15. # owner: root
16. # group: root
17. user::rw-
18. group::r-x #effective:r--
19. group:sales:rwx #effective:rw-
20. mask::rw-
21. other::r--

备份目录及其子目录中文件的ACL

1. [root@365linux test]# getfacl -R testdir/ > testdir.acl

为测试，删除原文件所有的ACL

1. [root@365linux test]# setfacl -R -b testdir/

查看删除后的权限

1. [root@365linux test]# getfacl -R testdir/
2. # file: testdir/
3. # owner: root
4. # group: root
5. user::rwx
6. group::r-x
7. other::r-x
8. # file: testdir//file.txt
9. # owner: root
10. # group: root
11. user::rw-
12. group::r--
13. other::r--

从testdir.acl文件中恢复被删除的ACL信息

1. [root@365linux test]# setfacl --restore testdir.acl

查看恢复后的效果

1. [root@365linux test]# getfacl -R testdir
2. 略……和备份前一样。

因为Linux系统的基本权限设置了用户、组、其他三种所有者类型，读、写、执行三种文件权限简单明了的文件权限管理模式，所以对Linux文件权限管理是比较简洁而容易配置的。实际应用中，对于文件权限的理解非常重要，它是Linux系统对于系统资源和进程之间关系的基础。在后续对于系统应用、服务的配置过程中，程序无法启动，服务无法正常提供资源，很多情况下可能是由文件的权限配置不正确引起，甚至需要优先排查。而粗放的文件权限管理也会给系统的安全埋下隐患。

# 网络连接

**【任务描述】**

一台Linux系统的主机，不管是作为桌面系统、工作站、还是服务器，大多数情况下，它都需要连接到网络。作为客户端，还是服务器端，连接到内网（局域网），还是外网（广域网），使用静态IP还动动态获取地址。如果您明确知道这些名词的概念，那么仅仅是配置Linux系统网络的方法是非常容易的，甚至在拥有DHCP（动态主机设置协议）服务的网络环境中，Linux系统已经完成自动配置并连接到网络。然而我们还是要深刻理解这些操作背后的原理和逻辑。

**【学习目标】**

* 1. IP地址、子网划分、路由及DNS的概念；
  2. 使用命令行工具配置系统IP地址；
  3. 使用ssh客户端工具连接到系统。

## 认识什么是网络

网络是指利用通信设备和线路将地理位置不同、功能独立的多个主机系统连接起来，以功能完善的网络软件实现网络上的硬件、软件及资源的共享和信息的传递。简单来说即连接两台或多台主机进行通信的系统。这里的“主机”通常是指计算机，但也不一定，它也可以是打印机、智能手机等。随着技术的发展，将来会有越来越多的不同类型的设备可连接到网络。

### 网络类型

网络按照其覆盖的范围可以分成局域网和广域网。

局域网（Local Area Network, LAN），又称内网。指覆盖局部区域（如办公室或楼层）的计算机网络。

广域网（Wide Area Network, WAN），又称外网、公网。是连接不同地区的局域网或城域网的计算机通信的远程网。通常跨接很大的物理范围，所覆盖的范围从几十公里到几千公里，它能连接多个地区、城市和国家，或横跨几个洲并能提供远距离通信，形成国际性的远程网络。广域网概念上并不等于互联网。

网络按照交换方式分，又可分为以太网、公用电话交换网、分组交换网、令牌环网等。

**NOTE**

网络是一个宽泛的概念，并不一定是指计算机网络。在对于Linux系统的学习，从应用角度出发，我们通常所说局域网一般情况下指内部计算机以太网（可能涉及光纤网络），而外网通常是指连接到外部的互联网。理解这两个概念。

### 互联网相关协议

有关互联网的协议可以分为3层：

最底层的是IP协议(英语：Internet Protocol, 即互联网协议），是用于报文交换网络的一种面向数据的协议，这一协议定义了数据包在网际传送时的格式。目前使用最多的是IPv4版本，这一版本中用32位定义IP地址，尽管地址总数达到43亿，但是仍然不能满足现今全球网络飞速发展的需求，因此IPv6版本应运而生。在IPv6版本中，IP地址共有128位，“几乎可以为地球上每一粒沙子分配一个IPv6地址”。IPv6目前并没有普及，许多互联网服务提供商并不支持IPv6协议的连接。但是，可以预见，将来在IPv6的帮助下，任何智能设备都有可能连入互联网。

上一层是UDP协议和TCP协议，它们用于控制数据流的传输。UDP是一种不可靠的数据流传输协议，仅为网络层和应用层之间提供简单的接口。而TCP协议则具有高的可靠性，通过为数据报加入额外信息，并提供重发机制，它能够保证数据不丢包、没有冗余包以及保证数据报的顺序。对于一些需要高可靠性的应用，可以选择TCP协议；而相反，对于性能优先考虑的应用如流媒体等，则可以选择UDP协议。

最顶层的是一些应用层协议，这些协议定义了一些用于通用应用的数据报结构，其中包括：

DNS：域名服务；

FTP：服务使用的是文件传输协议；

HTTP：所有的Web页面服务都是使用的超级文本传输协议；

POP3：邮局协议；

SMTP：简单邮件传输协议；

Telnet：远程登陆等。

### OSI与TCP/IP参考模型

#### OSI参考模型

开放式系统互联通信参考模型（英语：Open System Interconnection Reference Model，ISO/IEC 7498-1），简称为OSI模型（OSI model），一种概念模型，由国际标准化组织（ISO）提出，一个试图使各种计算机在世界范围内互连为网络的标准框架。

OSI将计算机网络体系结构划分为以下七层，如图0‑1所示：

**第7层应用层**

应用层能与应用程序界面沟通，以达到展示给用户的目的。在此常见的协议有: HTTP，HTTPS，FTP，TELNET，SSH，SMTP，POP3等。

**第6层表示层**

表示层能为不同的客户端提供数据和信息的语法转换内码，使系统能解读成正确的数据。同时，也能提供压缩解压、加密解密。

**第5层会话层**

会话层用于为通信双方制定通信方式，并创建、注销会话（双方通信）。

**第4层传输层**

传输层用于控制数据流量，并且进行调试及错误处理，以确保通信顺利。而传送端的传输层会为分组加上序号，方便接收端把分组重组为有用的数据或文件。

**第3层网络层**

网络层的作用是决定如何将发送方的数据传到接收方。该层通过考虑网络拥塞程度、服务质量、发送优先权、每次路由的耗费来决定节点X到节点Y的最佳路径。我们熟知的路由器就工作在这一层，通过不断的接收与传送数据使得网络变得相互联通。

**第2层数据链路层**

首先数据链路层的功能在于管理第一层的比特数据，并且将正确的数据传送到没有传输错误的路线中。创建还有辨认数据开始以及退出的位置同时予以标记。另外，就是处理由数据受损、丢失甚至重复传输错误的问题，使后续的层级不会受到影响，所以它运行数据的调试、重传或修正，还有决定设备何时进行传输。设备有：Bridge桥接器switch交换器

**第1层物理层**

物理层定义了所有电子及物理设备的规范。其中特别定义了设备与物理媒介之间的关系，这包括了针脚、电压、线缆规范、集线器、中继器、网卡、主机适配器（在SAN中使用的主机适配器）以及其他的设备的设计定义。因为物理层传送的是原始的比特数据流，即设计的目的是为了保证当发送时的信号为二进制“1”时，对方接收到的也是二进制“1”而不是二进制“0”。因而就需要定义哪个设备有几个针脚，其中哪个针脚发送的多少电压代表二进制“1”或二进制“0”，还有例如一个bit需要持续几微秒，传输信号是否在双向上同时进行，最初的连接如何创建和最终如何终止等问题。

#### TCP/IP参考模型

TCP/IP参考模型是一个抽象的分层模型，这个模型中，所有的TCP/IP系列网络协议都被归类到4个抽象的"层"中，如图0‑1所示。每一抽象层建立在低一层提供的服务上，并且为高一层提供服务。

**第4层应用层**

该层包括所有和应用程序协同工作，利用基础网络交换应用程序专用的数据的协议。常用的应用层协议有：HTTP、FTP、POP3、SMTP、TELNET、SSH、BOOTP、NTP、DNS、ECHO、SNMP、DHCP、ARP。

**第3层传输层**

传输层的协议，能够解决诸如端到端可靠性（“数据是否已经到达目的地？”）和保证数据按照正确的顺序到达这样的问题。在TCP/IP协议组中，传输协议也包括所给数据应该送给哪个应用程序。相关的协议如：TCP、UDP、DCCP、RTP。

**第2层网络互连层**

网络层解决在一个单一网络上传输数据包的问题。

对于TCP/IP来说这是因特网协议（IP）。（如ICMP和IGMP这样的必须协议尽管运行在IP上，也仍然可以看作是网络互连层的一部分；ARP不运行在IP上）

**第1层网络接口层**

网络接口层实际上并不是因特网协议组中的一部分，但是它是数据包从一个设备的网络层传输到另外一个设备的网络层的方法。

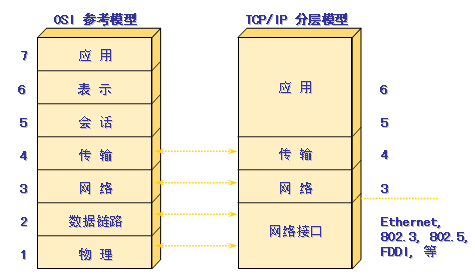


图9TCP/IP参考模型与OSI模型映射关系

### IPv4地址

IPv4地址是一个32位数字、通常使用点号分隔的四个十进制八位字节（取值范围从0到255）表示。该地址分为两个部分：网络位和主机位。网络位用来标识子网，具有相同的网络位的IP地址称为属于同一子网，同一子网中的所有主机可以在彼此之间直接通信，无需路由器。主机位用来标识子网中的特定主机，同一子网中每台主机的主机位必须是唯一的。

子网的大小是可变的。网络管理员通过指定子网掩码来指明多少位的IPv4地址属于同一子网。可供主机位使用的位数越多，子网中的可用的主机数就越多。

子网掩码可用两种格式表示。比如255.255.255.0和/24表示同样的值。

将子网中最低的地址（主机位二进制值全部为0）称为网络地址；子网中最高地址（主机位二进制值全部为1）用于广播消息，称为广播地址。

IP地址、子网掩码、网络地址及广播地址的关系见表0‑1，表0‑2和表0‑3：

表0‑1 192.168.1.107/24

|  |  |  |
| --- | --- | --- |
| 主机地址 | 192.168.1.107 | 11000000.10101000.00000001.01101011 |
| 子网掩码 | 255.255.255.0或/24 | 11111111.11111111.11111111.00000000 |
| 网络地址 | 192.168.1.0 | 11000000.10101000.00000001.00000000 |
| 广播地址 | 192.168.1.255 | 11000000.10101000.00000001.11111111 |

表0‑2172.16.0.123/16

|  |  |  |
| --- | --- | --- |
| 主机地址 | 172.16.0.123/16 | 10101100. 00010000.01111011 |
| 子网掩码 | 255.255.0.0或/16 | 11111111.11111111.00000000.00000000 |
| 网络地址 | 172.16.0.0 | 10101100. 00010000.00000000.00000000 |
| 广播地址 | 172.16.255.255 | 11000000.10101000.11111111.11111111 |

表0‑310.1.1.18/8

|  |  |  |
| --- | --- | --- |
| 主机地址 | 10.1.1.18/8 | 00001010.00000001.00000001.00010010 |
| 子网掩码 | 255.0.0.0或8 | 11111111.00000000.00000000.00000000 |
| 网络地址 | 10.0.0.0 | 00001010.00000000.00000000.00000000 |
| 广播地址 | 10.255.255.255 | 00001010.11111111.11111111.11111111 |

127.0.0.1/255.0.0.0这个特殊地址会始终执行本地系统（localhost），因此它可以使用网络协议进行自我通信。

#### IPv4地址分类

一开始，32位的IPv4地址只由8位的网络地址和“剩下的”主机位组成。这种格式用在局域网出现之前。这使得独立的网络的数量不能太多（最多254个），在那时，只有一些很少很大的的网络，例如ARPANET。在局域网出现的早期，就已经显得不够了。

为了和已存在的IP地址空间及IP数据报兼容，对IP地址的定义在1981年的RFC 791进行了修改。修改后的IP地址共有5类地址，见表0‑4和表0‑5。

表0‑4 IPv4地址分类

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 分类 | 前缀位 | 开始地址 | 结束地址 | 默认子网掩码 |
| A类地址 | 0 | 0.0.0.0 | 127.255.255.255 | 255.0.0.0 |
| B类地址 | 10 | 128.0.0.0 | 191.255.255.255 | 255.255.0.0 |
| C类地址 | 110 | 192.0.0.0 | 223.255.255.255 | 255.255.255.0 |
| D类地址 | 1110 | 224.0.0.0 | 239.255.255.255 | 未定义 |
| E类地址 | 1111 | 240.0.0.0 | 255.255.255.255 | 未定义 |

表0‑5 IPv4地址分类

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 分类 | 网络地址位数 | 剩余的位数 | 网络数 | 每个网络的主机数 |
| A类地址 | 8 | 24 | 128 | 16777214 |
| B类地址 | 16 | 16 | 16384 | 65534 |
| C类地址 | 24 | 8 | 2097152 | 254 |
| D类地址（[群播](http://zh.wikipedia.org/wiki/%E7%BE%A4%E6%92%AD)） | 未定义 | 未定义 | 未定义 | 未定义 |
| E类地址（保留） | 未定义 | 未定义 | 未定义 | 未定义 |

可用的主机地址总是2N - 2（N是所用的位数，减2是因为第一个和最后一个地址都是无效的）。因此，对于用8位来表示主机地址的C类地址来说，主机数就是254。

更多的网络位允许更多的网络，因此适应了互联网的持续增长。

有一些地址被设计为用作私有地址。在互联网的地址架构中，专用网络是指遵守RFC 1918和RFC 4193规范，使用私有IP地址空间的网络。私有IP无法直接连接互联网，需要公网IP转发。与公网IP相比，私有IP是免费的，也节省了IP地址资源，适合在局域网使用。私有IP地址在internet中不会被分配，私有地址的定义范围见表0‑6。

表0‑6 IPv4私有地址

|  |  |  |  |
| --- | --- | --- | --- |
| IP地址区段 | IP数量 | [分类网络](http://zh.wikipedia.org/wiki/%E5%88%86%E7%B1%BB%E7%BD%91%E7%BB%9C)说明 | 主机端位长 |
| 10.0.0.0 – 10.255.255.255 | 16777216 | 单个A类网络 | 24位 |
| 172.16.0.0 – 172.31.255.255 | 1048576 | 16个连续B类网络 | 20位 |
| 192.168.0.0 – 192.168.255.255 | 65536 | 256个连续C类网络 | 16位 |

#### 子网划分

在一个IP网络中划分子网使我们能将一个单一的大型网络分成若干个较小的网络。引入子网划分的概念，可以减少因特网路由表中的表项数量（通过隐藏一个站点内部所有独立子网的相关信息），以及减少网络开销（因为它将接收IP广播的区域划分成了若干部分）。

**示例：**将192.168.20.0/24这个C类地址子网掩码变成25位分成两个子网，如表0‑7所示。

表0‑7子网划分

|  |  |
| --- | --- |
| 子网掩码 | 11111111.11111111.11111111.10000000 (255.255.255.128) |
| 子网数 | 2^1=2 |
| 每个子网主机数 | 2^7-2=126 |
| 网络号 | 192.168.20.0 192.168.20.128 |
| 广播号 | 192.168.20.127 192.168.20.255 |

### IPv6地址

互联网通信协议第6版（英文：Internet Protocol version 6，缩写：IPv6）是互联网协议的最新版本，用于数据包交换互联网络的网络层协议，旨在解决IPv4地址枯竭问题。

IPv6地址采用了128位的地址，以16位为一组，每组以冒号":"隔开，可以分为8组，每组以4位十六进制方式表示（从0000到ffff）。例如：

1. 2001:0db8:85a3:08d3:1319:8a2e:0370:7344

由于IPv6地址很长，因此可以使用一些规则对它们进行简化。以下是简化规则：

* + 每项数字前导的0可以省略，省略后前导数字仍是0则继续；
  + 可以用双冒号"::"表示一组0或多组连续的0，但只能出现一次。

::1是IPv6版本的127.0.0.1 “localhost”地址。

### 网络路由和DNS概念

不管使用IPv4还是IPv6，网络通信都需要以主机到主机和网络到网络的形式进行传输。每个主机都有一个路由表，它用于告诉主机使用哪些网络接口与主机直接连接到的子网进行通信。如果网络通信未寻找到其中的一个子网，路由表通常会有针对所有其他网络的一个条目，该条目指向可访问子网中的路由器或网关。

如果路由器收到的通信并非将其作为寻址目标，则路由器不会忽略该通信，而是根据自己的路由表转发该通信。这种处理方式可能会将通信直接发送到目标主机（如果路由器恰巧与同一子网中），也可能转发到其他路由器。这种转发过程会一直进行，直到通信达到最终目标。

IP协议使用地址进行通信，但是比起冗长且难记的数字字符串，操作人员宁愿使用名称。DNS(即域名系统)是分布式服务器网络，可将主机名映射为IP地址。为使名称服务起作用，需要在主机系统中设置名称服务器的IP地址。该名称服务器无需与主机位于同一子网上，只需可供主机访问即可。

## 使用命令行配置服务器网络

在LINUX中配置联网最简单的方式是使用NetworkManager应用程序。它可以设置可影响所有用户的系统范围内的默认值；也可以配置仅在特定用户登录时，激活特定的网络接口。

1. **Debian网络连接基本配置**

**Debian对于网络连接管理和配置的两种方式**

当你在安装系统时候，如果开启了网络，一般情况下它已经被自动配置，可以检查在/etc/network/interfaces文件中已经包含一个可用的配置。这是传统的方式。在该配置文件中，所有以auto开始的网络接口将会在启动时被ifupdown和/etc/init.d/networking初始化脚本自动配置。

通常第一块网卡被称为eth0. 这个名称是由内核分配，以及udev规则来决定的（/etc/udev/rules.d/70-persistent-net.rules）。

NetworkManager作为一个新的替代推荐给移动工作的用户（比如在公司使用有线网络，而在家里使用无线wifi）。它方便用户在不同的地理位置不同的网络环境快速的切换网络连接。当选择安装了桌面环境时，该工具即默认被安装。可以在/etc/NetworkManager/system-connections/目录下创建连接的配置文件，或者使用图形化工具nm-connection-editor进行配置。NetworkMananger也可以做为默认的网络管理工具，记得将网络接口使用NM管理时，要禁用/etc/network/interfaces中的相关条目。

在服务器应用场景中，基于功能和稳定性的考量，官方仍然推荐使用传统的networking和interfaces的网络配置方式。

**配置IP地址**

手动修改配置文件：/etc/network/interfaces

**动态获取IP的方式：即dhcp**

1. auto eth0
2. iface eth0 inet dhcp

**静态IP的方式：即static**

1. auto eth0
2. iface eth0 inet static
3. address 192.168.0.3
4. netmask 255.255.255.0
5. gateway 192.168.0.1

其中auto eth0是设置系统开机启动时激活eth0；也可以使用 allow-hotplug eth0，表示内核探测到热插拔的网络设备即激活。

配置完成后，使网卡配置重新生效：ifdown eth0 && ifup eth0

**配置主机名**

使用hostname命令设置主机名，立即生效，永久保留要写入到/etc/hostname。

查看主机名是否生效可以用hostname直接查看，或者cat /proc/sys/kernel/hostname 。

**配置主机的域名**

域名通常不是由主机自己决定，而是DNS服务器进行指向解析。但是在本地/etc/hosts文件中最好记录本地地址的对应名称，如下格式：

1. 127.0.0.1 localhost
2. 127.0.0.1 server01.test.com server01

在/etc/nsswitch.conf文件中

1. host files dns

定义了系统通过域名查找IP的顺序是先查询hosts文件，再查询dns。

**DNS服务器的配置**

手动编辑/etc/resolv.conf

1. nameserver 223.5.5.5
2. NOTE
3. When the resolvconf package is not installed, DNS related configuration needs to be done manually by editing the “/etc/resolv.conf” as the following.

