

21世纪高等教育计算机规划教材

COMPUTER

Linux 编程基础

Linux Fundamentals

李养群 王攀 周梅 编著

实例丰富，面向基础

概念清晰，深入浅出

体系完整，注重细节



 人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等教育计算机规划教材

COMPUTER

Linux 编程基础

Linux Fundamentals

李养群 王攀 周梅 编著



人民邮电出版社

北京

图书在版编目 (CIP) 数据

Linux编程基础 / 李养群, 王攀, 周梅编著. -- 北京: 人民邮电出版社, 2015.2
21世纪高等教育计算机规划教材
ISBN 978-7-115-38059-3

I. ①L… II. ①李… ②王… ③周… III. ①Linux操作系统—程序设计—高等学校—教材 IV. ①TP316.89

中国版本图书馆CIP数据核字(2015)第011162号

内 容 提 要

本书是 Linux 编程方面的入门教材, 主要针对 Linux 基础读者。本书介绍了 Linux 编程方面的基础知识, 主要包括三部分: Linux 基本概念及 Linux 基本操作; Linux 编程环境; Linux 编程。其中, Linux 基本概念及 Linux 基本操作主要介绍了 Linux 操作系统基本概念和特点、Linux 常用命令和 VI 编辑器的基本使用; Linux 编程环境主要介绍了 Linux 的 Shell 编程、GCC/GDB 的使用及 Make 工具的使用; Linux 编程主要介绍了 Linux 文件和目录操作、进程管理、进程间通信、信号及信号处理、多线程编程、网络编程及数据库编程等内容。最后, 还提供了与 Linux 编程相关的实验指导手册。本书通过丰富实例, 详细展示了 Linux 基本知识的使用方法, 以帮助读者进一步深入学习 Linux 技术。

本书可作为高等学校本、专科工科类专业的教材, 也可作为编程爱好者的入门参考书。

◆ 编 著 李养群 王攀 周梅

责任编辑 武恩玉

责任印制 沈蓉 彭志环

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京圣夫亚美印刷有限公司印刷

◆ 开本: 787×1092 1/16

印张: 17.25

2015年2月第1版

字数: 452千字

2015年2月北京第1次印刷

定价: 39.80元

读者服务热线: (010) 81055256 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

前 言

随着开源软件和技术的发展，Linux 操作系统得到广泛的应用，为企业提供各种服务的服务器以及嵌入式开发应用等领域。Linux 操作系统的广泛应用，使得对其进行有效管理和在其之上进行应用开发成为一种重要技能。本书主要介绍 Linux 操作系统的基础操作和 Linux 的基础应用开发，初学者通过本书的学习，可熟悉和掌握 Linux 的基本概念和开发技能。

本书的编写集中了作者多年在“Linux 编程”教学过程中的一些经验。本课程是一门实践性和应用性很强的课程。在教学过程中，我们发现虽然学生已经学习了操作系统等课程，但在动手能力方面仍有所欠缺，无法将所学理论真正地 and 编程开发结合起来。另外，学生对 Windows 的所见即所得的操作方式非常熟练，但对于 Linux 的命令行操作模式非常陌生。同时，在使用现有的教材过程中，我们发现教材中的知识点过于分散，无法在课时有限的情况下，掌握 Linux 的基本操作和基本编程知识。还有就是，部分教材内容过于冗杂，部分内容更适合于自学或者后续进一步学习。也有些教材知识点较深，实例较少，不利于基本知识的掌握和应用。

本书着眼于 Linux 基本操作、Linux 编程环境和 Linux 编程基本开发三方面的内容，立足于全面介绍 Linux 开发所需相关的基础知识，同时着眼于提高学生的动手能力，进而加强对基本概念的认识。书中对主要概念和知识点都进行了实例验证和分析。本书按照基本概念介绍、使用方法说明、详细案例分析和结果分析的思路进行编写，便于在学习过程中、在了解知识的基础上进行理解和实践。

本书主要针对 Linux 操作系统的初学者和对 Linux 开发有兴趣准备入门的读者。通过本书，读者可对 Linux 编程有基本的认知，并能掌握基本的应用知识。本书内容相对独立，可根据自己需要选择部分章节进行学习，建议初学者先从第 1 章开始阅读。

作者在编写本书过程中参考了部分书籍和互联网上的资料，学到了很多知识并从中受益，在此表示衷心的感谢。李养群负责第 1 章~第 8 章内容的编写，王攀负责编写第 9 章~第 11 章的内容，周梅负责资料的搜集整理和本书的校订。在编写本书期间，作者还得到了很多同事和人民邮电出版社编辑的帮助，在此表示衷心感谢。