在一个网卡上配置多个IP：虚拟接口

1. iface eth0 inet dhcp
2. metric 0
3. iface eth0:0 inet static
4. address 192.168.0.1
5. netmask 255.255.255.0
6. network 192.168.0.0
7. metric 1
8. NOTE
9. metric IPv4路由度量操作工具 metric越大，越耗费代价，越不值钱。例如，如果dhcp获得的也是192.168.0.0网段的ip地址，那么要发送数据包到192.168.0.0网段，则是将数据包通过eth0发出去，这可以用于改变在接口的IPv4流量路由优先级。较低的度量有更高的优先级。

## 使用命令行配置IP

可以使用ifconfig route ip 等命令，配置后立即生效，重启系统（重启网络服务）后命令行设置失效（除非持久到写入到interfaces配置文件中）， 推荐使用ip命令（新一代的网络配置工具）。 以下是部分常用的ip命令用法：

6.1. 查看网卡ip 信息

1. # ip addr show
2. # ip addr show eth0

6.2 查看网络连接状态

1. # ip link show

6.3. 添加和删除IP

1. # ip addr add 192.168.139.168/24 dev eth0
2. # ip addr del 192.168.139.129/24 dev eth0

6.4. 路由操作

1. # ip route show
2. # ip route del default dev eth0
3. # ip route add to default via 192.168.139.2 dev eth0
4. # ip route add to 192.168.10.0/24 via 192.168.139.123 dev eth0

7. 额外的有用的命令

监测物理网卡的连接状态，需要网卡支持。

1. # mii-tool -w eth0

下面这个命令可以使用eth0上的LED闪烁30s

1. # ethtool -p eth0 30

还有一些实用的命令，比如：dhclient netstat nmap tcpdump wireshark nc ，请自行查阅相关的用途和用法。

## 远程管理Linux系统初探

当Linux主机建立好网络连接后，我们便可以远程从网络访问和管理系统。SSH是最通用的远程系统管理工具之一。它允许用户登录远程系统并在其上执行命令。SSH使用加密技术在网络传输数据，具有很高的安全性。在绝大多数Linux系统中，SSH是标准配置，并且默认启动，并允许访问（防火墙默认开放相应端口）。

在本章节我们简单介绍LINUX系统中OpenSSH的应用，方便大家学习过程中使用。关于OpenSSH服务的配置和其他Linux系统远程管理的工具将在后面介绍。

LINUX系统中ssh服务默认启动，并且无需配置即可使用。将LINUX主机作为服务器端，在网络连接畅通的情况下，可以使用ssh客户端工具很方便的连接到LINUX主机。

**ssh客户端**

根据不同的系统环境，有多种ssh客户端软件，而且大多数是开源软件。

* + Linux 系统自带客户端；
  + windows系统需要下载（安装）ssh客户端。

**Linux系统下的ssh客户端**

Linux系统下的ssh客户端即默认自带的ssh命令。

**示例：**使用ssh命令连接Linux主机

1. [demo@localhost ~]$ ssh zhangsan@192.168.148.131
2. The authenticity of host '192.168.148.131 (192.168.148.131)' can't be established.
3. RSA key fingerprint is 2b:86:16:de:86:31:91:0d:ae:0a:a3:bd:59:23:06:76.
4. Are you sure you want to continue connecting (yes/no)? yes
5. Warning: Permanently added '192.168.148.131' (RSA) to the list of known hosts.
6. reverse mapping checking getaddrinfo for bogon [192.168.148.131] failed - POSSIBLE BREAK-IN ATTEMPT!
7. zhangsan@192.168.148.131's password:
8. [zhangsan@365linux ~]$

以上命令使用zhangsan用户身份连接登录192.168.148.131这台主机。zhangsan是远程服务器上的用户。如果在连接时不指定用户（即ssh命令后面直接跟IP地址），则ssh命令使用当前执行命令的用户身份尝试登录远程主机，这需要远程主机上必须存在和本地客户端主机相同的用户。

第一次建立连接时，需要进行身份确认。在询问是否继续时输入yes。

输入远程主机将登录用户的密码，即可成功登录。

登录成功后，当前终端环境变成远程主机的终端环境。

**示例：**退出远程主机登录

1. [zhangsan@365linux ~]$ exit
2. logout
3. Connection to 192.168.148.131 closed.

**Windows系统下的ssh客户端**

Windows系统下ssh客户端推荐：

**putty**：开源软件，免费使用，软件小巧，免安装方便携带；

**xshell：**商业软件，对学校、家庭使用免费，功能强大。

**工作技巧**

所有使用的服务器远程管理客户端软件，首先选用开源软件，且软件一定在官方网站上（建议使用google进行搜索）下载原版（禁止使用所谓的汉化版或者破解版）。尽可能对下载的软件进行MD5/SHA1签名验证。一旦使用来路不明可能含有木马后门的ssh客户端软件，将对服务器系统造成巨大的安全威胁。

**示例：**使用putty远程连接Linux系统

1. 到putty官方网站http://www.putty.org/下载putty软件，如图0‑2所示，并进行签名验证。



图10putty软件图标

无需安装，直接运行。填写IP和端口（默认22），点击“Open”，打开对话框如图0‑3所示。

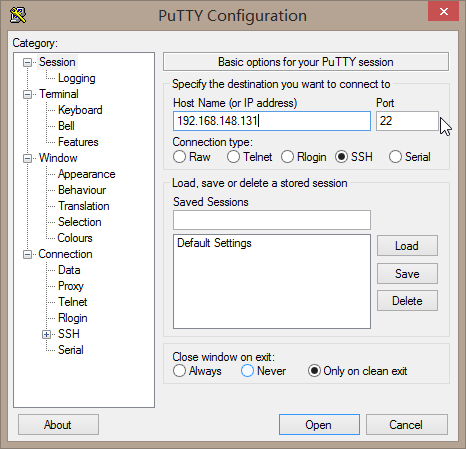


图11启动putty

输入要登录的用户名，比如我这里用root用户登录；然后输入用户密码（输入密码过程中，密码不回显在屏幕上），如图0‑4所示，按回车确定。



图12 putty登录过程

登录成功，效果如图0‑5所示。

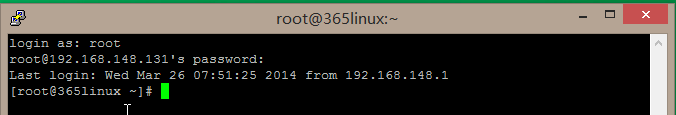


图13登录成功

**工作技巧**

在生产环境系统中，为安全起见，ssh服务器一般配置为禁止root用户远程登录。正确的做法是使用一个普通用户登录到系统，然后需要root权限时再使用su或者sudo切换到管理员。

Linux系统广泛应用于服务器领域，每一个服务的提供都依赖于网络通信，所以为系统建立网络连接是对系统进行配置的第一步。对于单台主机的网络配置相对简单，而当服务器融入到复杂的生产环境的网络中时，对于主机网络的配置，更多的是要考虑网络的规划以及如何在复杂的网络中进行合理的连接，以在实现网络通信时保证网络的安全。这些都需要基本（甚至深入）的网络相关知识作为支持。对于非专职从事网络管理的系统工程师而言，至少要理解网络架构的设计以及主机在该网络中所处的位置，当出现网络通信问题时，能够判断故障是发生在网络层面还是系统层面。

# 软件安装

**【任务描述】**

在使用桌面操作系统时，因为工作生活的需要，需要在系统中安装各种软件包。同样的Linux操作系统不管是在作为桌面端使用，还是作为服务器系统用于生产环境，都有大量的软件包可以选择性安装。需要理解的是，不同操作系统平台的软件安装包一般是不能通用的。Linux由于开源特性，操作系统版本众多，不同版本的操作系统软件包管理方式各不相同。

**【学习目标】**

1. 了解Linux系统不同的软件包安装和管理方式；
2. 使用dpkg命令对deb包进行安装、查询、检验、卸载；
3. apt的配置使用和软件仓库的搭建。

## 认识Linux系统软件包

**DPKG**

Debian的软件包安装管理工具。它和RPM一样功能强大。软件包一般是以deb包的形式发布。

**APT**

Advanced Package Tool，是一种和yum类似的在线的软件安装管理机制，功能强大，而且免费。APT源自Debian，但是它已经扩展到可以支持RPM。

**RPM**

RPM是RedHatR软件包管理器。rpm包是一种预先已经编译好的二进制文件，安装rpm包的优点在于安装方便，不需要重新编译源代码；rpm在安装的过程中会记录软件包的安装信息到RPM数据库，方便查询、升级和卸载。但同时也带来一定的缺点，比如安装环境要与打包时的环境保持一致；需要满足安装套件的依赖关系。

**YUM**

rpm在安装过程中最大的问题就是要解决依赖关系。yum的出现克服了这一问题。yum是一种基于rpm的智能交互式的软件管理机制。使用yum安装软件时，会通过分析rpm软件包的表头数据，从yum server上下载文件进行安装。

yum功能强大，可以用来安装、查询、升级和卸载软件包，还能更新系统，甚至搭建本地的yum软件仓库。

**Tarball**

软件源码打包发布的一种形式。在安装的时候需要参考它提供的安装说明文档，进行编译安装。优点是灵活性强，定制自由度高。缺点是安装过程稍繁琐。

## 在Debian9的系统上安装和管理软件包

### 使用命令行安装和管理软件包

dpkg应被视为一个系统工具（后端），apt-get 作为一个贴近用户的工具，克服了前者的缺陷。这些工具一起工作，每一个具有其特殊性，适合特定的任务。

apt-get 所有的命令行操作，包括软件包的安装，删除，和版本更新；Advanced Packaging Tool (APT), front-end for dpkg providing "http", "ftp", and "file" archive access methods (apt-get/apt-cache commands included)

aptitude 一个交互式文本接口，用来管理安装的软件包和查找可用的软件包。

deb包的分类和数量：

| **area** | **number of packages** | **criteria of package component** |
| --- | --- | --- |
| main | 40500 | DFSG compliant and no dependency to non-free |
| contrib | 248 | DFSG compliant but having dependency to non-free |
| non-free | 526 | not DFSG compliant |

通常使用apt-get来安装软件而不使用dpkg，apt使用本地源或者网络源。dpkg用来查询。

举例：163的源

1. deb http://mirrors.163.com/debian/ wheezy main non-free contrib
2. deb http://mirrors.163.com/debian/ wheezy-proposed-updates main non-free contrib
3. deb-src http://mirrors.163.com/debian/ wheezy main non-free contrib
4. deb-src http://mirrors.163.com/debian/ wheezy-proposed-updates main non-free contrib

查询系统中安装的以b开头的软件包：

1. [root@365linux ~]# dpkg -l 'b\*'

查询系统中是否安装了vim的软件包：

1. [root@365linux ~]# dpkg -l |grep vim

查询软件包安装了哪些文件到系统：

1. [root@365linux ~]# dpkg -L vim

查询匹配文件名的文件提供的软件包：

1. [root@365linux ~]# dpkg -S date
2. [root@365linux ~]# dpkg -S /bin/date
3. coreutils: /bin/date

apt 常用指令：

apt-get

update upgrade install remove purge dist-upgrade autoremove clean autoclean

apt-cache show

apt-cache search

dpkg是Debian开发并推广的一种强大的软件包管理工具，它通过在软件包封装时加入软件的安装信息，依赖关系，安装前后程序，在安装时记录安装数据库，来达到deb软件包的安装、升级、查询、检验，卸载的功能。deb包安装方便，快速，也便于管理。在版本合适情况下，可优先选择。deb的缺点是对安装环境、系统版要求其编译打包环境基本一致才能成功安装，且安装时灵活性和定制自由度不够。

Tarball源码包优点在于定制性强，通过配置时参数可以定制软件功能和其他软件进行整合。由软件安装完后文件分布在各个目录，源码包卸载起来比较麻烦，也不便于查询管理。若在安装时，自定义了安装目录，则卸载时，只需要将安装目录完全删除即可。

apt最常用的软件包管理工具。它本身是一种先进的软件包管理工具。通过在线的软件仓库，改变了原始的软件包安装方式需要手动解决依赖关系的困难。apt安装的软件同样可以通过dpkg进行查询管理和卸载。

# 磁盘管理

**【任务描述】**

在安装Linux系统时，根据服务器的用途，需要对硬盘进行合理分区规划。在系统运行的过程中，随着业务的发展变化，不断增长的数据存储的要求，也需要对磁盘存储进行调整和管理。目前在服务器上使用得最多，技术最成熟可靠的仍然是传统的机械硬盘。随着技术的发展，在相当多的领域也使用了基于闪存技术的SSD硬盘。在这一章节，对于硬盘的物理组成介绍主要是指的机械硬盘。

**【学习目标】**

* 1. 理解分区和文件系统的基本概念；
  2. 对磁盘分区、格式化、挂载和卸载。

## 认识分区和文件系统的基本概念

硬盘是由多个盘片组成，按照硬盘片能够容纳的数据量，分为单盘（一块硬盘里面只有一个硬盘片）和多盘（一块硬盘里面有多个硬盘片）。

硬盘里有磁头（head）在磁盘上读写，磁头固定在机械手臂上，机械手臂上有多个磁头，可以进行读写。当磁头固定不动时，硬盘片转一圈所画出来的圆就是磁道（track）。如前所述，一块硬盘可能有多个盘片，所有硬盘上相同半径的磁道就组成了柱面（cylinder）。在旧的系统的版本中，柱面是分区时的最小计量单位。

由圆心向外画直线，可以将磁道再细分为扇区，扇区就是硬盘上的最小存储物理量。通常，一个扇区的大小约为512字节。而在新的LINUX的版本中，分区的最小计量单位是扇区。

当硬盘读取数据时，硬盘会转动，利用机械手臂将磁头单方向的来回移动定位到正确的数据位置，将数据准确读出。硬盘的磁头与硬盘片的接触空间很细微，如果有抖动或者是有脏污，就会造成数据或物理硬盘的损坏。所以在使用计算机的过程中，不要移动主机，以避免硬盘抖动。关机时使用正确的关机方式而不是直接关闭电源，因为正确关机过程中，系统会使硬盘的机械手臂回归原位。

目前新兴的固态硬盘在物理构造上与传统的机械硬盘完全不同，用电子芯片存储取代了磁介质存储，因此在SSD固态硬盘中没有磁头的机械运动和盘片的转动。

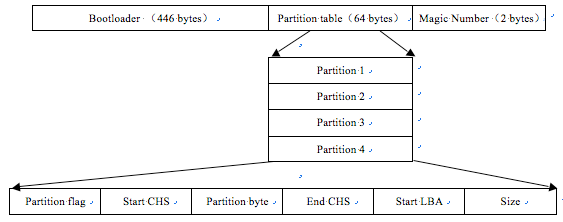
**分区和文件系统的基本概念**

在开始对硬盘分区之前，必须要了解一下根分区和文件系统相关的一些基本概念。

1. **MBR**

主引导记录（Master BootRecord，缩写：MBR），是计算机系统开机后访问硬盘时所必须要读取的首个扇区，它在硬盘上的三维地址为0柱面0磁头1扇区。所以MBR的容量大小就是一个扇区的大小，512字节。而其中开头的446字节内容为系统的“主引导记录”（MBR）信息，为避免混淆，称为BootLoader；其后的64字节被分成4个16字节的“磁盘分区表”（DPT），即Disk PartitonTable；以及2字节的结束标志（55AA）。

MBR的组成如**图0‑1**所示。



**图14MBR的组成**

很显然，跟硬盘分区有关的就是Partition table，它的4个16字节的固定设计决定了一个硬盘最多只能够分出4个主分区，如果需要5个以上的分区，那就需要将其中一个主分区变为扩展分区，在扩展分区的基础上再建立逻辑分区。

1. **文件系统**

硬盘当然是用来存储数据，即存放系统和用户的文件，而分区的文件系统就规定了数据存储单元在硬盘上排列的规则。不同的文件系统，存储数据的方式不一样，这些不同的文件系统可以共存于同一个系统的硬盘之上，但不能共存于同一个磁盘分区之上。Linux最重要的特性之一就是支持多种文件系统。而Linux通过VFS（Virtual File System）支持多个不同的文件系统，能够与不同的操作环境实现资源共享。

Linux支持的文件系统包括：

ext2 minix MS-DOS FAT iso9660

ext3 ext4 ReiserFS NTFS JFS XFS

NFS SMBFS

**示例：**可以通过以下命令查看当前Linux系统支持的文件系统。

1. [root@365linux~]#ls -l /lib/modules/`uname -r`/kernel/fs

**示例：**当前系统启用的文件系统。

1. [root@365linux~]# cat /proc/filesystems

默认情况下，LINUX采用ext4文件系统。

## 建立和使用磁盘分区

### 使用fdisk命令进行分区操作

1. **查看现有的系统磁盘分区。**
2. [root@365linux ~]#fdisk -l
3. Disk /dev/sda: 21.5 GB, 21474836480 bytes
4. 255 heads, 63 sectors/track, 2610 cylinders, total 41943040 sectors
5. Units = sectors of 1 \* 512 = 512 bytes
6. Sector size (logical/physical): 512 bytes / 512 bytes
7. I/O size (minimum/optimal): 512 bytes / 512 bytes
8. Disk identifier: 0x000caf05
9. Device Boot Start End Blocks Id System
10. /dev/sda1 \* 2048 1026047 512000 83 Linux
11. /dev/sda2 1026048 9414655 4194304 82 Linux swap / Solaris
12. /dev/sda3 9414656 41943039 16264192 83 Linux
13. Disk /dev/sdb: 10.7 GB, 10737418240 bytes
14. 255 heads, 63 sectors/track, 1305 cylinders, total 20971520 sectors
15. Units = sectors of 1 \* 512 = 512 bytes
16. Sector size (logical/physical): 512 bytes / 512 bytes
17. I/O size (minimum/optimal): 512 bytes / 512 bytes
18. Disk identifier: 0x00000000

查看整个系统内的设备分区列表。通过分析列出的信息，我们可以了解到该硬盘的使用情况。如上例中所示的，该系统中存在两块硬盘，名称分别为sda和sdb，其中sda是在安装系统时进行了规划，从命令输出的结果数据表明，它的容量大小为21.5GB，总共有41943040个扇区；sdb的容量大小是10.7 GB，20971520个扇区。

sda使用情况是被分成3个分区，sda1, sda2, sda3，从分区的起始扇区可以看到三个分区连续排列，且最一个分区sda3的最后结束扇区为41943039，接近于硬盘的扇区总数，说明该硬盘已没有剩余的空闲空间可以进行分区操作。

sdb则是一个没有任何分区的完整的空闲的硬盘。接下来使用fdisk的命令对sdb进行分区的操作。

1. 分区操作。使用fdisk命令进入交互分区模式。
2. [root@365linux ~]#fdisk /dev/sdb
3. Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
4. Building a new DOS disklabel with disk identifier 0x1dfc88be.
5. Changes will remain in memory only, until you decide to write them.
6. After that, of course, the previous content won't be recoverable.
7. Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)
8. Command (m for help): n //键入n创建新的分区，可以通过键入m获得命令的帮助
9. Command action
10. e extended
11. p primary partition (1-4)
12. p //键入p选择创建主分区
13. Partition number (1-4): 1 //分区编号为1
14. First sector (2048-20971519, default 2048): //起始扇区，不填默认为2048
15. Using default value 2048
16. Last sector, +sectors or +size{K,M,G} (2048-20971519, default 20971519): +2G //分区结束扇区，可以使用扇区数，也可以使用单位大小，比如+2G即表示分2G大小的分区，比较直观。
17. Command (m for help): p //分完区后输出当前的分区表查看sdb的分区情况
18. Disk /dev/sdb: 10.7 GB, 10737418240 bytes
19. 255 heads, 63 sectors/track, 1305 cylinders, total 20971520 sectors
20. Units = sectors of 1 \* 512 = 512 bytes
21. Sector size (logical/physical): 512 bytes / 512 bytes
22. I/O size (minimum/optimal): 512 bytes / 512 bytes
23. Disk identifier: 0x1dfc88be
24. Device Boot Start End Blocks Id System
25. /dev/sdb1 2048 4196351 2097152 83 Linux
26. Command (m for help): w //键入w保存分区表到MBR的第二区域DPT。
27. The partition table has been altered!
28. Calling ioctl() to re-read partition table.
29. Syncing disks.