虽然作者多年从事 Linux 方面的教学工作，但由于时间仓促，加之水平有限，书中不足之处在所难免，敬请读者批评指正。如有什么问题，可通过电子邮箱 sxlyqq@sina.com 联系，欢迎来信与作者进行进一步的交流。

编 者

2014 年 10 月

目 录

第 1 章 Linux 基础 1

1.1 什么是 Linux? 1

 1.1.1 Linux 操作系统特点 1

 1.1.2 Linux 操作系统组成部分 1

1.2 Linux 版本及 Fedora 操作系统 2

 1.2.1 Linux 内核版本 2

 1.2.2 Linux 发行版 2

 1.2.3 Fedora 操作系统 3

 1.2.4 Fedora 发行方式 3

 1.2.5 Fedora 系统安装基本要求 3

1.3 Fedora 20 的安装 4

 1.3.1 虚拟机下的 Fedora 20 的安装准备 4

 1.3.2 Fedora 20 安装 5

1.4 Linux 常用命令 7

 1.4.1 Linux 命令执行方法 8

 1.4.2 Linux 常用各种命令 8

1.5 VI 编辑器的使用 24

1.6 POSIX 标准和 LSB 标准 27

总结 28

习题 28

第 2 章 Linux 编程环境 29

2.1 GCC 编译器的使用 29

 2.1.1 GCC 编译器简介 29

 2.1.2 GCC 常用选项: 预处理控制 30

 2.1.3 GCC 常用选项: 编译及警告信息控制选项 31

 2.1.4 GCC 常用选项: C 语言标准控制选项和程序调试及优化选项 33

 2.1.5 GCC 常用选项: 搜索路径控制和 GCC 链接选项 34

 2.1.6 利用 GCC 创建库文件 34

2.2 GDB 调试器的使用 37

 2.2.1 GDB 调试器的使用 37

 2.2.2 GDB 调试器使用实例 43

2.3 Make 工具 44

 2.3.1 第一个 Makefile 文件 45

 2.3.2 Makefile 编写规则 46

 2.3.3 Make 的基本工作原理及过程 46

 2.3.4 Makefile 文件 47

总结 59

习题 60

第 3 章 Shell 编程 61

3.1 Shell 基础 61

 3.1.1 Shell 交互方式 61

 3.1.2 Shell 基本功能 61

3.2 Bash 编程 62

3.3 Shell 中的特殊字符 63

 3.3.1 转义符 “\” 63

 3.3.2 单引号 64

 3.3.3 双引号 64

 3.3.4 命令替换符号 65

3.4 Shell 变量 65

 3.4.1 Shell 用户变量定义 65

 3.4.2 Shell 环境变量 66

 3.4.3 Shell 内部变量 67

 3.4.4 Shell 参数扩展 67

 3.4.5 Shell 变量的算术扩展 68

 3.4.6 条件表达式 69

 3.4.7 Shell 字符串操作 71

3.5 Shell 控制语句 73

 3.5.1 条件语句 73

 3.5.2 循环语句 75

3.6 Shell 其他命令 77

 3.6.1 管道命令 77

 3.6.2 重定向命令 78

 3.6.3 echo 命令 79

 3.6.4 shift 命令 79

3.7 Shell 函数 80

3.8 Shell 数组	81	第 6 章 信号及信号处理	126
3.9 Shell 中 Dialog 工具	82	6.1 信号的基本概念	126
3.10 Bash 调试	84	6.1.1 信号的使用和产生	126
总结	85	6.1.2 信号的状态	127
习题	86	6.2 信号的分类	127
第 4 章 文件 I/O 操作	87	6.2.1 可靠与不可靠信号	127
4.1 概述	87	6.2.2 实时信号与非实时信号	127
4.2 文件 I/O 操作	87	6.3 信号的处理	128
4.2.1 文件的创建	88	6.3.1 signal 信号处理机制	129
4.2.2 文件的打开及关闭	88	6.3.2 sigaction 信号处理机制	131
4.2.3 文件的读取/写入	89	6.4 信号发送函数	134
4.2.4 文件的定位	90	6.5 可重入函数	135
4.2.5 文件删除	90	6.6 父子进程的信号处理	136
4.2.6 文件描述符属性控制 fcntl	91	6.7 信号处理机制的应用	137
4.2.7 文件操作实例	95	6.8 系统定时信号	138
4.3 目录	97	6.8.1 睡眠函数	138
4.3.1 目录概述	97	6.8.2 计时器	139
4.3.2 Linux 文件系统 ext2 基本结构	98	总结	140
4.3.3 与目录有关的系统调用	100	习题	141