使用fdisk对分区进行新建、修改、删除的操作，在键入w保存命令之前都是没有写入分区表的，随时可以撤销，或者键入q不保存退出。

**NOTE**

fdisk命令工具创建的DOS格式的分区表，这是一种最常见而且最大程度兼容别的操作系统的分区表，但因为历史的原因DOS格式的分区表最大只能支持2TB的分区大小。

**工作技巧**

在生产环境中，通常情况下不允许对正在使用的硬盘进行分区操作，比如对已经安装并正在运行的操作系统的硬盘进行分区是不允许的。即使在特殊的情况下必须对正在使用的硬盘进行分区，分区调整后保存到分区表，这时正在运行的系统内核并不会识别新的分区表，也就是说调整后的分区不会被系统识别。必须进行重启或者执行（partprobe或者partx）强制使用新的分区表生效。

1. **创建文件系统**

**示例：**将/dev/sdb1格式化为ext4的文件系统。

1. [root@365linux ~]# mkfs.ext4 /dev/sdb1
2. **挂载和使用磁盘分区**

**示例：**将/dev/sdb1挂载自定义的/data目录。

1. [root@365linux ~]#mkdir /data
2. [root@365linux ~]# mount /dev/sdb1 /data

要使用Linux分区，在格式化完后，还需要对分区进行挂载（mount）。所谓的挂载，就是将需要使用的分区载入到某一个目录。该目录就称为挂载点。挂载点必须要提前存在，挂载成功后，在挂载点目录下创建的所有文件实际上就是存储在挂载到该目录下对应的分区上。

**NOTE**

若挂载点本身非空，载入分区后，原目录下的文件会暂时“消失”（不可访问），被挂载上来的分区上的内容所取代，等到该分区被卸载后，原目录中的文件就会再次显示。一般不会使用非空的目录做为常用的挂载点。

**示例：**查看分区挂载情况

1. [root@365linux ~]#df -h
2. FilesystemSize Used Avail Use% Mounted on
3. /dev/sda3 16G 626M 14G 5% /
4. tmpfs 947M 0 947M 0% /dev/shm
5. /dev/sda1 477M 23M 429M 6% /boot
6. /dev/sdb1 2.0G 3.0M 1.9G 1% /data

使用mount命令挂载分区到挂载点，是即生效，但是系统重启后不会自动挂载，如果需要在系统每次开机都自动将指定的分区挂载到某个特定的目录，则需要将挂载信息写入系统的文件系统列表配置文件。

**示例：**/dev/sdb1开机自动挂载到/data目录。

1. [root@365linux ~]# vim /etc/fstab
2. ……
3. /dev/sdb1 /data ext4 defaults 0 0

在fstab配置文件最后添加一行挂载记录。fstab文件格式分为六列，每一列的内容含意如下：

1. 设备名或者设备的卷标
2. 挂载点
3. 分区的文件系统类型
4. 文件系统的挂载参数
5. 是否使用dump命令备份，0表示不要做dump备份，1表示进行dump备份，2表示备份，但重要性比1小
6. 是否使用fsck命令检测，0表示不要检验，1是要检验，2是要检验，不过优先级小于1

**工作技巧**

在fstab中分区的设备名称使用/dev/sdb1是不推荐的，因为如果以后对该分区所在的硬盘进行分区调整，其分区号数字可能会发生变化。所以更推荐使用LABLE或者UUID的格式。

1. 卸载分区

如果是临时挂载使用某个分区，在使用完成后，可以将其卸载。卸载之后，分区上的数据将无法访问。

**示例：**通过分区设备名称卸载。

1. [root@365linux ~]#umount /dev/sdb1

**示例：**通过挂载点卸载。

1. [root@365linux ~]#umount /data

**NOTE**

特别注意的是，在卸载分区之前，先要确保该分区数据未被占用使用，包括当前用户退出其挂载点目录。

在Linux系统管理的工作中，对于存储设备的管理是日常操作之一。在本章节我们了解了磁盘存储的物理结构构成，理解分区的实质存在，分区的意义和具体操作。因为Linux的基于文件的设计理念，所以对于分区的访问，是需要通过一个目录做为分区对应的入口，这个目录即被称为挂载点。这一点必须要深刻理解，在工作中，灵活运用这样的特性，可以发掘出实际使用的很多技巧，比如数据的迁移。

Linux对多种文件系统都很好的支持，而且在使用上保持了一致性，限于篇幅和难度设计，这里并没有展开讨论。比如，对于FAT或NTFS文件系统的支持，对于远程文件系统的挂载使用，SWAP交换分区的调整等。

# 计划任务

**【任务描述】**

人在工作生活中，经常制定自己的计划任务，比如计划到了某个时间点外出，或者计划好每周一的10:00召开会议。操作系统在运行时也需要制定各种计划任务，让它定时自动运行，比如定时关机，周期自动清理系统等。计划任务一般分为两种情况：在未来某个时间点执行的一次性计划任务；在未来某个时间点周期性循环执行的计划任务。

**【学习目标】**

* 1. 一次性计划任务at的使用；
  2. 周期性计划任务cron的使用

## 创建一次性的计划任务

有些计划任务是只需要执行一次就完成了的。这种情况Linux系统使用atd这个服务来进行任务分配。使用at命令创建和管理这类型一次性的计划任务。

**示例：**在2015年3月23日执行命令wall “hello”。

1. [root@365linux ~]# at 16:50 03232015
2. at> wall "hello"
3. at> <EOT>

任务指令输入完后按Ctrl+D存盘退出。

at命令指定时间的书写格式非常灵活，如：

at 5:30pm

at now +5minutes

at 17:30 tomorrow

更多at支持的时间格式，可以通过man at来查看。

**示例：**查询当前等待执行的任务。

1. [root@365linux ~]# atq
2. 3 2010-03-23 16:50 a root

**示例：**删除一个等待的任务。

1. [root@365linux ~]# atrm 3

任务编号3即通过atq查询得到的任务编号。

**NOTE**

默认情况下，任何用户都可以使用at服务，但这样对于系统不是很安全。可以通过两个文件/etc/at.allow，/etc/at.deny来进行白名单和黑名单的控制。

在at.allow中记录的用户才可以使用at服务。

在at.deny中记录的用户不可以使用at服务。

一般来说，系统只需要二者中的一个记录文件即可。

**示例：**查看系统中保存计划任务的文件

1. [root@365linux ~]# ls /var/spool/at/
2. a000040142cf1a

at的计划任务记录文件默认放在/var/spool/at目录下，当at命令创建一个任务后，会在该目录下建立一个文件，该文件包含了将被执行的命令。如果将文件删除，那么对应的计划任务也会消失。

## 在系统中创建周期性执行的计划任务

如果计划任务需要循环来执行，那么就需要crond这个服务来控制。系统有很多工作是需要计划定时进行的，比如日志轮替、一些系统数据库的定期更新等等，所以这个服务默认是启动的。

除了系统的计划任务，我们可以通过crontab命令来自定义用户的计划任务。

**示例：**root用户创建一个计划任务，每天凌晨3点整执行脚本/root/test.sh。

1. [root@365linux ~]# crontab -e
2. 0 3 \* \* \* /root/test.sh

使用crontab -e进入计划任务的编辑界面，使用vi工具对任务的时间和要执行的命令进行书写或修改。五个时间参数的含义如表表格 12‑1所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 域 | 分钟 | 小时 | 日期 | 月份 | 周 |
| 范围 | 0-59 | 0-23 | 1-31 | 1-12 | 0-7 |

表格 12‑1 时间参数

另外范围取值还有一特殊字符的用法。

“\*”表示任何时间都接受；

“，”表示分隔的时段都适用；如1，3，5表示某一个域内1，3，5都适用。

“-”表示一段时间范围都适用；如2-5表示2、3、4、5都适用。

“/”表示每隔多少时间；如在分钟域的\*/5表示每隔5分钟。

**示例：**列出当前用户的周期计划任务列表

1. [root@365linux ~]# crontab -l
2. 0 3 \* \* \* /root/test.sh

**示例：**删除当前用户所有的计划任务。

1. [root@365linux ~]# crontab –r

如果只是要删除部分，则使用crontab -e 编辑即可。

**示例：**查看系统中保存的用户周期性计划任务的文件

1. [root@365linux cron]# ls /var/spool/cron/
2. root

cron计划任务的记录文件是放在/var/spool/cron目录下。

**工作技巧**

当用户在系统中创建多个cron周期性计划任务时，每个任务执行的时间点应该不同并保持一定的时间间隔，比如每天执行的任务放在02:10，则每周执行的任务放在03:20，每月执行的任务则放在04:10。这样避免在某些特定日子，任务集中在一起执行消耗过多的系统资源。另外，周期性计划任务一般是安排在系统负载较低的时候自动运行，比如凌晨。

系统级需要定时运行的任务，在系统/etc/crontab文件中进行定义。

**示例：**查看/etc/crontab配置文件内容

1. [root@365linux ~]# vim /etc/crontab
2. SHELL=/bin/bash
3. PATH=/sbin:/bin:/usr/sbin:/usr/bin
4. **M**AILTO=root
5. HOME=/
6. # For details see man 4 crontabs
7. # Example of job definition:
8. # .---------------- minute (0 - 59)
9. # | .------------- hour (0 - 23)
10. # | | .---------- day of month (1 - 31)
11. # | | | .------- month (1 - 12) OR jan,feb,mar,apr ...
12. # | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
13. # | | | | |
14. # \* \* \* \* \* user-name command to be executed

默认情况下，该配置文件中并没有定义任何的周期性计划任务。用户可以添加系统级别的计划任务到该配置文件中，使用某个时间点，以某个用户的身份，执行什么任务这样的格式。

系统运行必要的计划任务事件在/etc/cron.d中定义和调用。

用户可以定制自己的计划任务。系统计划任务只有管理员可以编辑。

系统通过/etc/cron.allow或/etc/cron.deny来对用户使用计划任务进行白名单或黑名单的控制。

计划任务是非常实用功能，系统管理员常用来利用计划任务进行定期检查，数据备份，发送通知等，通过编写脚本，加入到计划任务定期自动执行。为了保证计划任务正常被执行，尤其是系统中存在必要的cron周期性系统任务，在系统中，必要保持atd和crond这两个服务开机自动运行。

在编辑计划任务的时候，我们需要注意三个问题：

1. 避免不必要的屏幕输出信息发送；
2. 多个任务要时间交错执行；
3. 命令使用绝对路径。

**工作技巧**

在系统中计划任务是定时自动运行的，所以计划任务也是系统被入侵后自动执行恶意脚本的手段之一。

# 状态检测

**【任务描述】**

系统管理员要定期对运行中的系统进行健康检查，通过系统进程的状态，进程对系统资源的占用，分析系统的运行状况，解决故障或排除潜在问题。

**【学习目标】**

* 1. 系统进程的控制；
  2. 系统状态检测；
  3. 系统内核信息查看与修改。

## 查看和管理系统进程

1. **进程的概念**

进程是Linux用来表示正在运行的程序的一种抽象的概念。Linux系统上所有运行的东西都可以称之为进程。包括用户任务、批处理进程和守护进程。进程的一个比较正式的定义是：在自身的虚拟地址空间运行的一个单独的程序。

进程一般来说由程序产生，但不等同于程序。进程是一个动态的概念。一个程序可以启动多个进程。

运行一个程序则触发一个事件，这个事件被称为作业。在bash的环境下，可以对作业进行管理，比如将作业放置到后台，或前台，暂停或程序启动。

系统最开始的进程是init。init的PID（进程号）始终为1。

一个进程可以产生另外一个进程。如在登录的bash下执行cat命令。则cat就是bash的子进程。通过pstree，或ps -ely（比较PID跟PPID）可以清楚的知道进程之间的关系。除了init这个进程外，所有的进程都有父进程。当一个进程不正常时，可以通过追踪它的父进程来确定进程的作用及采取适当的措施。如果父进程在子进程终止之前消亡，那么这个“孤儿”将会被init“收养”。

一般来说，子进程会继承父进程的权限。进程的UID就是其创建者的UID号。那么，系统只允许进程所有者和root对其进行操作。同时EUID规定了进程对哪些资源和文件有访问的权限。

**示例：**查看进程状态

1. [root@365linux ~]# ps aux
2. USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
3. root 1 0.0 0.1 2064 624 ? Ss Mar03 0:01 init [3]
4. root 2 0.0 0.0 0 0 ? S< Mar03 0:00 [migration/0]
5. root 3 0.0 0.0 0 0 ? SN Mar03 0:00 [ksoftirqd/0]
6. root 4 0.0 0.0 0 0 ? S< Mar03 0:00 [watchdog/0]
7. root 5 0.0 0.0 0 0 ? S< Mar03 0:00 [events/0]

ps用来查看进程信息，功能强大，用法也比较复杂，最常用用的参数有：ps aux ，ps -ef ，ps -ely

**选项说明如下：**

a：显示所有进程，但不包括不属于任何终端的进程。

u：显示进程拥有者、状态、资源占用等的详细信息（注意有“-”和无“-”的区别）。

x：显示没有控制终端的进程。

-e：显示所有进程。

-f：显示全格式。

-l：显示长格式。

-y：与-l配合使用。

**输出中每列的意义：**

USER：进程拥有者，即创建者；

PID：进程ID号；

%CUP：占用CPU的百分比；

%MEM：占用内存的百分比；

VSZ：占用虚拟内存的大小（KB）；

RSS：占用物理内存的大小（KB）；

TTY：当前进程执行终端号；

STAT：该进程的状态：D不可中断的睡眠状态，除非发生指定事件，否则不会被唤醒；R正在执行中；S睡眠状态，可被某些信号唤醒；T暂停执行；Z僵尸进程，例如已终止但未能被父进程回收的子进程； N低优先序进程；<：高优先级进程；L有内存分页分配并锁在内存中；s：一个会话的前导字符；l：多线程；+：前台进程。

START：进程开始时间；

TIME：进程实际使用CPU运行的时间；

COMMAND：进程实际命令。

**查看进程的其他常用命令：**

pstree是以进程树的形式显示进程之间的父子关系。

pgrep是以进程名字或属性（如进程UID）来显示查找进程的PID。

**示例：**查看系统即时运行状态（资源管理器）

1. [root@365linux ~]# top
2. top - 01:44:09 up 5:17, 3 users, load average: 0.00, 0.00, 0.00
3. Tasks: 65 total, 1 running, 64 sleeping, 0 stopped, 0 zombie
4. Cpu(s): 0.0%us, 0.0%sy, 0.0%ni, 99.7%id, 0.0%wa, 0.0%hi, 0.3%si, 0.0%st
5. Mem: 515444k total, 435564k used, 79880k free, 58088k buffers
6. Swap: 1052248k total, 0k used, 1052248k free, 327472k cached
7. PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
8. 1 root 15 0 2064 624 536 S 0.0 0.1 0:01.19 init
9. 2 root RT -5 0 0 0 S 0.0 0.0 0:00.00 migration/0
10. 3 root 34 19 0 0 0 S 0.0 0.0 0:00.00 ksoftirqd/0
11. 4 root RT -5 0 0 0 S 0.0 0.0 0:00.00 watchdog/0

top命令是Linux下常用的性能分析工具，能够实时的显示系统资源占用状况。

top命令显示结果上半部分是系统整体的统计信息，分别表示：

当前时间是01:44:09，系统运行时间5小时17分，当前有3个用户登录，系统负载：1分钟0.00，5分钟0.00，15分钟0.00。

进程总数65个，1个在运行，64个在睡觉，0个暂停，0个僵死。

CPU占用百分比，用户空间占用0.0%，内核空间占用0.0%，用户进程空间内改变过优先级的进程占用CPU的百分比为0.0%，CPU空闲99.7%，等得输入输出的CPU时间为0.0%。

物理内存大小是512444KB，435564K被使用了，79880K空闲，58088K用作缓存。

交换空间的大小是1052248K，0K被使用了，1052248K空闲，缓冲交换区的大小是327472K。

top命令显示结果的下半部分是各进程的详细信息。每列的含义如表格 13‑1所示：

|  |  |  |
| --- | --- | --- |
| **序号** | **列名** | **含义** |
| A | PID | 进程PID号 |
| E | USER | 进程拥有者的用户名 |
| H | PR | 优先级 |
| I | NI | NICE值，负值表示高优先级，正值表示低优先级。 |
| O | VIRT | 进程使用的虚拟内存总量，单位KB。VIRT=SWAP+RES |
| Q | RES | 进程使用的、未被换出的物理内存大小。单位KB。RES=CODE+DATA |
| T | SHR | 共享内存的大小，单位KB。 |
| W | S | 进程状态。 |
| K | %CPU | 上次更新到现在CPU时间使用百分比。 |
| N | %MEM | 进程使用内存百分比。 |
| M | TIME+ | 进程使用CPU时间。 |
| X | COMMAND | 命令名/行。 |

表格 13‑1 top进程信息

这是top命令默认显示的列。可以在top输出界面下，按f（或F）键显示所有的列，按相应的序列号可显示或隐藏对应的列，按回车键确认。

**在top的执行过程中，还可以使用以下的按键命令：**

？：显示在top中可以输入的按键命令；

P：按CPU的使用资源排序显示；

M：按内存的使用资源排序显示；

N：按PID来排序；

T：按该进程使用的CPU时间累积排序；

k：给某个PID一个信号（signal），默认值是信号15；

r：重新安排一个进程的优先级别；

i：忽略（显示）闲置和僵死的进程；

S：切换到累计模式；

s：改变两次刷新之间的时间。默认是5秒；

o：按 o 键可以改变列的显示顺序。按小写的 a-z 可以将相应的列向右移动，而大写的 A-Z 可以将相应的列向左移动。最后按回车键确定。

O：然后按 a-z 可以将进程按照相应的列进行排序。而大写的 R 键可以将当前的排序倒转；

l： 切换显示平均负载和启动时间信息；

m：切换显示内存信息；

t：切换显示进程和CPU状态信息；

c：切换显示命令名称和完整命令行；

W：将当前设置写入~/.toprc文件中。这是写top配置文件的推荐方法；

q：退出程序。

**top命令常用的选项：**

-d：后面可以接秒数，指定每两次屏幕信息刷新之间的时间间隔；

-p：指定某个进程来进行监控；

-b -n：以批处理方式执行top命令。通常使用数据流重定向，将处理结果输出为文件；

**进程控制**

**示例：**以nice值为-20运行top命令

1. [root@365linux ~]# nice -n -20 top

Linux系统中每个进程都有一个优先级（PR，priority），PR值越小，表示优先级越高，则越被优先执行。Nice值是我们可以设定的值，取值范围（root -20~19，普通用户 0~19）。NI的正负影响到PR值，所以我们可以设定NI来提升或降低进程的优先级。普通用户仅能降低（增加NI值）属于自己进程的优先级。

**示例：**临时修改进程的优先级

1. [root@365linux ~]# renice 10 3942

将PID为3942的进程的nice值改为10。前文提到，也可以通过top命令中的r按键来修改NI值。

**示例：**终止进程

1. [root@365linux ~]# kill 3942

kill命令是发送一个信号给进程。默认是发送15（TERM）终止。

**示例：**强制中断PID为3942的进程

1. [root@365linux ~]# kill -9 3942

**示例：**平滑重启进程

1. [root@365linux ~]# kill -HUP 3942

**示例：**终止所有httpd启动的进程

1. [root@365linux ~]# killall httpd

**示例：**结束用户user001的所有进程

1. [root@365linux ~]# pkill -9 -U user001

**示例：**查看哪个进程在使用该文件

1. [root@365linux ~]# fuser /var/run/atd.pid

**示例：**查看哪些进程监听了80端口

1. [root@365linux ~]# lsof -i :80

**示例：**查看httpd相关的进程PID号

1. [root@365linux ~]# pidof httpd
2. **收集系统运行状态信息**

**示例：**报告设备或分区的io相关的统计，每三秒钟刷新一次，报告两次。

1. [root@365linux ~]# iostat -d 3 2

**示例：**报告处理器相关的统计

1. [root@365linux ~]# mpstat

**示例：**报告虚拟内存相关的统计

1. [root@365linux ~]# vmstat

**示例：**报告系统运行时间和系统负载

1. [root@365linux ~]# uptime

**示例：**收集、报告、或者保存系统活动信息

1. [root@365linux ~]# sar 2 3

**示例：**以多种格式显示sar收集的信息

1. [root@365linux ~]# sadf
2. **虚拟文件系统**

proc目录是由procfs的文件系统所产生出来的，是kernel加载后，在内存里面建立的一个虚拟目录，包含系统运行时内核信息，进程信息，硬件信息，网络设置，内存使用等，实际上就是系统运行时的内核状态在用户空间的显示。为了保障系统的稳定性，proc目录内的文件不能进行写操作，但/proc/sys目录下的一些文件可以进行重定向开关和更改，以达到在系统运行时，调整和优化内核参数。

**示例：**查看proc文件系统的挂载状态

1. [root@365linux ~]# mount
2. /dev/sda3 on / type ext3 (rw)
3. proc on /proc type proc (rw)
4. ……

**示例：**将proc挂载到另外的目录下

1. [root@365linux vm]# mount -t proc /mnt /mnt

**proc目录内容介绍：**

number：代表目前正在系统中运行的程序。

cpuinfo：CUP硬件信息；

cmdline：加载kernel执行的相关参数；

devices：/dev目录中设备文件分类方式；

filesystems：系统已加载的文件系统；

interrupts：系统上IRQ分配状态；

ioports：系统上设备对应的I/O地址；

kcore：系统上的物理内存大小。不能cat；

meminfo：内存信息；

modules：系统上使用的模块；

partitions：系统分区信息；

net：网络相关的文件；

scsi：scsi设备信息；

sys：核心配置参数。

sys里面文件的设置值是在系统核心里面使用的，部分参数可以实时更改。

**示例：**实时打开内核网络转发的功能

1. [root@365linux ipv4]# echo "1" >/proc/sys/net/ipv4/ip\_forward

通过proc调整内核参数是立即生效，但不会保存（即下次重启失效）。用户可以通过修改/etc/sysctl.conf文件设置内核参数默认值。

我们要学会对系统信息进行收集，分析，了解系统运行状况。

对进程的关注是Linux系统管理最关键的部分，查看每个进程的运行状况，以及对系统资源的占用，对优化系统性能，防范系统潜在的安全威胁都是非常重要的。

要熟悉系统/proc虚拟文件系统的内容，特别对/proc/sys目录有所留意。通过修改/proc/sys内文件的设置值可以直接修改内核运行参数，实时的改变系统运行状态。也可以对/etc/sysctl.conf文件的修改来更改系统内核运行参数的默认值达到按需定制系统，提高系统性能和安全性的要求。

# 日志系统

**【任务描述】**

日志是系统守护进程、内核以及各种工具在运行的过程中产生的信息记录。目的是为了帮助管理员能够通过读取系统日志来获得系统信息，检查错误原因，追踪网络威胁等。

**【学习目标】**

* 1. 查看和分析系统常用的日志文件；
  2. 日志产生的机制；
  3. 日志切割与轮替机制。

## 查看系统日志

在Linux系统中，日志记录文件默认放在系统/var/log目录下，大部分可以通过文本查看命令直接查看。但为安全考虑，与连接登陆有关的日志已二进制文件形式存在，需要用相关命令来查看。

常用的日志文件：

anaconda.\*.log 系统安装日志；

audit 存放系统安全审计的日志，通常与SELINUX有关;

boot.log 系统引导过程中日志;从系统初始化开始的日志，不包含内核引导的日志。且只保留最近一次的系统引导日志记录;

btmp 用户登录系统失败的日志记录，要使用lastb命令进行输出查看;

ConsoleKit/history 图形界面中普通用户管理员授权的历史记录;

cron 周期性计划任务日志;

cups 打印设备使用日志；

dmesg 引导时内核加载的日志;

dracut.log 在安装系统时产生的创建initial ramdisk images为内核提供预加载的模块（块设备模块，访问root文件系统需要）；

gdm 图形界面相关的日志；

lastlog 显示系统中所有用户最后登录的时间，只能使用lastlog命令来查看；

libvirt 虚拟化相关的日志；

miallog 邮件服务器日志;

messages 系统通用日志记录文件（没有特别指定日志文件的日志信息都会记录到此文件）；

ntpstats 时间同步服务相关的日志；

pm-powersave.log pm-suspend.log 电源管理（挂起和休眠）相关的日志；

prelink/prelink.log 记录库的分配的虚拟地址空间位置；

rhsm 红帽授权认证系统管理相关的日志；

sa 收集系统运行的状态信息；

secure 涉及帐号或密码的安全相关的信息；

wtmp 系统帐号登录成功的日志记录，必须要用last命令查看；

Xorg.\*.log xorg图形服务相关的日志；

yum.log yum安装软件的日志。

很多服务日志会记录在/var/log/service\_name 目录下归类管理，日志的存放位置可以自定义。

## 日志产生机制

syslog是一个日志记录系统，在RHEL6中它升级为rsyslog，是syslog的多线程版本。它由一个守护程序组成，随系统启动并持续运行。它能接受系统应用产生的日志信息并且根据/etc/rsyslog.conf的配置信息分类处理，将不同类别，不同程度的消息发送到不同的地方。

/etc/rsyslog.conf的基本格式为：选择域<tab>动作域。

选择域指明发送日志的“设备”和消息的严重性级别。动作域指明如何处理这一消息（记录到文件或发送到设备）。

**示例：**来自邮件系统的info级别以上（包括info）的消息记录到/var/log/maillog文件中。

1. mail.info /var/log/maillog

设备的名称和消息严重性必须从已定义的值中选择。设备分别为内核、常用应用程序组、以及本地编写的程序进行定义。任何其他程序则归为普通设备“用户（user）”一类。

设备名称可以是\*和none，分别表示“所有(除了mark)”和“没有”。多个设备用“，”隔开，多个选择域使用同一动作越可以“；”号连接。

**示例：**所有设备info以上（包括info）消息都记录到massages文件，但除了mail、authpriv、cron。

1. \*.info;mail.none;authpriv.none;cron.none /var/log/messages

**rsyslog识别的发送消息的设备主要有如**表格 0‑1**所示：**

|  |  |
| --- | --- |
| auth | 与安全和授权有关的命令。 |
| authpriv | 敏感/保密的授权消息。 |
| cron | 与计划任务有关。 |
| deamon | 与系统守护进程有关的。 |
| ftp | FTP相关。 |
| kern | 与kernel相关。 |
| local0~7 | 本地消息的8种类型。 |
| lpr | 打印机有关。 |
| mail | 与邮件系统有关。 |
| mark | 由syslogd定期产生的时间戳。 |
| news | Usenet新闻系统。 |
| syslog | syslog本身产生的消息。 |
| user | 用户进程（如果没有指定，这将是默认值）。 |
| uucp | 为UUCP保留，并未使用 |

表格 0‑1 发送消息的设备

**rsyslog的严重性级别定义如**表格 0‑2**所示：**

|  |  |
| --- | --- |
| debug | 仅供调试 |
| info | 提供基本的消息 |
| notice | 需要注意的消息 |
| warning（warn） | 警告消息 |
| error(err) | 错误消息 |
| crit | 临界状态 |
| alert | 严重紧急状态 |
| emerg（pnic） | 恐慌崩溃状态 |

表格 0‑2 日志严重性级别

使用\*表示对某个设备的所有消息进行处理，使用none表示忽略某个设备的消息。

设备和严重性级别之间用“**.**”进行连接，表示严重性要大于等于定义的级别才会被处理。Linux支持“**.=**”和“**.!**”，分别表示只匹配定义级别，除定义的之外的级别。

**rsyslog支持的对消息的处理动作：**

|  |  |
| --- | --- |
| filename | 把消息记录到本机上的一个文件里。 |
| device | 把消息输出到设备，如打印机。 |
| @hostname | 把消息发给远程上的主机，需要远程主机支持。 |
| @ipaddress | 把消息发给远程上的主机，需要远程主机支持。 |
| |fifoname | 把消息通过管道发送。 |
| username | 把消息发送给在线用户。 |
| \* | 把消息发送给在线的所有用户。 |

Linux中许多服务和应用程序都使用syslog。常见的有：

cron、cups、ftpd、inetd、imapd、login、lpd、named、ntpd、passwd、sendmail、ssh、su、sudo、syslog、tcpd、vmlinuz、xinetd 。

## 日志切割与轮替

由于日志消息总是不断的追加，日志文件很变得越来越大，不仅占用了有效的磁盘空间，而且对于庞大的文本文件的写入也会极大的降低系统的性能。如果一次性将旧的日志删除，那会给系统管理员想查看以往的日志造成极大的不便。所以我们需要一个规则，定期的将日志切割轮替，保留一段时间内的日志信息，并将更旧的日志删除。logratate就负责这一工作。

logratate工具配合计划任务进行周期性的日志切割。

**示例：**关于logratate的系统计划任务

1. [root@365linux ~]# cat /etc/cron.daily/logrotate
2. #!/bin/sh
3. /usr/sbin/logrotate /etc/logrotate.conf >/dev/null 2>&1
4. EXITVALUE=$?
5. if [ $EXITVALUE != 0 ]; then
6. /usr/bin/logger -t logrotate "ALERT exited abnormally with [$EXITVALUE]"
7. fi
8. exit 0

**示例：**手动触发日志轮替，并显示轮替的过程信息。

1. [root@365linux log]# logrotate -vf /etc/logrotate.conf

/etc/logratate.conf配置文件定义了轮替的规则。

**示例：**查看logratate.conf配置文件信息

1. [root@365linux ~]# grep -Ev '^#' /etc/logrotate.conf

**主要参数详解：**

**weekly #**默认每周对日志文件进行一次轮替。

**rotate 4 #**保留4个日志文件。

**create #**创建新的日志文件。

**include /etc/logrotate.d #**包括该目录下的文件。

**/var/log/wtmp { #**对wtmp日志文件的单独定义。

**monthly** #每一个月进行一次轮替。

**minsize 1M** #或者是日志大小超过1M时进行轮替。

**create 0664 root utmp** #新建日志文件的权限和用户及所属组。

**rotate 1** #只保留一个日志文件。

**}**

示例：查看子配置文件目录下的syslog的单独定义的轮替规则

1. [root@365linux logrotate.d]# cat /etc/logrotate.d/syslog
2. /var/log/messages /var/log/secure /var/log/maillog /var/log/spooler /var/log/boot.log /var/log/cron { #定义了多个日志文件的轮替。
3. sharedscripts #与endscript搭配，可以定义轮替前或轮替后执行的命令。
4. postrotate #以下是轮替很执行的命令。如要在轮替前执行，则使用prerotate。
5. /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true
6. /bin/kill -HUP `cat /var/run/rsyslogd.pid 2> /dev/null` 2> /dev/null || true
7. endscript
8. }

一般来说，系统进程的运作，系统故障和安全攻击都会留下日志信息，管理员可以通过查看和分析日志，来找出系统异常原因，防范恶意攻击。查看和分析日志时，可以通过时间戳截断需要的时间段的日志进行分析，或者采用关键字过滤提高效率。当需要分析日志量非常大，且需要统计日志数据，如对Web服务器的访问日志进行分析，则需要借助专门的日志分析工具，通过图表直观的展示。

对于管理较多的Linux服务器来说，可以单独提供一台中央日志记录主机，将其他服务器产生的较重要的消息通过远程日志功能统一集中到中央日志记录主机上管理。这样做一方面可以方便系统管理员操作，另一方面可以在一定程度上防止服务器被恶意入侵后，侵入者删除本地日志痕迹，无迹可循。

# SSH服务器

**【任务描述】**

在实际工作中，企业的服务器一般都托管在机房（可能是异地）中，所以在大多数情况下，系统管理员对于服务器的管理和维护的工作，都要通过网络远程操作。

**【学习目标】**

* 1. 通过命令行ssh服务器安全的远程登录到Linux服务器；
  2. 通过VNC服务器远程打开图形界面。

## 使用ssh服务通过命令行远程登录到Linux服务器

在远程连接的过程中，因为要涉及输入服务器的IP地址，登录的账号，密码等安全敏感信息，所以在部署实施远程连接，进行远程登录操作时，要特别注意安全要素，和对服务器进行安全加固，对客户端进行安全审查。

## 通过ssh远程连接

SSH 为 Secure Shell 的缩写，由 IETF 的网络工作小组（Network Working Group）所制定；SSH 为建立在应用层和传输层基础上的安全协议。SSH 是目前较可靠，专为远程登录会话和其他网络服务提供安全性的协议。利用 SSH 协议可以有效防止远程管理过程中的信息泄露问题。SSH最初是UNIX系统上的一个程序，后来又迅速扩展到其他操作平台。SSH在正确使用时可弥补网络中的漏洞。SSH客户端适用于多种平台。几乎所有UNIX平台—包括HP-UX、Linux、AIX、Solaris、Digital UNIX、Irix，以及其他平台，都可运行SSH。SSH在连接和传输数据的过程受用双向非对称加密。

如何查看ssh服务是否在运行

[root@guest01 ~]# systemctl -l status sshd

● sshd.service - OpenSSH server daemon

Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)

Active: active (running) since 一 2016-11-21 09:36:32 CST; 5h 27min ago

Docs: man:sshd(8)

man:sshd\_config(5)

Main PID: 1314 (sshd)

CGroup: /system.slice/sshd.service

└─1314 /usr/sbin/sshd -D

11月 21 09:36:32 guest01 systemd[1]: Started OpenSSH server daemon.

11月 21 09:36:32 guest01 systemd[1]: Starting OpenSSH server daemon...

11月 21 09:36:33 guest01 sshd[1314]: Server listening on 0.0.0.0 port 22.

11月 21 09:36:33 guest01 sshd[1314]: Server listening on :: port 22.

11月 21 09:48:39 guest01 sshd[2042]: Accepted password for root from 192.168.122.1 port 54642 ssh2

11月 21 14:12:01 guest01 sshd[10009]: Accepted password for root from 192.168.122.1 port 54644 ssh2

11月 21 14:13:39 guest01 sshd[10128]: Accepted password for root from 192.168.122.1 port 54646 ssh2

[root@guest01 ~]# ss -ntupl |grep ssh

tcp LISTEN 0 128 \*:22 \*:\* users:(("sshd",pid=1314,fd=3))

tcp LISTEN 0 128 :::22 :::\* users:(("sshd",pid=1314,fd=4))

sshd 已经默认开机启动。 sshd做为一个提供网络连接的服务，它需要监听网络接口和服务端的端口。sshd 默认监听本机上所有的网络接口的ipv4 和 ipv6 的所有的地址， 默认监听22号端口，采用的tcp协议。（PS：每个服务默认规定使用的端口可以通过/etc/services文件查看）

[root@geust02 ~]# ss -ntup |grep sshd

tcp ESTAB 0 0 192.168.122.109:22 192.168.122.1:54646 users:(("sshd",pid=1444,fd=3))

显示了sshd服务当前正在连接的客户端， 客户端的源IP是 192.168.122.1 ， 源端口是54646 ， 服务器端（目标IP）是192.168.122.109 ， 目标端口22。

## 客户端连接

Linux客户端

[root@xueing ~]# ssh test001@192.168.1.101 （或者主机名）

test001@192.168.1.101's password:

Last login: Sat Mar 16 10:57:26 2013

[test001@test001 ~]$ exit

[root@xueing ~]# ssh 192.168.1.101

root@192.168.1.101's password:

默认以客户端当前用户连接登录，即root

Windows 客户端 ， 推荐使用putty , xshell

## 对sshd服务进行配置，主要是安全加固。

sshd的配置文件：

在Linux的服务配置文件中，大多数情况下， 以#号开头的表示注释行，注释掉的选项不起作用（采用应用的默认值。）

sshd配置文件的说明参考 sshd\_config

常用的安全加固的配置选项：

[root@xueing ~]# vim /etc/ssh/sshd\_config

Port 8778 因为默认的22号端口（小于1024的被默认定义为服务端口）是最容易被攻击的对象，所以通常要改为一个大于1024的随机端口。

AddressFamily inet 从实际应用考虑，为安全性，只监听ipv4地址

ListenAddress 192.168.122.200 明确只监听某个IP接口（比如只监听内网的接口）

Protocol 2

LoginGraceTime 1m 减少最大的登录时长，但要合理，不能太小，太小的话正常请求都有可能登陆不上。

PermitRootLogin no 因为root用户名固定容易被攻击，通常不允许root远程登录，而使用普通用户登录，普通用户名可以随机设置为较复杂（包括密码）。

#StrictModes yes

MaxAuthTries 6 重试次数

MaxSessions 10 最大会话数

AllowUsers zhangsan 只允许普通用户zhangsan远程登录。

UseDNS no 不要使用DNS解析主机名

......