4.4 文件与目录的属性	102	第 7 章 进程间通信	142
4.4.1 获得文件或目录属性	102	7.1 进程间通信基本概念	142
4.4.2 文件或目录的模式	103	7.1.1 进程通信的作用	142
4.4.3 符号链接	105	7.1.2 进程通信的实现和方法	142
4.4.4 文件属性的更改	107	7.2 管道通信	143
4.5 标准文件 I/O	108	7.2.1 无名管道	144
4.6 处理系统调用中的错误	109	7.2.2 管道与重定向	146
总结	110	7.2.3 popen 的介绍	149
习题	110	7.2.4 命名管道	151
第 5 章 Linux 进程管理	112	7.3 System V 信号量	156
5.1 进程基本概念	112	7.3.1 信号量的用法	156
5.2 进程创建和命令执行	113	7.3.2 信号量实例	157
5.3 进程退出	117	7.4 POSIX 有名信号量	160
5.3.1 守护进程	118	7.4.1 POSIX 有名信号量的使用	160
5.3.2 僵尸进程	120	7.4.2 有名信号量实例	161
5.3.3 进程退出状态	121	7.5 共享内存	163
5.4 进程开发实例	123	7.5.1 共享内存步骤	164
总结	124	7.5.2 System V 共享内存 API	164
习题	124	7.5.3 共享内存实例	165
		7.5.4 mmap 共享内存机制	168

7.6 消息队列	171	9.2.3 Socket stream 服务	208
7.6.1 消息队列的实现原理	171	9.2.4 Socket 数据包服务	211
7.6.2 消息队列系统调用	171	9.2.5 Socket 原始套接字服务	213
7.6.3 消息队列实例	173	9.3 Linux 网络编程高级 I/O	218
总结	175	9.3.1 Socket 阻塞/非阻塞方式	218
习题	175	9.3.2 非阻塞 Socket 用法	220
第 8 章 多线程编程	177	9.3.3 Socket 与多路复用	222
8.1 多线程概念	177	9.4 Linux 网络并发编程	235
8.2 线程状态与线程编程	178	总结	239
8.2.1 线程的创建和参数传递	178	习题	239
8.2.2 线程终止 pthread_exit	180	第 10 章 Linux 下的数据库编程	240
8.2.3 线程挂起 pthread_join	183	10.1 MySQL 数据库简介	240
8.2.4 线程其他相关系统调用	184	10.1.1 Linux 数据库编程应用	240
8.3 线程的同步与互斥	184	10.1.2 MySQL API 的两种形式	241
8.3.1 互斥量	184	10.1.3 MySQL C API 的使用	241
8.3.2 互斥量的使用	185	10.2 Linux 数据库编程基本方法	241
8.3.3 信号量	188	10.3 MySQL 数据库数据结构及 API	242
8.3.4 信号量的使用方法	188	10.3.1 数据结构	242
8.3.5 条件变量	192	10.3.2 MySQL 操作 API	242
8.3.6 条件变量的使用	192	10.4 MySQL 数据库编程实例	248
总结	195	总结	253
习题	195	习题	253
第 9 章 Linux 网络编程	197	附录 Linux 编程基础实验	254
9.1 计算机网络概述	197	实验一 Linux 基本命令使用	
9.1.1 计算机网络的组成及特点	197	(验证性实验)	254
9.1.2 计算机网络协议	197	实验二 Linux Shell 编程	
9.1.3 网络协议分层	198	(设计性实验)	258
9.1.4 TCP/IP	198	实验三 Makefile 实验	
9.1.5 Client/Server 模型	199	(验证性和设计性)	259
9.1.6 Linux 网络编程概述	199	实验四 GCC/GDB 实验	262
9.1.7 网络协议栈	199	实验五 Linux 文件系统编程	264
9.2 Socket 编程	200	实验六 Linux 多进程与进程间通信	267
9.2.1 什么是 Socket?	200	参考文献	268
9.2.2 Socket 编程基本系统调用	201		

第 1 章

Linux 基础

Linux 操作系统由于其具有开源、开放和免费的特点，得到了广泛的应用，在企业的服务器领域，如 Web 服务器、FTP 服务器、数据库服务器等都有大量的使用。基于 Linux 的应用开发需求也越来越多，因而在进行 Linux 编程开发之前，对 Linux 操作系统需要有一个基本的了解。本章对 Linux 的基本知识、安装方法和常用命令做了基本介绍，以为后面的编程开发学习提供基础。

1.1 什么是 Linux?