其他配置选项，参见， man 5 sshd\_config

对于sshd的安全加固注意事项：

1. 修改了ssh的端口前一定确保防火墙放行了新的端口号，否则应用生效后会导致连接不上。

2.禁止root登录前，一定义确保系统中有可登录的普通用户;

3.ssh的管理通常要配合安全脚本（DenyHost,fail2ban）或防火墙作到防暴力破解；

4.经常检查系统安全日志，关注安全登录情况。

对配置文件修改后要重启服务，使配制生效。

[root@guest01 ~]# systemctl restart sshd （或者reload）

[root@guest01 ~]# ss -ntupl |grep ssh

tcp LISTEN 0 128 \*:2222 \*:\* users:(("sshd",pid=12169,fd=3))

tcp LISTEN 0 128 :::2222 :::\* users:(("sshd",pid=12169,fd=4))

## ssh密钥对登录。

1. 在客户端生成密钥对。（想想，为什么是在客户端生成密钥对呢，而不是在服务端？）

[root@xueing ~]# ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /root/.ssh/id\_rsa.

Your public key has been saved in /root/.ssh/id\_rsa.pub.

The key fingerprint is:

b1:ac:8b:69:ac:f1:71:e1:c7:be:39:76:3b:b8:9a:c1 root@xueing

The key's randomart image is:

+--[ RSA 2048]----+

| |

| |

| . |

| . o |

| . S |

| o + |

| ... E o. |

| oo= \*+.o |

| .o+ +o==.o |

+-----------------+

[root@xueing ~]# ls .ssh/

id\_rsa id\_rsa.pub known\_hosts

2.将密钥对中的公钥id\_rsa.pub上传到指定的服务器上。上传后会自动更名为authorized\_keys。

[root@xueing ~]# ssh-copy-id root@192.168.1.101

root@192.168.1.101's password:

Now try logging into the machine, with "ssh 'root@192.168.1.101'", and check in:

.ssh/authorized\_keys

to make sure we haven't added extra keys that you weren't expecting.

NOTE : ssh-copy-id不支持非22号端口，如果已改变ssh端口，可临时改回默认，上传完公钥后再改成特殊端口，不会产生影响。又或者使用scp或rsync的文件上传功能替换ssh-copy-id。示例如下：

[hdaojin@huangdaojin ~]$ scp -P 8778 .ssh/id\_rsa.pub root@192.168.122.200:/root/.ssh/authorized\_keys

3.上传成功后，再次登录该服务则不需要使用密码。

[root@xueing ~]# ssh root@192.168.1.101

Last login: Tue Mar 19 19:15:36 2013 from 192.168.1.14

[root@test001 ~]# ls .ssh/

authorized\_keys

思考，为什么是在客户端生成密钥对呢，而不是在服务端？

因为私钥不能在网络上传输，存在被人截获的可能。

## ssh远程复制文件的功能

在图形界面可以使用“连接到服务器”的功能，然后使用sftp的协议登录，上传下载。

命令行里，可使用scp或rsync命令。

1.上传文件到远程服务器

[chuyue@xueing 桌面]$ scp ssh.txt test001@192.168.1.101:/home/test001/Desktop

The authenticity of host '192.168.1.101 (192.168.1.101)' can't be established.

RSA key fingerprint is 12:4c:00:43:ec:28:71:2d:0b:c6:1b:18:0c:cf:8c:ee.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '192.168.1.101' (RSA) to the list of known hosts.

test001@192.168.1.101's password:

ssh.txt 100% 250 0.2KB/s 00:00

2.从远程服务器上下载指定的文件

[chuyue@xueing 桌面]$ scp test001@192.168.1.101:/home/test001/test.txt /home/chuyue/

scp命令可以使用-r的参数上传和下载目录。

3. rsync命令同样可以实现类似scp的功能，不过rsync的远程同步的功能更强大，支持压缩传输、断点续传、差量复制、守护进程等功能。rsync要另外安装。

[chuyue@xueing 桌面]$ rsync -avP /boot/initramfs-2.6.32-358.el6.x86\_64.img test001@192.168.1.101:/home/test001/

test001@192.168.1.101's password:

sending incremental file list

initramfs-2.6.32-358.el6.x86\_64.img

16541420 100% 12.69MB/s 0:00:01 (xfer#1, to-check=0/1)

sent 16543540 bytes received 31 bytes 2545164.77 bytes/sec

total size is 16541420 speedup is 1.00

**ssh的转发图形界面的功能**

[demo@365linux ~]$ ssh -X ccj@192.168.20.105

然后在执行打开图形工具的命令，比如gedit

SSH是个安全的命令行远程连接的工具，它的优点是安全（加密），速度快（传输的是基于文本的命令行信息）

# RSYNC服务器

**【任务描述】**

rsync是Unix下的一款应用软件，它能同步更新两处计算机的文件与目录，并适当利用差分编码以减少数据传输量。rsync中的一项同类软件不常见的重要特性是每个目标的镜像只需发送一次。rsync可以拷贝／显示目录内容，以及拷贝文件，并可选压缩以及递归拷贝。

在常驻模式（daemon mode）下，rsync默认监听TCP端口873，以原生rsync传输协议或者透过远程shell如RSH或者SSH提供文件。SSH模式下，rsync客户端运行程序必须同时在本地和远程机器上安装。

rsync是以GNU通用公共许可证发行的自由软件。

**【学习目标】**

1. rsync同步命令的使用能力；
2. rsync同步服务的配置能力。

## 了解rsync的语法

rsync命令是一个远程数据同步工具，可通过LAN/WAN快速同步多台主机间的文件。rsync使用所谓的“rsync算法”来使本地和远程两个主机之间的文件达到同步，这个算法只传送两个文件的不同部分，而不是每次都整份传送，因此速度相当快。 rsync是一个功能非常强大的工具，其命令也有很多功能特色选项，我们下面就对它的选项一一进行分析说明。

**语法**

Usage: rsync [OPTION]... SRC [SRC]... DEST

or rsync [OPTION]... SRC [SRC]... [USER@]HOST:DEST

or rsync [OPTION]... SRC [SRC]... [USER@]HOST::DEST

or rsync [OPTION]... SRC [SRC]... rsync://[USER@]HOST[:PORT]/DEST

or rsync [OPTION]... [USER@]HOST:SRC [DEST]

or rsync [OPTION]... [USER@]HOST::SRC [DEST]

or rsync [OPTION]... rsync://[USER@]HOST[:PORT]/SRC [DEST]

对应于以上七种命令格式，rsync有7不同的工作模式：

1. 拷贝本地文件。当SRC和DES路径信息都不包含有单个冒号":"分隔符时就启动这种工作模式。如：rsync -a /data /backup
2. 使用一个远程shell程序(如[rsh](http://man.linuxde.net/rsh)、[ssh](http://man.linuxde.net/ssh))来实现将本地机器的内容拷贝到远程机器。当DST路径地址包含单个冒号":"分隔符时启动该模式。如：rsync -avz \*.c foo:dst
3. 使用一个远程shell程序(如rsh、ssh)来实现将远程机器的内容拷贝到本地机器。当SRC地址路径包含单个冒号":"分隔符时启动该模式。如：rsync -avz foo:src/bar /data
4. 从远程rsync服务器中拷贝文件到本地机。当SRC路径信息包含"::"分隔符时启动该模式。如：rsync -av root@192.168.78.192::www /databack
5. 从本地机器拷贝文件到远程rsync服务器中。当DST路径信息包含"::"分隔符时启动该模式。如：rsync -av /databack root@192.168.78.192::www
6. 列远程机的文件列表。这类似于rsync传输，不过只要在命令中省略掉本地机信息即可。如：rsync -v rsync://192.168.78.192/www
7. 如:rsync -v src rsync://192.168.78.192/www

**选项**

-v, --verbose 详细模式输出。

-q, --quiet 精简输出模式。

-c, --checksum 打开校验开关，强制对文件传输进行校验。

-a, --archive 归档模式，表示以递归方式传输文件，并保持所有文件属性，等于-rlptgoD。

-r, --recursive 对子目录以递归模式处理。

-R, --relative 使用相对路径信息。

-b, --backup 创建备份，也就是对于目的已经存在有同样的文件名时，将老的文件重新命名为~filename。可以使用--suffix选项来指定不同的备份文件前缀。

--backup-dir 将备份文件(如~filename)存放在在目录下。

-suffix=SUFFIX 定义备份文件前缀。

-u, --update 仅仅进行更新，也就是跳过所有已经存在于DST，并且文件时间晚于要备份的文件，不覆盖更新的文件。

-l, --links 保留软链结。

-L, --copy-links 想对待常规文件一样处理软链结。

--copy-unsafe-links 仅仅拷贝指向SRC路径目录树以外的链结。

--safe-links 忽略指向SRC路径目录树以外的链结。

-H, --hard-links 保留硬链结。

-p, --perms 保持文件权限。

-o, --owner 保持文件属主信息。

-g, --group 保持文件属组信息。

-D, --devices 保持设备文件信息。

-t, --times 保持文件时间信息。

-S, --sparse 对稀疏文件进行特殊处理以节省DST的空间。

-n, --dry-run现实哪些文件将被传输。

-w, --whole-file 拷贝文件，不进行增量检测。

-x, --one-file-system 不要跨越文件系统边界。

-B, --block-size=SIZE 检验算法使用的块尺寸，默认是700字节。

-e, --rsh=command 指定使用rsh、ssh方式进行数据同步。

--rsync-path=PATH 指定远程服务器上的rsync命令所在路径信息。

-C, --cvs-exclude 使用和CVS一样的方法自动忽略文件，用来排除那些不希望传输的文件。

--existing 仅仅更新那些已经存在于DST的文件，而不备份那些新创建的文件。

--delete 删除那些DST中SRC没有的文件。

--delete-excluded 同样删除接收端那些被该选项指定排除的文件。

--delete-after 传输结束以后再删除。

--ignore-errors 及时出现IO错误也进行删除。

--max-delete=NUM 最多删除NUM个文件。

--partial 保留那些因故没有完全传输的文件，以是加快随后的再次传输。

--force 强制删除目录，即使不为空。

--numeric-ids 不将数字的用户和组id匹配为用户名和组名。

--timeout=time ip超时时间，单位为秒。

-I, --ignore-times 不跳过那些有同样的时间和长度的文件。

--size-only 当决定是否要备份文件时，仅仅察看文件大小而不考虑文件时间。

--modify-window=NUM 决定文件是否时间相同时使用的时间戳窗口，默认为0。

-T --temp-dir=DIR 在DIR中创建临时文件。

--compare-dest=DIR 同样比较DIR中的文件来决定是否需要备份。

-P 等同于 --partial --progress。

--progress 显示备份过程。

-z, --compress 对备份的文件在传输时进行压缩处理。

--exclude=PATTERN 指定排除不需要传输的文件模式。

--include=PATTERN 指定不排除而需要传输的文件模式。

--exclude-from=FILE 排除FILE中指定模式的文件。

--include-from=FILE 不排除FILE指定模式匹配的文件。

--version 打印版本信息。

--address 绑定到特定的地址。

--config=FILE 指定其他的配置文件，不使用默认的rsyncd.conf文件。

--port=PORT 指定其他的rsync服务端口。

--blocking-io 对远程shell使用阻塞IO。

-stats 给出某些文件的传输状态。

--progress 在传输时现实传输过程。

--log-format=formAT 指定日志文件格式。

--password-file=FILE 从FILE中得到密

-h, --help 显示帮助信息。

## rsync ssh模式

rsync -avzhP --delete -e 'ssh -p 5000' test@linux.com:/home/hdaojin/ ./

-a, ––archive 归档模式，表示以递归方式传输文件，并保持所有文件属性，等价于 -rlptgoD (注意不包括 -H,-A,-X)

-v, ––verbose 详细输出模式

-z, ––compress 在传输文件时进行压缩处理

-h, ––human-readable 输出文件大小使用易读的单位（如，K，M等）

-P 等价于 ––partial ––progress

––partial 保留那些因故没有完全传输的文件，以是加快随后的再次传输

––progress 在传输时显示传输过程

-e, ––rsh=COMMAND 指定替代 rsh 的 shell 程序

在备份脚本中使用，只需要采用：

rsync -azvhq --delete --partial --log-file=/var/log/rsync.log

-q 安静模式

## rsyncd模式

xinetd 和rsyncd模式

# vim /etc/rsyncd.conf

[test]

comment = Just test

path = /test

use chroot = yes

read only = yes

list = yes

uid = nobody

gid = nobody

max connections = 30

hosts allow = 192.168.122.0/24

strict modes = yes

ignore errors = no

ignore nonreadable = yes

transfer logging = no

timeout = 600

refuse options = checksum dry-run remove-source-files

dont compress = \*.gz \*.tgz \*.zip \*.z \*.rpm \*.deb \*.iso \*.bz2 \*.tbz \*.rar \*.img

可选：

# vim /etc/xinetd.d/rsync

service rsync

{

disable = no

flags = ipv4

socket\_type = stream

wait = no

user = root

server = /usr/bin/rsync

server\_args = --daemon

log\_on\_failure += USERID

}

客户端：

# rsync -aqzH --delete 192.168.122.201::test ~/notes

# NTP服务器

**【任务描述】**

网络时间协议（英语：Network Time Protocol，缩写：NTP）是在数据网络潜伏时间可变的计算机系统之间通过分组交换进行时钟同步的一个网络协议，位于OSI模型的应用层。自1985年以来，NTP是目前仍在使用的最古老的互联网协议之一。NTP由特拉华大学的David L. Mills设计。

NTP意图将所有参与计算机的协调世界时（UTC）时间同步到几毫秒的误差内。它使用Marzullo算法的修改版来选择准确的时间服务器，其设计旨在减轻可变网络延迟造成的影响。NTP通常可以在公共互联网保持几十毫秒的误差，并且在理想的局域网环境中可以实现超过1毫秒的精度。不对称路由和拥塞控制可能导致100毫秒（或更高）的错误。

**【学习目标】**

1. 了解NTP协议的基本概念
2. 创建内部网络的NTP时间服务器
3. 使用NTP客户端同步服务器时间

## NTP协议的基本概念

NTP是网络时间协议(Network Time Protocol)，它是用来同步网络中各个计算机的时间的协议。该协议通常描述为一种主从式架构，但它也可以用在点对点网络中，对等体双方可将另一端认定为潜在的时间源。发送和接收时间戳采用用户数据报协议（UDP）的端口123实现。这也可以使用广播或多播，其中的客户端在最初的往返校准交换后被动地监听时间更新。

## 使用ntpdate命令同步时间

root@debian:~# apt install ntpdate

root@debian:~# date

2019年 04月 10日 星期三 10:40:03 CST

root@debian:~# ntpdate 182.92.12.11

10 Apr 10:14:56 ntpdate[2126]: step time server 182.92.12.11 offset -1536.577345 sec

root@debian:~# date

2019年 04月 10日 星期三 10:15:25 CST

在中国境内，通过<http://www.ntp.org.cn/pool.php>获取ntp服务器的地址。

## 搭建ntp服务器

root@102:~# apt install ntp

root@102:~# vim /etc/ntp.conf

root@102:~# systemctl status ntp

root@debian:~# date -s "10:40"

2019年 04月 10日 星期三 10:40:00 CST

root@debian:~# ntpdate 192.168.38.102

10 Apr 10:23:02 ntpdate[2137]: step time server 192.168.38.102 offset -1040.282222 sec

root@debian:~# date

2019年 04月 10日 星期三 10:23:06 CST

## 无外网连接的情况下使用ntp服务

root@102:~# ping 8.8.8.8

connect: Network is unreachable

root@102:~# vim /etc/ntp.conf

# pool 0.debian.pool.ntp.org iburst

# pool 1.debian.pool.ntp.org iburst

# pool 2.debian.pool.ntp.org iburst

# pool 3.debian.pool.ntp.org iburst

server 127.127.1.0

fudge 127.127.1.0 stratum 10

root@102:~# systemctl restart ntp

# DNS服务器

**【任务描述】**

DNS 是域名系统 (Domain Name System) 的缩写，是因特网的一项核心服务，它作为可以将域名和 IP 地址相互映射的一个分布式数据库，能够使人更方便的访问互联网，而不用去记住能够被机器直接读取的 IP 数串。

**【学习目标】**

* 1. 充分理解DNS服务分层设计的体系机构和工作原理
  2. 能够使用Bind软件搭建本地DNS服务器
  3. 理解Bind的chroot运行模式
  4. 实现正向解析和反向解析
  5. 理解DNS轮循，别名，邮件标记，泛解析，连续解析
  6. 使用DNS服务器端管理工具检验和管理DNS服务器
  7. 使用DNS客户端测试工具测试DNS服务器的工作状态

## 配置DNS域名解析服务器

域名到 IP 的解析过程：客户端本地缓存（浏览器和系统）🡪 hosts 🡪 路由器缓存 🡪 本地 DNS 服务器（由 ISP（互联网服务提供商）提供(本地权威或缓存非权威）) 🡪 被转发的 DNS 服务器 🡪 根域名服务器 🡪 顶级域名服务器 🡪 主域名服务器 -🡪 保存结果至缓存 🡪 返回结果给客户机，任何一级得到结果都会返回给客户机。

对于用户，只需要记住要访问的服务器的名称而不用记住一串IP数字；对于管理员，改变服务器的IP不会影响基于名称的访问，以及可以明确哪些IP处理这些请求。

Bind是一款开放源码的DNS服务器软件，Bind由美国加州大学Berkeley分校开发和维护的，全名为 Berkeley Internet Name Domain 它是目前世界上使用最为广泛的 DNS 服务器软件，支持各种 unix 平台和 windows 平台。

1. 安装

root@dns:~# apt search bind

root@dns:~# apt install bind9 bind9-doc bind9utils

2. 配置（正向解析）

root@dns:/etc/bind# vim named.conf.local

zone "app4you.com" IN {

type master;

file "/etc/bind/db.app4you.com";

};

root@dns:/etc/bind# cp -a db.local db.app4you.com

root@dns:/etc/bind# vim db.app4you.com

$TTL 604800

@ IN SOA localhost. root.localhost. (

2 ; Serial

604800 ; Refresh

86400 ; Retry

2419200 ; Expire

604800 ) ; Negative Cache TTL

;

@ IN NS localhost.

@ IN A 127.0.0.1

www A 192.168.88.130

3.启动

root@dns:/etc/bind# systemctl restart bind9

root@dns:/etc/bind# systemctl status bind9

root@dns:/etc/bind# ss -ntupl |grep named|xargs -n 7

udp UNCONN 0 0 192.168.88.129:53 \*:\* users:((named,pid=1652,fd=517),(named,pid=1652,fd=516))

4.测试

root@client1:~# vim /etc/resolv.conf

nameserver 192.168.88.129

root@client1:~# nslookup www.app4you.com

Server: 192.168.88.129

Address: 192.168.88.129#53

Name: www.app4you.com

Address: 192.168.88.130

## 内外网智能解析

root@dns:/etc/bind# vim named.conf

include "/etc/bind/named.conf.options";

// include "/etc/bind/named.conf.local";

// include "/etc/bind/named.conf.default-zones";

view "internal" {

match-clients { 192.168.88.0/24; };

recursion yes;

zone "app4you.com" IN {

type master;

file "/etc/bind/db.app4you.com";

};

include "/etc/bind/named.conf.default-zones";

};

view "external" {

match-clients { any; };

recursion no;

zone "app4you.com" IN {

type master;

file "/etc/bind/db.app4you.com.ex";

};

};

root@dns:/etc/bind# vim db.app4you.com

$TTL 604800

@ IN SOA localhost. root.localhost. (

2 ; Serial

604800 ; Refresh

86400 ; Retry

2419200 ; Expire

604800 ) ; Negative Cache TTL

;

@ IN NS localhost.

@ IN A 127.0.0.1

www A 192.168.88.130

root@dns:/etc/bind# vim db.app4you.com.ex

$TTL 604800

@ IN SOA localhost. root.localhost. (

2 ; Serial

604800 ; Refresh

86400 ; Retry

2419200 ; Expire

604800 ) ; Negative Cache TTL

;

@ IN NS localhost.

@ IN A 127.0.0.1

www A 114.5.6.7

root@dns:/etc/bind# systemctl restart bind9

root@dns:/etc/bind# systemctl status bind9

配置完后可以使用named-checkconf检查配置文件的正确性；

配置完后可以使用named-checkzone命令检查zone文件的正确性；

rndc是一个命令行的nameserver管理工具，可以从本地或远程管理。

相关文件：

/etc/named.conf 可在named.conf中使用controls语句进行安全设定

/etc/rndc.conf rndc工具的默认配置文件

/etc/rndc.key rndc默认的key文件

rndc命令示例：

# rndc status 检查服务器的状态

# rndc reload 重新载入配置和zone

# rndc reload localhost 重新载入某个zone

# rndc reconfig 仅重新载入配置

# rndc querylog 开启（或关闭）查询日志

什么时候需要建立nameserver?

在购买域名时，域名注册商通常已经提供域名到ip的解析功能，一般用户只需要在基于web页面的域名管理后台添加资源记录即可。或者不使用域名注册商提供的dns解析服务，而使用第三方功能和能力更强大的dns服务（如dnspod，通常需要购买相应的服务）。所以域名注册商或专业的dns服务商需要建立nameserver，而一般的个人域名用户或规模较小的公司直接使用dns服务商提供的即可。

在互联网上也有较多由ISP或企业提供的公共的域名缓存服务器，如223.5.5.5 ; 8.8.8.8;等。

对于dns域名解析性能和安全性要求较高的企业通常会利用较好的服务器和带宽自己nameserver。

对于组织架构较复杂的公司，通常需要自建nameserver进行内部服务器进行名称解析。

authoritative and recursive

理论上可以同时使用authoritative 和 recursive nameserver ,但是不建议这样。

官方说明如下：

Although a nameserver can be both authoritative and recursive at the same time, it is recommended not to combine the configuration types. To be able to perform their work, authoritative servers should be available to all clients all the time. On the other hand, since the recursive lookup takes far more time than authoritative responses, recursive servers should be available to a restricted number of clients only, otherwise they are prone to distributed denial of service (DDoS) attacks.

To prevent distributed denial of service (DDoS) attacks, it is recommended that you use the allow-query-cache option to restrict recursive DNS services for a particular subset of clients only.

# APACHE服务器

**【任务描述】**

公司有一个新的项目，要搭建一个门户网站，该网站要求使用加密访问，支持虚拟主机等功能。做为服务器系统管理员，你觉得应该使用apache服务器来满足上述要求。

**【学习目标】**

* 1. 学会创建web虚拟主机；
  2. 实现web服务器目录索引的功能；
  3. 支持ssl加密https。

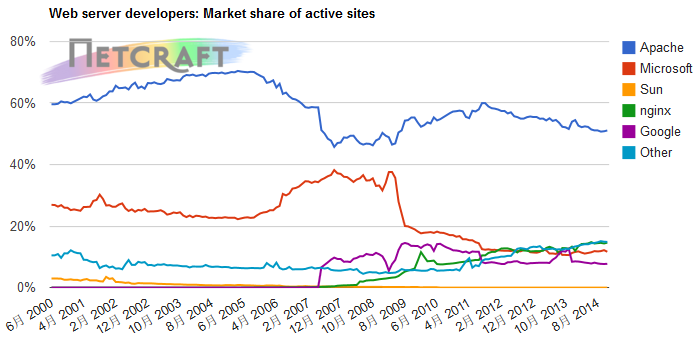
## 了解web服务器的基本概念

万维网（亦作“Web”、“WWW”、“'W3'”，英文全称为“World Wide Web”），是一个由许多互相链接的超文本组成的系统，通过互联网访问。在这个系统中，每个有用的事物，称为一样“资源”；并且由一个全域“统一资源标识符”（URI）标识；这些资源通过超文本传输协议（Hypertext Transfer Protocol）传送给用户，而后者通过点击链接来获得资源。

HTTP 超文本传输协议（英文：HyperText Transfer Protocol，缩写：HTTP）是互联网上应用最为广泛的一种网络协议。设计HTTP最初的目的是为了提供一种发布和接收HTML页面的方法。通过HTTP或者HTTPS协议请求的资源由统一资源标识符（Uniform Resource Identifiers，URI）来标识。

Apache HTTP Server（简称Apache）是Apache软件基金会的一个开放源码的网页服务器，可以在大多数计算机操作系统中运行，由于其多平台和安全性被广泛使用，是最流行的Web服务器端软件之一。

实现该服务器端的软件还有比如nginx , IIS(WINDOWS平台)，lighttpd等等。



## 安装apache服务器软件

root@web:~# aptitude search apache

root@web:~# aptitude install apache2

root@web:/etc/apache2# service apache2 restart

root@web:/etc/apache2# ss -ntupl |grep apache2

tcp LISTEN 0 128 :::80 :::\* users:(("apache2",pid=3770,fd=4),("apache2",pid=3769,fd=4),("apache2",pid=3727,fd=4))

修改运行用户：

root@web:/etc/apache2# vim envvars

export APACHE\_RUN\_USER=www

export APACHE\_RUN\_GROUP=www

root@web:/etc/apache2# useradd www

错误页面最小信息

ServerTokens Prod

ServerSignature Off

## 创建新的虚拟主机

root@web:/etc/apache2/sites-available# cp -a 000-default.conf 001-www.apps4you.com.conf

<VirtualHost \*:80>

ServerName www.apps4you.com

DocumentRoot /htdocs/www

ErrorLog ${APACHE\_LOG\_DIR}/www.apps4you.com.error.log

CustomLog ${APACHE\_LOG\_DIR}/www.apps4you.com.access.log combined

</VirtualHost>

# mkdir -p /htdocs/www

# echo "the frist web page." >> /htdocs/www/index.html

# a2ensite 001-www.apps4you.com.conf

root@web:/etc/apache2# ll sites-enabled/

total 8.0K

drwxr-xr-x 2 root root 4.0K Dec 23 21:36 .

drwxr-xr-x 8 root root 4.0K Dec 23 21:26 ..

lrwxrwxrwx 1 root root 35 Dec 23 20:45 000-default.conf -> ../sites-available/000-default.conf

lrwxrwxrwx 1 root root 44 Dec 23 21:36 001-www.apps4you.com.conf -> ../sites-available/001-www.apps4you.com.conf

root@web:/etc/apache2# service apache2 reload

## 目录索引

<Directory /htdocs/www/>

Options Indexes

</Directory>

root@web:/htdocs/www# rm -rf index.html

root@web:/htdocs/www# touch a.mp3 b.mp4 c.txt d.db

## ssl加密https

root@web:/etc/apache2# a2enmod ssl

root@web:/etc/apache2/sites-available# sed -e '/#.\*$/d' -e '/^$/d' default-ssl.conf > 001-www.apps4you.com.ssl.conf

<IfModule mod\_ssl.c>

<VirtualHost \*:443>

ServerAdmin webmaster@localhost

DocumentRoot /htdocs/www

ServerName www.apps4you.com

ErrorLog ${APACHE\_LOG\_DIR}/www.apps4you.com.error.log

CustomLog ${APACHE\_LOG\_DIR}/www.apps4you.com.access.log combined

SSLEngine on

SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem

SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

<FilesMatch "\.(cgi|shtml|phtml|php)$">

SSLOptions +StdEnvVars

</FilesMatch>

<Directory /usr/lib/cgi-bin>

SSLOptions +StdEnvVars

</Directory>

BrowserMatch "MSIE [2-6]" \

nokeepalive ssl-unclean-shutdown \

downgrade-1.0 force-response-1.0

BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

</VirtualHost>

</IfModule>

root@web:/etc/apache2/sites-available# a2ensite 001-www.apps4you.com.ssl.conf

root@web:/etc/apache2/sites-available# systemctl restart apache2

**使用自签名证书**

root@web:/etc/apache2/ssl# openssl req -new -x509 -nodes -out web.crt -keyout web.key

**客户端在命令行信任证书**

root@client1:~# cp web.crt /usr/local/share/ca-certificates/

root@client1:~# update-ca-certificates

**模拟CA签名**

root@ca:~# mkdir /ca

root@ca:~# vim /etc/ssl/openssl.cnf

dir = /ca

root@ca:~# vim /usr/lib/ssl/misc/CA.pl

$CATOP="/ca";

root@ca:~# /usr/lib/ssl/misc/CA.pl -newca

root@ca:~# ls /ca/

cacert.pem certs crlnumber index.txt.attr newcerts serial

careq.pem crl index.txt index.txt.old private

root@web:~# /usr/lib/ssl/misc/CA.pl -newreq-nodes

root@web:~# ls

newkey.pem newreq.pem

root@web:~# scp newreq.pem root@192.168.213.136:/ca/

root@ca:/ca# /usr/lib/ssl/misc/CA.pl -sign

root@ca:/ca# ls newcert.pem

newcert.pem

root@ca:/ca# scp newcert.pem 192.168.213.130:~/

root@web:~# cp newcert.pem newkey.pem /etc/apache2/ssl/

**http永久跳转到https**

root@web:/etc/apache2/sites-enabled# vim 002-intranet.apps4you.com.conf

<VirtualHost \*:80>

ServerName intranet.apps4you.com

Redirect 301 / https://intranet.apps4you.com/

</VirtualHost \*:80>

apache具有非常强大的功能和可配置性，在作为对外（大多数情况下是对所有网络）提供访问服务的应用，所以在功能配置，安全性，稳定性，性能调整上都有极大的可配置性。比如工作模式，进程（线程）数，与动态应用的交互方式，安全加固，运行可执行的CGI，静态缓存，代理，负载均衡等。

apache作为当前最流行的web服务器，重要性不言而喻；而当前在国内大量的web应用（比如淘宝）开始使用web服务器后起之秀Nginx。

# NGINX服务器

**【任务描述】**

Nginx（发音同 engine x）是一款轻量级的Web 服务器／反向代理服务器及电子邮件（IMAP/POP3）代理服务器，并在一个BSD-like 协议下发行。由俄罗斯的程序设计师Igor Sysoev所开发，最初供俄国大型的入口网站及搜寻引擎Rambler（俄文：Рамблер）使用。 其特点是占有内存少，并发能力强，事实上nginx的并发能力确实在同类型的网页伺服器中表现较好.目前中国大陆使用nginx网站用户有：新浪、网易、 腾讯,另外知名的微网志Plurk也使用nginx。

**【学习目标】**

* 了解nginx服务器的应用场景、与Apache的对比
* 掌握nginx作为Web服务器的配置
* 熟悉nginx的运维管理、故障排察、性能调优
* 能够正确安装、配置、运行nginx虚拟主机；
* 能够自定义nginx的功能配置、日志等。

## 安装配置lnmp架构

安装配置lnmp（linux+nginx+MySQL+PHP）, 安装wordpress应用，[网址为www.ccbili.com](http://xn--www-c88dn62bls2d.ccbili.com)；

root@gz-server01:/etc/nginx/sites-available# apt install php-fpm

root@gz-server01:/etc/nginx/sites-available# vim /etc/php/7.3/fpm/pool.d/www.conf

listen = 127.0.0.1:9000

server{

listen 80;

server\_name www.ccbili.com;

root /www/ccbili;

index index.php index.html;

access\_log /var/log/nginx/ccbili.com.log;

error\_log /var/log/nginx/error\_ccbili.com.log;

location ~ \.php$ {

include snippets/fastcgi-php.conf;

fastcgi\_pass 127.0.0.1:9000;

}

}

## nginx支持https

root@gz-server01:/www/ccbili# cat /etc/nginx/sites-enabled/ccbili.conf

server{

listen 80;

listen 443 ssl;

ssl\_certificate /etc/ssl/certs/ssl-cert-snakeoil.pem;

ssl\_certificate\_key /etc/ssl/private/ssl-cert-snakeoil.key;

server\_name www.ccbili.com;

root /www/ccbili;

index index.php index.html;

access\_log /var/log/nginx/ccbili.com.log;

error\_log /var/log/nginx/error\_ccbili.com.log;

location ~ \.php$ {

include snippets/fastcgi-php.conf;

fastcgi\_pass 127.0.0.1:9000;

}

}

## nginx实现url跳转

访问http(s)://ccbili.com自动跳转到www.ccbili.com, 访问http(s)://any.ccbili.com或http(s)://IP则拒绝访问。

1. server{
2. listen 80;
3. server\_name ccbili.com;
4. return 301 http://www.ccbili.com;
5. }
6. root@gz-server01:/etc/nginx/sites-enabled# vim default
7. server {
8. listen 80 default\_server;
9. listen [::]:80 default\_server;
10. server\_name \_;
11. return 403;
12. }

# SQUID服务器

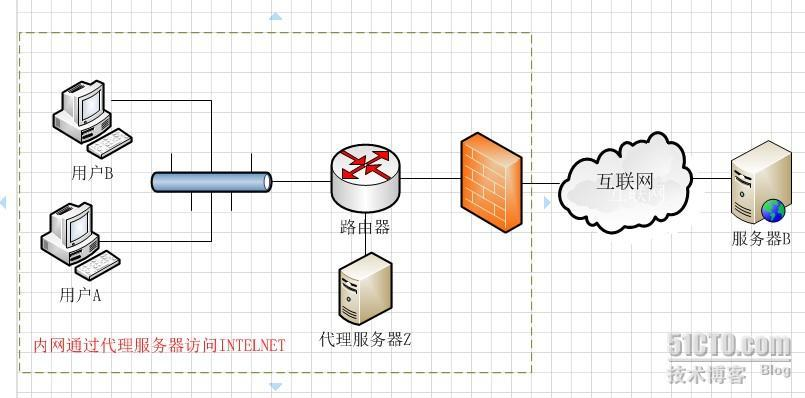
**【任务描述】**

Squid Cache（简称为Squid）是HTTP代理服务器软件。Squid用途广泛，可以作为缓存服务器，可以过滤流量帮助网络安全，也可以作为代理服务器链中的一环，向上级代理转发数据或直接连接互联网。Squid程序在Unix一类系统运行。

Squid历史悠久，功能完善。除了HTTP外，对FTP与HTTPS的支持也相当好，在3.0测试版中也支持了IPv6。

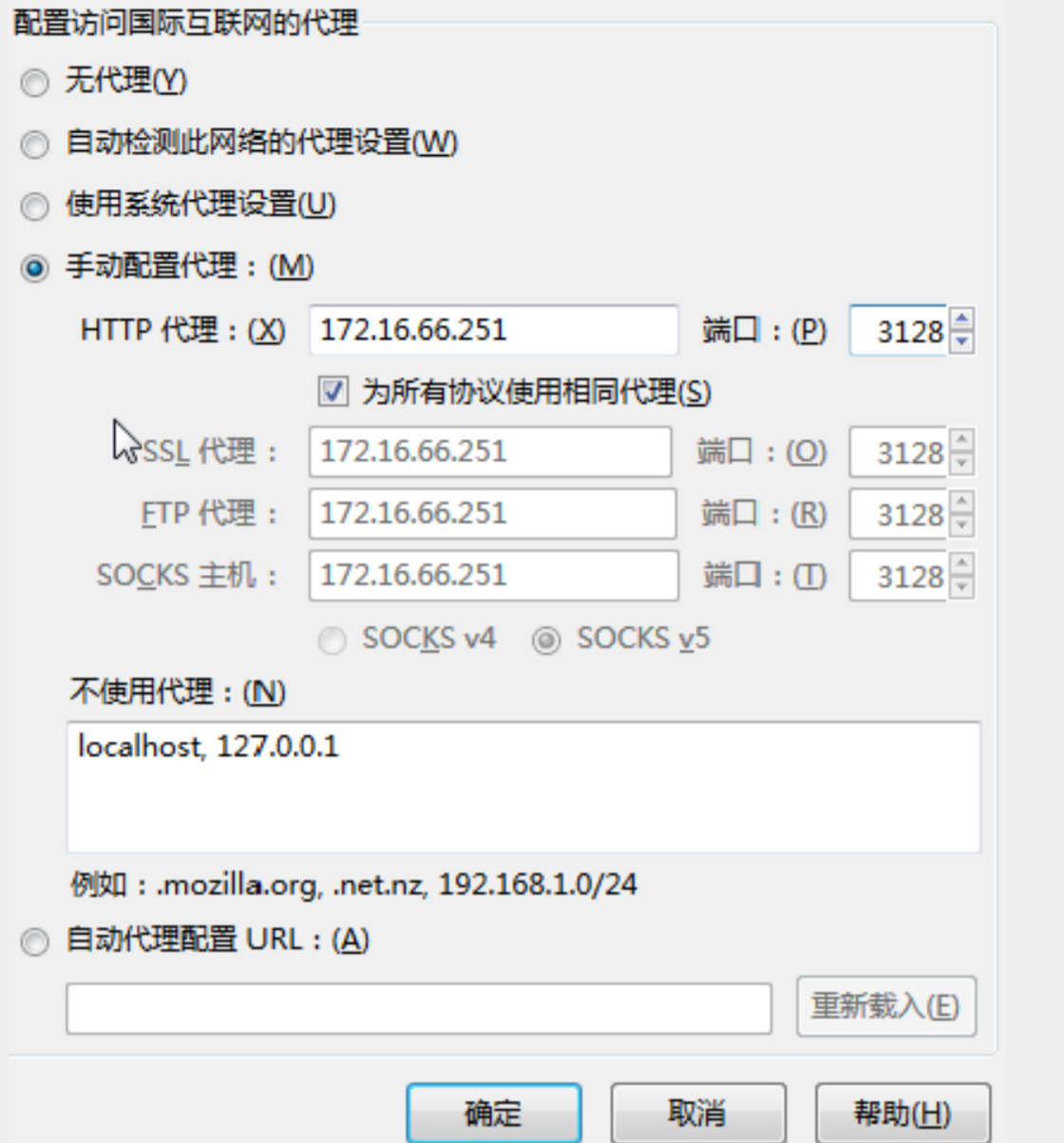
**【学习目标】**

## 配置Squid正向代理



Debian 10.2中squid默认是4.6，镜像中仍然有squid3的版本提供。

1. root@squid:~# apt install squid
2. root@squid:/etc/squid# grep -v "^#" squid.conf.backup |grep -v "^$" > squid.conf
3. acl SSL\_ports port 443
4. acl Safe\_ports port 80 # http
5. acl Safe\_ports port 21 # ftp
6. acl Safe\_ports port 443 # https
7. acl Safe\_ports port 70 # gopher
8. acl Safe\_ports port 210 # wais
9. acl Safe\_ports port 1025-65535 # unregistered ports
10. acl Safe\_ports port 280 # http-mgmt
11. acl Safe\_ports port 488 # gss-http
12. acl Safe\_ports port 591 # filemaker
13. acl Safe\_ports port 777 # multiling http
14. acl CONNECT method CONNECT
15. acl internal src 192.168.10.0/24
16. http\_access deny !Safe\_ports
17. http\_access deny CONNECT !SSL\_ports
18. http\_access allow localhost manager
19. http\_access deny manager
20. http\_access allow localhost
21. http\_access allow internal
22. http\_access deny all
23. visible\_hostname squid\_server
24. cache\_mem 256 MB
25. cache\_dir ufs /var/spool/squid 1000 16 256
26. http\_port 3128
27. coredump\_dir /var/spool/squid
28. refresh\_pattern ^ftp: 1440 20% 10080
29. refresh\_pattern ^gopher: 1440 0% 1440
30. refresh\_pattern -i (/cgi-bin/|\?) 0 0% 0
31. refresh\_pattern . 0 20% 4320