1.1.1 Linux 操作系统特点

Linux 内核是一位名叫 Linus Torvalds 的芬兰赫尔辛基大学的学生于 1991 年完成开发的。Linus 当时使用 Minix（用来示范教学的一种操作系统）学习操作系统。期间，他自己编写了一个操作系统原型，然后将其放在因特网上公开，后来许多人陆续参与到内核的改进、扩充以及完善之中，从而促进了 Linux 的快速发展。Linux 是一种开放源码、能够免费使用及自由传播的类似于 UNIX 操作系统。它可运行于各种平台（如 Intel x86 系列、SUN 的 Sparc 工作站等不同硬件平台）之上。因为它不但具有 UNIX 系统的良好性能（如稳定性、安全性），同时还具有很好的性价比，所以广泛被应用于企业、政府、教育等部门中。同时，由于其具有开源和免费的特点，也越来越广泛地应用于各种嵌入式系统中。

1.1.2 Linux 操作系统组成部分

Linux 内核加上来自于开源组织 GNU（GNU's Not UNIX）的各种应用软件及开发库组成了 Linux 操作系统，如图 1.1 所示。Linux 内核负责对计算机硬件的管理并通过 Shell 向用户提供使用计算机硬件资源的接口。应用层中包含了多种应用软件，例如，开发工具包括 Linux 操作系统中的 GCC 编译器、Make 工具、GDB 调试器、Bash Shell 环境、Emacs 编辑器等，还包括用户使用的一些娱乐软件、办公软件等。Linux 是一个类 UNIX 操作系统，因此它上面的很多软件可以不加改变地运行在 UNIX 操作系统中。同样，UNIX 操作系统中的软件也可以以二进制的形式运行在 Linux 操作系统中。与 UNIX 一样，Linux 操作系统具有多用户、多任务、多进



图 1.1 Linux 操作系统

程、多线程、实时性好等特点，它比 UNIX 更加灵活，允许用户针对内核进行定制。

Linux 操作系统主要由内核、Shell、文件结构和实用工具组成。内核是 Linux 的心脏，是在系统引导时装入系统执行的应用程序。它为应用程序访问底层硬件各种资源提供了统一的接口，使得应用程序开发人员不需关心太多的物理硬件的细节。内核主要包括了进程管理、内存管理、硬件设备驱动管理、文件系统驱动、网络管理等部分。因此内核是操作系统的核心部分，负责管理各种资源。Shell 是用户与内核交互操作的接口，它接收用户的命令然后传递给内核，内核执行命令并将执行结果通过该 Shell 返回给用户。Shell 有自己的脚本语言，用户可根据需要利用 Shell 脚本执行各种任务。Shell 既是一种命令解释器，也是一种编程语言。它有自身的语言结构，例如循环、条件语句等。Shell 有很多种版本，目前主要包括如下版本的 Shell：

Bourne Shell，源于 UNIX 早期版本的 Shell；

Bash，又名 Bourne Again Shell，来自 GNU 项目，它是 Linux 的主要 Shell，可以免费获取其源代码，它与 Korn Shell 有许多相似的地方；

Korn Shell，它是对 Bourne Shell 的发展；

C Shell，是 SUN 公司 Shell 的 BSD 版本。

1.2 Linux 版本及 Fedora 操作系统

经过多年的发展，Linux 产生了多种版本。关于版本一般而言主要分别指内核版本和发行版本。内核版本是指 Linux 最核心部分的版本，内核在原作者的领导下已经发展到 3.x 版本。Linux 的开发版本是指由开发商或者相关组织在内核的基础上加上定制或者自行开发的软件、文档构成的商业发行版本。它是各大 IT 企业在开源 Linux 内核的基础上通过定制，包含了自身的一些管理工具和应用软件，从而发展出具有不同特色的 Linux 操作系统发行版。