以下浏览器的设置图片仅供参考。也可以在图形网络连接设置里进行代理设置。  
 

## 配置Squid的ACL

acl nobaidu dstdomain .baidu.com

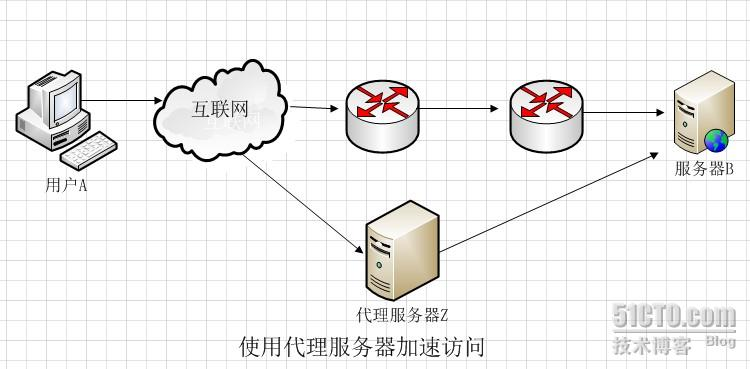
http\_access deny nobaidu



## 配置Squid透明代理

1. http\_port 3128 intercept
2. root@router:/etc/squid# iptables -t nat -I PREROUTING -s 192.168.10.0/24 -p tcp --dport 80 -j REDIRECT --to-port 3128
3. root@router:/etc/squid# iptables -t nat -nL
4. Chain PREROUTING (policy ACCEPT)
5. target prot opt source destination
6. REDIRECT tcp -- 192.168.10.0/24 0.0.0.0/0 tcp dpt:80 redir ports 3128
7. Chain INPUT (policy ACCEPT)
8. target prot opt source destination
9. Chain OUTPUT (policy ACCEPT)
10. target prot opt source destination
11. Chain POSTROUTING (policy ACCEPT)
12. target prot opt source destination
13. MASQUERADE udp -- 192.168.10.0/24 !192.168.10.0/24 udp dpt:53
14. MASQUERADE tcp -- 192.168.10.0/24 !192.168.10.0/24 tcp dpt:53
15. MASQUERADE tcp -- 192.168.10.0/24 0.0.0.0/0 tcp dpt:443

## 配置Squid反向代理



1. http\_port 80 accel defaultsite=192.168.10.102
2. cache\_peer 192.168.10.102 parent 80 0 no-query originserver

# NFS服务器

**【任务描述】**

是网络附加储存（NAS）的一种，其功能旨在允许客户端主机可以像访问本地存储一样通过网络访问服务器端文件。主要用在LINUX和LINUX系统之间提供共享存储。

NFS和其他许多协议一样，是基于开放网络运算远程过程调用（ONC RPC）协议之上的。它是一个开放、标准的RFC协议，任何人或组织都可以依据标准实现它。

服务的配置和应用过程： 装 配 启 测

1. 装： 安装  
   配： 配置  
   启： 启动  
   测： 测试

**【学习目标】**

* 掌握NFS共享目录
* 理解NFS的用户映射
* 掌握NFS的访问控制
* 能够配置NFS的自动挂载

## 配置基本的NFS共享目录

1. 安装
2. root@nfs-server:~# aptitude install rpcbind nfs-kernel-server

nfs的网络文件系统的功能是由kernel提供，安装的软件包其实只是安装了nfs utils。

1. 启动
2. root@nfs-server:~# systemctl status rpcbind
3. root@nfs-server:~# systemctl status nfs-server
4. root@nfs-server:~# netstat -ntupl |grep rpc
5. root@nfs-server:~# rpcinfo -p
6. 配置
7. root@nfs-server:~# mkdir -p /nfs/public
8. root@nfs-server:~# echo "hello nfs" >> /nfs/public/nfs.txt
9. root@nfs-server:~# vim /etc/exports
10. /nfs/public \*(ro,no\_subtree\_check)
11. root@nfs-server:~# exportfs -av
12. exporting \*:/nfs/public
13. # 客户端测试
14. root@nfs-client:~# aptitude install nfs-client
15. root@nfs-client:~# showmount -e 192.168.38.103
16. Export list for 192.168.38.103:
17. /nfs/public \*
18. root@nfs-client:~# mount -t nfs 192.168.38.103:/nfs/public /mnt
19. root@nfs-client:~# ls /mnt
20. nfs.txt
21. root@nfs-client:~# cat /mnt/nfs.txt
22. hello nfs
23. root@nfs-client:~# df -h |grep mnt
24. 192.168.38.103:/nfs/public 19G 783M 17G 5% /mnt
25. root@nfs-client:~# mount |grep mnt
26. 192.168.38.103:/nfs/public on /mnt type nfs4 (rw,relatime,vers=4.2,rsize=131072,wsize=131072,namlen=255,hard,proto=tcp,port=0,timeo=600,retrans=2,sec=sys,clientaddr=192.168.38.102,local\_lock=none,addr=192.168.38.103)

## 配置NFS共享支持可读写的目录

配置NFS可以，要满足nfs的权限设置可写，被共享的文件夹本身可写。

1. root@nfs-client:/mnt# touch w.md
2. touch: cannot touch 'w.md': Read-only file system
3. /nfs/public \*(rw,sync,no\_subtree\_check)
4. root@nfs-server:~# exportfs -a
5. root@nfs-server:~# chmod 777 /nfs/public
6. root@nfs-server:~# ls -ld /nfs/public/
7. drwxrwxrwx 2 root root 4096 Apr 16 21:57 /nfs/public/

测试

1. root@nfs-client:/mnt# touch w.md
2. root@nfs-client:/mnt# ls -l w.md
3. -rw-r--r-- 1 nobody nogroup 0 Apr 17 02:31 w.md
4. root@nfs-server:/nfs/public# ls -l
5. total 4
6. -rw-r--r-- 1 root root 10 Apr 16 21:57 nfs.txt
7. -rw-r--r-- 1 nobody nogroup 0 Apr 17 02:31 w.md

## NFS客户端和服务器端的用户映射

NFS默认没有用户登录的机制。普通用户的映射关系如下：

1. demo@nfs-client:~$ touch /mnt/demo-file.db
2. demo@nfs-client:~$ ls -l /mnt/demo-file.db
3. -rw-r--r-- 1 demo demo 0 Apr 17 02:34 /mnt/demo-file.db
4. demo@nfs-client:~$ id demo
5. uid=1000(demo) gid=1000(demo) groups=1000(demo),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),108(netdev)
6. root@nfs-server:/nfs/public# ls -l demo-file.db
7. -rw-r--r-- 1 demo demo 0 Apr 17 02:34 demo-file.db
8. root@nfs-server:/nfs/public# id demo
9. uid=1000(demo) gid=1000(demo) groups=1000(demo),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),108(netdev)
10. root@nfs-client:/mnt# id zhaolei
11. uid=1001(zhaolei) gid=1001(zhaolei) groups=1001(zhaolei)
12. zhaolei@nfs-client:~$ touch /mnt/zhaolei-file.db
13. zhaolei@nfs-client:~$ ls -l /mnt/zhaolei-file.db
14. -rw-r--r-- 1 zhaolei zhaolei 0 Apr 17 02:36 /mnt/zhaolei-file.db
15. root@nfs-server:/nfs/public# id zhaolei
16. id: ‘zhaolei’: no such user
17. root@nfs-server:/nfs/public# ls -l zhaolei-file.db
18. -rw-r--r-- 1 user008 user008 0 Apr 17 02:36 zhaolei-file.db
19. root@nfs-server:/nfs/public# id user008
20. uid=1001(user008) gid=1001(user008) groups=1001(user008)
21. root@nfs-server:/nfs/public# ls -l houyi-file.db
22. -rw-r--r-- 1 3078 3078 0 Apr 17 02:39 houyi-file.db

结论：

1. 对于普通用户，NFS默认保留文件的拥有者身份（uid）。Linux系统对于用户的识别是通过uid来完成，那么在客户端和服务器端，同一个uid对应的用户名可能一样，也可能不一样，或者不存在。

2. 对于root用户，NFS默认将其映射为nobody,nogroup。

3. 可以通过配置，取消nfs对于root用户的匿名映射。

4. 可以通过配置，将所有的用户都映射成为某个指定的用户。

演示第3点：

1. /nfs/public \*(rw,sync,no\_subtree\_check,no\_root\_squash)

演示第4点：

1. /nfs/public \*(rw,sync,no\_subtree\_check,all\_squash,anonuid=1001,anongid=1001)

## NFS客户端访问控制

通过man exports 得到相关的实例：

1. # sample /etc/exports file
2. / master(rw) trusty(rw,no\_root\_squash)
3. /projects proj\*.local.domain(rw)
4. /usr \*.local.domain(ro) @trusted(rw)
5. /home/joe pc001(rw,all\_squash,anonuid=150,anongid=100)
6. /pub \*(ro,insecure,all\_squash)
7. /srv/www -sync,rw server @trusted @external(ro)
8. /foo 2001:db8:9:e54::/64(rw) 192.0.2.0/24(rw)
9. /build buildhost[0-9].local.domain(rw)

## NFS客户端实现自动挂载

1. 方式1： /etc/fstab
2. root@nfs-client:~# vim /etc/fstab
3. # nfs mount
4. 192.168.38.103:/nfs/public /mnt nfs4 defaults 0 0
5. root@nfs-client:~# umount /mnt
6. root@nfs-client:~# mount -a
7. root@nfs-client:~# mount |grep mnt
8. 192.168.38.103:/nfs/public on /mnt type nfs4 (rw,relatime,vers=4.2,rsize=131072,wsize=131072,namlen=255,hard,proto=tcp,port=0,timeo=600,retrans=2,sec=sys,clientaddr=192.168.38.102,local\_lock=none,addr=192.168.38.103)
9. # 方式2：autofs
10. root@nfs-client:~# aptitude install autofs
11. # 使用IP的访问nfs
12. root@nfs-client:~# vim /etc/auto.master
13. /net -hosts
14. root@nfs-client:~# systemctl reload autofs.service
15. # 测试：
16. root@nfs-client:~# mkdir /net/
17. root@nfs-client:~# cd /net/192.168.38.103
18. root@nfs-client:/net/192.168.38.103# ls nfs/public/
19. cc.db demo-file.db houyi-file.db nfs.txt root.db w.md zhaolei-file.db
20. root@nfs-client:/net/192.168.38.103# df -h |grep net
21. 192.168.38.103:/nfs/public 19G 783M 17G 5% /net/192.168.38.103/nfs/public
22. # 指定挂载点
23. root@nfs-client:~# vim /etc/auto.master
24. /mnt /etc/auto.misc
25. root@nfs-client:~# vim /etc/auto.misc
26. nfs -fstype=nfs4,defaults 192.168.38.103:/nfs/public
27. root@nfs-client:~# systemctl reload autofs.service
28. root@nfs-client:~# cd /mnt/nfs
29. root@nfs-client:/mnt/nfs# ls
30. cc.db demo-file.db houyi-file.db nfs.txt root.db w.md zhaolei-file.db
31. root@nfs-client:/mnt/nfs# df -h |grep 103
32. 192.168.38.103:/nfs/public 19G 783M 17G 5% /mnt/nfs

autofs服务实现，当你去访问nfs共享在autofs的配置文件中指定的挂载点时，会自动挂载。如果长时间（默认300s）没有数据交互，autofs会自动卸载，下一次访问的时候又重新挂载。

nfs属于一个分布收式的文件系统，支持多个客户端节点进行同时连接访问。nfs用户的映射关系。实现可写共享的两个层面的权限（服务，文件）。 自动挂载。

# SAMBA服务器

**【任务描述】**

网络附加存储之Samba是种用来让UNIX系列的操作系统与微软Windows操作系统的SMB/CIFS（Server Message Block/Common Internet File System）网络协议做链接的自由软件。

第三版不仅可访问及分享SMB的文件夹及打印机，本身还可以集成入Windows Server的网域，扮演为网域控制站（Domain Controller）以及加入Active Directory成员。简而言之，此软件在Windows与UNIX系列操作系统之间搭起一座桥梁，让两者的资源可互通有无。主要是用来Linux和Windows之间进行数据共享。Linux和Linux之间也可以。

**【学习目标】**

1. 掌握Linux下windows与Linux文件共享
2. 能够配置Samba匿名共享
3. 能够配置基于用户验证的Samba共享

## 基本的安装配置

1. 安装
2. root@smb-server:~# apt install samba
3. 配置
4. root@smb-server:~# vim /etc/samba/smb.conf
5. [public]
6. path = /samba/public
7. read only = yes
8. public = yes
10. root@smb-server:~# mkdir -p /samba/public
11. root@smb-server:~# touch /samba/public/smb.db
12. # 启
13. root@smb-server:~# systemctl restart smbd
14. root@smb-server:~# systemctl status smbd
15. root@smb-server:~# netstat -ntupl |grep -E '(s|n)mbd'
16. tcp 0 0 0.0.0.0:445 0.0.0.0:\* LISTEN 1657/smbd
17. tcp 0 0 0.0.0.0:139 0.0.0.0:\* LISTEN 1657/smbd
18. tcp6 0 0 :::445 :::\* LISTEN 1657/smbd
19. tcp6 0 0 :::139 :::\* LISTEN 1657/smbd
20. udp 0 0 192.168.38.255:137 0.0.0.0:\* 1539/nmbd
21. udp 0 0 192.168.38.103:137 0.0.0.0:\* 1539/nmbd
22. udp 0 0 0.0.0.0:137 0.0.0.0:\* 1539/nmbd
23. udp 0 0 192.168.38.255:138 0.0.0.0:\* 1539/nmbd
24. udp 0 0 192.168.38.103:138 0.0.0.0:\* 1539/nmbd
25. udp 0 0 0.0.0.0:138 0.0.0.0:\* 1539/nmbd
26. 测试

Windows下客户端测试，在文件浏览器中，输入\\192.168.38.103来进行访问。

在Linux下，在文件浏览器中，其他位置--连接到服务器，输入smb://192.168.38.103 访问。

在命令行下测试，

1. root@smb-client:~# aptitude install samba-client cifs-utils
2. root@smb-client:~# smbclient -L 192.168.38.103
3. public Disk
4. root@smb-client:~# mount -t cifs //192.168.38.103/public /mnt
5. Password for root@//192.168.38.103/public:
6. root@smb-client:~# ls /mnt
7. smb.db
8. root@smb-client:~# touch /mnt/test.db
9. touch: cannot touch '/mnt/test.db': Permission denied

也可以使用smbclient命令进行交互式登录samba文件系统。

## 创建匿名可写的smb共享

1. [public]
2. path = /samba/public
3. read only = no
4. # 或者 writable = yes
5. public = yes
6. root@smb-server:~# chmod 777 /samba/public
7. root@smb-server:~# systemctl reload smbd
8. root@smb-client:~# mount -t cifs -o remount //192.168.38.103/public /mnt
9. root@smb-client:/mnt# touch test.db
10. root@smb-client:/mnt# ls -l
11. total 0
12. -rw-r--r--+ 1 root root 0 Apr 18 03:20 smb.db
13. -rw-r--r--+ 1 nobody nogroup 0 Apr 18 03:44 test.db

## 基于用户验证的共享

1. [files]
2. path = /samba/files
3. #public = no
4. valid users = zhangxi, liupeng, @sales
5. write list = zhangxi

writable = yes 所有可访问的用户都有写入权限

write list = xxx 能访问的用户中只有xxx可以写入

两个选项只能二选一

1. root@smb-server:~# mkdir /samba/files
2. root@smb-server:~# chown zhangxi /samba/files
3. root@smb-server:~# useradd -m -s /sbin/nologin zhangxi
4. root@smb-server:~# smbpasswd -a zhangxi
5. root@smb-server:~# systemctl reload smbd

命令行下测试，

1. root@smb-client:~# mount -t cifs -o user=zhangxi //192.168.38.103/public /mnt
2. Password for zhangxi@//192.168.38.103/public: \*\*\*\*\*\*
3. root@smb-client:~# touch /mnt/linux.txt
4. root@smb-client:~# ls /mnt/linux.txt -l
5. -rw-r--r--+ 1 1002 1002 0 Apr 18 04:01 /mnt/linux.txt

其他用户liupeng , 组sales自行添加测试

用户共享时，smb用户首先是系统的用户；smb密码不是系统的密码，而是自己独立管理；smb用户和客户端没有关系；在Windows下，smb用户能否登陆跟Windows组策略有关系；smb的文件共享的读写权限常常配合Linux用户管理、文件权限、ACL等使用。

# FTP服务器

**【任务描述】**

文件传输协议（英文：File Transfer Protocol，缩写：FTP）是一个用于在计算机网络上在客户端和服务器之间进行文件传输的应用层协议。vsftpd是Linux下一种ftp服务器软件。还有其他的一些ftp软件，比如tftp,proftpd,pure-ftpd等等。FTP是一个跨平台的文件共享服务， 在windows , Linux, macOS都有服务器端和客户端软件的支持。

**【学习目标】**

1. 理解FTP服务器的文件传输过程
2. 能够配置匿名FTP文件共享服务
3. 能够配置基于用户验证的FTP文件共享服务
4. 管理FTP的客户端访问控制

## VSftpd的基本安装配置

装：

1. [root@data2 ~]# aptitude   vsftpd
2. [root@data2 ~]# aptitude  install  -y  vsftpd

配：

vsftpd 默认配置即是一个匿名的只读共享(在debian9系统中，默认没有开启匿名用户登录)。Debian9 开启匿名用户登录：

1. root@data2:~# vim /etc/vsftpd.conf
2. anonymous\_enable=YES

启：

1. [root@data2 ~]# systemctl   restart  vsftpd
2. [root@data2 ~]# ss  -ntupl  |grep  vsftpd
3. tcp    LISTEN     0      32       :::21                   :::\*                   users:(("vsftpd",pid=2781,fd=3))

特别说明：

FTP， 用ss命令看到的端口只是ftp的命令端口，而当ftp传输数据的时候，它会打开新的端口进行数据传输。

FTP的数据传输分为主动模式和被动模式。

FTP监听的21端口只是FTP命令端口，用于用户登录，执行命令等操作。而FTP服务器和客户端进行数据传输（下载或上传）是通过数据端口来完成。

在主动模式下：FTP会采用20号端口传输数据；

在被动模式下：FTP会采用大于1024的随机端口传输数据。

FTP传输数据采用主动模式还是被动模式是由FTP的客户端来决定。

在写防火墙规则时要特别注意。

测：

客户端的访问： windows ,linux,或其他平台的文件浏览器中输入，ftp://192.168.122.109

Linux命令行中，使用ftp IP地址 命令，用户ftp 或 anonymous代表匿名登录，在浏览器中也同样可以访问。

1. [root@365linux ~]# ftp  192.168.122.109
2. Connected to 192.168.122.109 (192.168.122.109).
3. 220 (vsFTPd 3.0.2)
4. Name (192.168.122.109:root): ftp
5. 331 Please specify the password.
6. Password:
7. 230 Login successful.
8. Remote system type is UNIX.
9. Using binary mode to transfer files.
10. ftp> ls
11. 227 Entering Passive Mode (192,168,122,109,154,218).
12. 150 Here comes the directory listing.
13. drwxr-xr-x    2 0        0               6 Aug 03  2015 pub
14. 226 Directory send OK.

默认情况下，匿名用户访问的共享目录是/srv/ftp/

1. [root@vhost01 ~]# ls /var/ftp/
2. pub

## 更改匿名访问的默认目录

1. [root@data2 ~]# vim /etc/vsftpd/vsftpd.conf
2. 在最后添加：
3. anon\_root=/share/ftp
4. [root@data2 ~]# mkdir   -p   /share/ftp
5. [root@data2 ~]# echo  "hello ftp"  >  /share/ftp/test.txt
6. [root@data2 ~]# systemctl    restart   vsftpd

测试略。

**FTP匿名用户上传文件**

默认情况下， vsftpd不允许在它共享根目录（/share/ftp）下上传文件, 只允许在/share/ftp/下创建一个子目录来进行上传。

找到配置文件中的`anon\_upload\_enable=YES`去掉`#`注释。

1. [root@data2 vsftpd]# mkdir   /share/ftp/upload
2. [root@data2 vsftpd]# chmod   777  /share/ftp/upload
3. [root@data2 vsftpd]# systemctl   restart  vsftpd

测试略。

上传的文件的用户和权限如下：

1. [root@data2 vsftpd]# ll  /share/ftp/upload/
2. 总用量 228
3. -rw------- 1 ftp ftp 231643 12月 12 15:32 Screenshot\_01-rhel7.2\_2016-11-29\_14:28:51.png

上传过程中，发现上传目录和删除上传的文件都是失败的，而上传文件是成功的。自己上传的文件，自己下载不了。

匿名用户上传目录

1. anon\_mkdir\_write\_enable=YES  （上传目录权限）
2. anon\_other\_write\_enable＝YES  （修改，删除，重命名文件的权限）

**匿名不能下载自己上传的文件**

原因是：anon\_world\_readable\_only＝YES（默认情况下，匿名用户只能够下载全世界（所有人）可读的文件，而匿名用户上传上去的文件，生成的权限默认是600，所以不能下载）解决的方法：

1. anon\_world\_readable\_only＝NO
2. 或者
3. anon\_umask＝022（默认是077）

不能对vsftpd匿名共享根目录/share/ftp的权限设为777 ，否则ftp访问不了。测试如下：

1. [root@365linux ~]# ftp  192.168.122.109
2. Connected to 192.168.122.109 (192.168.122.109).
3. 220 (vsFTPd 3.0.2)
4. Name (192.168.122.109:root): ftp
5. 331 Please specify the password.
6. Password:
7. 500 OOPS: vsftpd: refusing to run with writable root inside chroot()
8. Login failed.

扩展思考：如果非要对ftp的根目录上传文件，怎么做？

通过上面的配置和测试，发现vsftpd匿名用户要上传文件和目录有很多限制，需要一步步开放权限。基本原则就是服务本身开放可写的同时，要开放上传文件夹的权限。

## 本地用户验证登录

默认情况下，`vsftpd`使用的是`pam`（可热插拔的用户认证系统）的方式进行用户认证，而目前`Linux`系统本地用户登录采用的也pam管理。即`vsftpd`和`login`是同一套用户体系（`/etc/passwd /etc/shadow`）

直接用服务器端系统已存在的普通用户帐号进行登录`vsftpd`即可。

1. [root@data2 vsftpd]# useradd   liubei
2. [root@data2 vsftpd]# passwd   liubei
3. [root@365linux ~]# ftp  192.168.122.109
4. Connected to 192.168.122.109 (192.168.122.109).
5. 220 (vsFTPd 3.0.2)
6. Name (192.168.122.109:root): liubei
7. 331 Please specify the password.
8. Password:
9. 230 Login successful.
10. Remote system type is UNIX.
11. Using binary mode to transfer files.
12. ftp> pwd
13. 257 "/home/liubei"
14. ftp> help

默认本地用户登录到自己的家目录，可以进行上传下载的操作都可以。

默认情况下，本地用户可以登录FTP后，可以切换到别的系统目录去。这样很不安全。

1. ftp> cd   /etc/
2. 250 Directory successfully changed.
3. ftp> ls

限制方法：

1. chroot\_local\_user=YES　限制所有用户

或者：以下是限制部分用户，在`chroot\_list`

1. chroot\_list\_enable=YES
2. chroot\_list\_file=/etc/vsftpd/chroot\_list （在这个列表文件中写上要限制的用户名）

但是， 当你把用户限制在自己家目录时，对本地用户而言，家目录`/home/liubei`即是它的ftp的根目录，而本地用户对自己的家目录是可写的，那么和ftp的默认安全策略（不允许用户对ftp的根目录可写）相冲突，导致ftp无法登录。

如何解决：

* + - 1. **让本地的ftp根目录不可写**

但是让用户的家目录都不可写，不符合实际需求。改变ftp的本地用户根目录到另外一个只读目录。

1. local\_root=/var/srv
2. [root@data2 vsftpd]# mkdir  /var/srv
3. [root@data2 vsftpd]# touch   /var/srv/ftpchroot.txt
4. [root@365linux ~]# ftp  192.168.122.109
5. Connected to 192.168.122.109 (192.168.122.109).
6. 220 (vsFTPd 3.0.2)
7. Name (192.168.122.109:root): liubei
8. 331 Please specify the password.
9. Password:
10. 230 Login successful.
11. Remote system type is UNIX.
12. Using binary mode to transfer files.
13. ftp> ls
14. 227 Entering Passive Mode (192,168,122,109,183,136).
15. 150 Here comes the directory listing.
16. -rw-r--r--    1 0        0               0 Dec 12 08:09 ftpchroot.txt
17. 226 Directory send OK.
18. ftp> pwd
19. 257 "/"
20. ftp> cd  /etc/
21. 550 Failed to change directory.

可以看到，本地用户被限制在了自己的家目录。 有一个新的问题，`local\_root=/var/srv `对所有用户生效。 默认情况每个用户去到自己的家目录（/home下不同的目录），现在每个用户都是登录到`/var/srv` 。

思考： 如果我要做，用户被限制在自己的家目录，同时每个用户登录后去到不同的目录，怎么做？

提示：使用vsftpd的针对不同用户使用不同的子配置文件。（即对每个用户设置不同`local\_root`）(更简单的方法，是使用`user\_sub\_token=$USER` , `local\_root=/data/$USER`)

**2. 让ftp支持根目录可写（思考）**

1. allow\_writeable\_chroot=YES
2. allow\_writable\_chroot=YES

 在不同的vsftpd版里可能使用该选项，少了个e

## vsftpd对于本地用户黑白名单限制

在默认情况下，ftp允许所有的系统本地用户登录。如果有所限制，使用以下的两个文件（主要是`user\_list`）。

1. [root@data2 vsftpd]# pwd
2. /etc/vsftpd
3. [root@data2  vsftpd]# ls
4. ftpusers  user\_list

`user\_list`既可以作为白名单，也可以做为黑名单，取决于主配置文件中的`userlist\_deny`的选项。默认情况下，主配置文件中，`userlist\_deny=YES`。那么，在`usre\_list`文件中用户将不能访问ftp，即黑名单。

如果`userlist\_deny=NO`，那么，在`usre\_list`文件中用户将可以访问ftp，即白名单。

同时还检查用户名是否在`ftpusers`文件中，主要放了一系统服务的用户，这个文件中的用户永远都不能访问ftp，不管`user\_list`是白还是黑。

## FTP托管模式

如果`vsftpd`的服务不是一个频繁使用的服务，没有必要长期运行在系统中，占用系统资源，使用托管模式，有访问的时候才启动。

在Linux系统使用`xinetd`服务托管其他服务，而`xinetd`有很多的安全配置选项，使用得服务更安全。

**xinetd服务**

安装：

1. [root@data2 vsftpd]# aptitude  search  xinetd
2. [root@data2 vsftpd]# aptitude   install xinetd

配置：

1. [root@data2 ~]# systemctl   stop vsftpd
2. [root@data2 ~]# systemctl   disable  vsftpd
3. [root@data2 ~]# vim /etc/vsftpd.conf
4. listen=NO
5. listen\_ipv6=NO
6. /etc/xinetd.conf
7. [root@data2 ~]# vim  /etc/xinetd.d/vsftpd
8. service ftp
9. {
10. disable             = no
11. socket\_type         = stream
12. wait                = no
13. nice                = 10
14. user                = root
15. server              = /usr/sbin/vsftpd
16. server\_args         = /etc/vsftpd/vsftpd.conf
17. }

启动：

1. [root@data2 ~]# systemctl restart  xinetd
2. [root@data2 ~]# ss  -ntupl |grep  xinetd
3. tcp    LISTEN     0      64       :::21                   :::\*                   users:(("xinetd",pid=3232,fd=5))

测：略

用`xinetd`来托管服务的好处是，可以利用`xinetd`守护进程的特性，访问控制，流量限制，日志增强，应用防火墙等等。还可以节省系统开销。对于访问量不大不频繁的ftp服务器，可以使用托管模式。

# IPTABLES防火墙

**【任务描述】**

iptables 是一个配置 Linux 内核 防火墙 的命令行工具，是 netfilter 项目的一部分。术语 iptables 也经常代指该内核级防火墙。

iptables是运行在用户空间的应用软件，通过控制Linux内核netfilter模块，来管理网络数据包的处理和转发。通常iptables需要内核模块支持才能运行，此处相应的内核模块通常是Xtables。

iptables 用于 ipv4，ip6tables 用于 ipv6。

**【学习目标】**

1. 掌握Linux防火墙的基本概念
2. 能够编写流量过滤型防火墙规则
3. 能够编写流量转发型防火墙规则

## 理解Linux防火墙基本概念

Linux下的防火墙IPTABLES基本概念。

iptables、ip6tables等都使用Xtables框架。存在“表（tables）”、“链（chain）”和“规则（rules）”三个层面。

每个“表”指的是不同类型的数据包处理流程，如filter表表示进行数据包过滤，而nat表针对连接进行地址转换操作。每个表中又可以存在多个“链”，系统按照预订的规则将数据包通过某个内建链，例如将从本机发出的数据通过OUTPUT链。在“链”中可以存在若干“规则”，这些规则会被逐一进行匹配，如果匹配，可以执行相应的动作，如修改数据包，或者跳转。跳转可以直接接受该数据包或拒绝该数据包，也可以跳转到其他链继续进行匹配，或者从当前链返回调用者链。当链中所有规则都执行完仍然没有跳转时，将根据该链的默认策略（“policy”）执行对应动作；如果也没有默认动作，则是返回调用者链。

* 1. **filter表**

filter表是默认的表，如果不指明表则使用此表。其通常用于过滤数据包。其中的内建链包括：

INPUT，输入链。发往本机的数据包通过此链。

OUTPUT，输出链。从本机发出的数据包通过此链。

FORWARD，转发链。本机转发的数据包通过此链。

* 1. **nat表**

nat表如其名，用于"网络地址转换")操作。其中的内建链包括：

PREROUTING，路由前链，在处理路由规则前通过此链，通常用于目的地址转换（DNAT）。

POSTROUTING，路由后链，完成路由规则后通过此链，通常用于源地址转换（SNAT）。

OUTPUT，输出链，类似PREROUTING，但是处理本机发出的数据包。

* 1. **mangle表**

mangle表用于处理数据包。其和nat表的主要区别在于，nat表侧重连接而mangle表侧重每一个数据包。其中内建链列表如下：

PREROUTING

OUTPUT

FORWARD

INPUT

POSTROUTING

* 1. **raw表**

raw表用于处理异常，有如下两个内建链：

PREROUTING

OUTPUT

## 访问防火墙所在的服务器上的应用

1. root@apache-server:~# iptables -t filter -A INPUT -s 192.168.38.102 -p tcp --dport 80 -j ACCEPT
2. root@apache-server:~# iptables -nvL
3. Chain INPUT (policy ACCEPT 36 packets, 2304 bytes)
4. pkts bytes target prot opt in out source destination
5. 0 0 ACCEPT tcp -- \* \* 192.168.38.102 0.0.0.0/0 tcp dpt:80
6. root@apache-server:~# iptables -P INPUT DROP
7. root@apache-server:~# iptables -A INPUT -j REJECT --reject-with icmp-host-unreachable
8. root@apache-server:~# iptables -A OUTPUT -j REJECT --reject-with icmp-host-unreachable
9. iptables -I OUTPUT -s 192.168.38.101 -p tcp --sport 22 -j ACCEPT
10. iptables -I OUTPUT -s 192.168.38.101 -p tcp --sport 80 -j ACCEPT
11. root@apache-server:~# iptables -I OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
12. root@apache-server:~# iptables -nL --line-number
13. Chain INPUT (policy DROP)
14. num target prot opt source destination
15. 1 ACCEPT tcp -- 192.168.38.102 0.0.0.0/0 tcp dpt:80
16. 2 ACCEPT tcp -- 0.0.0.0/0 192.168.38.101 tcp dpt:22
17. 3 REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-unreachable
18. Chain FORWARD (policy ACCEPT)
19. num target prot opt source destination
20. Chain OUTPUT (policy ACCEPT)
21. num target prot opt source destination
22. 1 ACCEPT all -- 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
23. 2 ACCEPT tcp -- 192.168.38.101 0.0.0.0/0 tcp spt:80
24. 3 ACCEPT tcp -- 192.168.38.101 0.0.0.0/0 tcp spt:22
25. 4 REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-unreachable
26. root@apache-server:~# iptables -D OUTPUT 3
27. root@apache-server:~# iptables -D OUTPUT 2
28. root@apache-server:~# iptables -nL --line-number
29. root@apache-server:~# apt -y install iptables-persistent

保存和重载（开机加载）防火墙规则的方法：

1. iptables-save , iptables-restore
2. systemctl restart netfilter-persistent
3. netfilter-persistent (推荐)

## 内网用户通过防火墙转发（代理）上网

1. root@gateway:~# ip addr show
2. 1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
3. link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
4. inet 127.0.0.1/8 scope host lo
5. valid\_lft forever preferred\_lft forever
6. inet6 ::1/128 scope host
7. valid\_lft forever preferred\_lft forever
8. 2: ens33: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000
9. link/ether 00:0c:29:33:73:4b brd ff:ff:ff:ff:ff:ff
10. inet 192.168.38.103/24 brd 192.168.38.255 scope global ens33
11. valid\_lft forever preferred\_lft forever
12. inet6 fe80::20c:29ff:fe33:734b/64 scope link
13. valid\_lft forever preferred\_lft forever
14. 3: ens37: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000
15. link/ether 00:0c:29:33:73:55 brd ff:ff:ff:ff:ff:ff
16. inet 172.16.100.254/24 brd 172.16.100.255 scope global ens37
17. valid\_lft forever preferred\_lft forever
18. inet6 fe80::20c:29ff:fe33:7355/64 scope link
19. valid\_lft forever preferred\_lft forever
20. root@client:~# ip a
21. 1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
22. link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
23. inet 127.0.0.1/8 scope host lo
24. valid\_lft forever preferred\_lft forever
25. inet6 ::1/128 scope host
26. valid\_lft forever preferred\_lft forever
27. 2: ens33: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000
28. link/ether 00:0c:29:d1:a5:54 brd ff:ff:ff:ff:ff:ff
29. inet 172.16.100.101/24 brd 172.16.100.255 scope global ens33
30. valid\_lft forever preferred\_lft forever
31. inet6 fe80::20c:29ff:fed1:a554/64 scope link
32. valid\_lft forever preferred\_lft forever
33. root@client:~# route -n
34. Kernel IP routing table
35. Destination Gateway Genmask Flags Metric Ref Use Iface
36. 0.0.0.0 172.16.100.254 0.0.0.0 UG 0 0 0 ens33
37. 172.16.100.0 0.0.0.0 255.255.255.0 U 0 0 0 ens33
38. root@gateway:~# echo 1 > /proc/sys/net/ipv4/ip\_forward
39. root@gateway:~# vim /etc/sysctl.conf
40. net.ipv4.ip\_forward=1
41. root@gateway:~# iptables -t nat -A POSTROUTING -s 172.16.100.0/24 -j SNAT --to-source 192.168.38.103
42. #或者 iptables -t nat -A POSTROUTING -s 172.16.100.0/24 -j MASQUERADE
43. root@client:~# ping www.baidu.com
44. PING www.a.shifen.com (119.75.213.51) 56(84) bytes of data.
45. 64 bytes from 119.75.213.51 (119.75.213.51): icmp\_seq=1 ttl=127 time=38.3 ms
46. root@gateway:~# iptables -A FORWARD -j DROP
47. root@gateway:~# iptables -I FORWARD -s 172.16.100.0/24 -j ACCEPT
48. root@gateway:~# iptables -I FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
49. root@gateway:~# iptables -nL FORWARD
50. Chain FORWARD (policy ACCEPT)
51. target prot opt source destination
52. ACCEPT all -- 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
53. ACCEPT all -- 172.16.100.0/24 0.0.0.0/0
54. DROP all -- 0.0.0.0/0 0.0.0.0/0

记得保存。

## 来自于外网的客户端访问防火墙后面的内网中服务器的应用

内核转发要打开

1. root@gateway:~# iptables -t nat -F
2. root@gateway:~# iptables -F
3. root@gateway:~# iptables -t nat -A PREROUTING -d 192.168.38.103 -p tcp --dport 80 -j DNAT --to 172.16.100.101

根据需要去配置FORWARD的规则。

1. 本章节部分内容来自维基百科和其他互联网内容（如发行版官网介绍等）。 [↑](#footnote-ref-1)