1.2.1 Linux 内核版本

内核版本指的是在 Linus Torvalds 领导下的开发小组开发出来的系统内核版本号。内核版本号一般表示为： $x.y.z$ 。其中第一位表示主版本号，表示有较大的改动。第二位表示内核的稳定性版本号。如果是偶数（如 0、2、4 等）表示为稳定版本号，若为奇数（如 1、3、5 等）表示是处于开发过程中的版本号，一般不太稳定，有可能包含致命错误。第三位为修订号，表明是对这一版本的增补。目前（2014 年 9 月）最新的版本号是 3.12.27，这表明该版本是一个稳定的版本。

1.2.2 Linux 发行版

当前有很多公司开发了自己的发行版，主要有 Red Hat Linux、Debian Linux、Ubuntu Linux、Mandriva Linux 以及国产的 Turbo Linux、红旗 Linux、CLEEX For Linux、Xteam Linux 等。下面主要介绍并且使用 Fedora 操作系统。之所以介绍 Fedora 操作系统，是因为与 Ubuntu 操作系统相比，Fedora 提供了更为完整的开发工具和软件，一般情况下，当安装了开发软件时，就可方便地进行软件开发。而 Ubuntu 的强项则在于桌面系统和个人用户使用的便捷性方面，在某些情况下，在开发相关软件时，需要额外的软件安装和管理。

1.2.3 Fedora 操作系统

Fedora 操作系统由来自全球开源社区开发人员共同参与的 Fedora 项目所开发，Fedora 由红帽（Red Hat）赞助，后者是目前世界上最受信赖的开源技术提供商。它是可自由使用的开源操作系统，具有稳定、安全及易使用等特点。当前 Fedora 得到了广泛的使用，使用者包括 Linux 内核的作者 Linus Torvalds、美国国家航空和宇宙航行局系统、世界排名第一的超级计算机系统 Roadrunner 等。Fedora 发行版操作系统具有如下几个特点。

- (1) 100% 自由软件和开源：Fedora 是 100% 免费并且只包含自由和开源软件。
- (2) 丰富的应用软件和系统工具：提供了超过 10000 的软件和工具。
- (3) 大量的用户和开放成熟的社区：具有超过百万的用户在使用 Fedora。
- (4) 强大、稳定、可靠的操作系统：它是 Red Hat 开发的企业版 Linux 操作系统 Red Hat Enterprise 的基础。

Fedora 20 是目前最新和稳定的版本。相比较之前的版本，其所增加的新的特征主要包括：

- 将 ARM 作为主要支持架构，意味着绝大多数软件包必须可以在 ARM 上构建通过。
- 强化 Network Manager，允许其在命令行界面上创建和编辑链接，且支持接口桥接和绑定。
- 不再默认安装 Sendmail，systemd journal 将替代 syslog 成为默认日志记录系统。
- 可以在图形化管理工具中创建虚拟机快照，允许在 x86 主机上运行 ARM 镜像。
- 将 Ruby on Rails 升级至 4.0，Perl 升级至 5.18。
- 搭载 GNOME 3.10 预览版，除了新的音乐和地图程序，还包括 Wayland 支持。
- 集成 KDE 4.11 桌面环境。

1.2.4 Fedora 发行方式

Fedora 操作系统的主要发行方式包括如下几种：

- (1) Fedora DVD/CD，包含了所有主要软件包的 DVD 或 CD 套装，大约 4.3GB；
- (2) Live 光盘，CD 或 DVD 大小的光盘镜像，可用于创建 Live CD 或从 USB 设备启动，并可选安装到硬盘，大约 900MB；
- (3) 最小 CD，用于通过 HTTP、FTP 或 NFS 安装，大于 300MB。

其中 Fedora DVD 的发行方式最全面，所占用空间最大。而 Live 光盘主要为了方便携带，主要提供了 Fedora 核心功能和核心工具，有些软件需要通过 Fedora DVD 中的软件包安装或者通过网络下载的方式安装。而网络形式的安装为了节省下载软件的时间，因此其只提供最核心和最基本的功能。

1.2.5 Fedora 系统安装基本要求

Fedora 20 的安装需求首先是硬件需求，这里我们主要指基于 Intel 平台的硬件环境需求，包括以下几点：

- 64 位 Intel 兼容 PC；
- 1GB 内存（RAM）；
- 至少 10GB 的硬盘空间。

Fedora 的安装主要有两种方式：一种是直接安装在硬盘中，作为独立的操作系统直接运行在主机之上；另外一种方式是安装在虚拟机中，作为虚拟操作系统运行在宿主机操作系统之上。

后者较为适合初学者使用，这种安装可在不影响主机已有的操作系统以及之上的数据安全的基础上运行 Fedora 操作系统，Fedora 的运行与安装对宿主机操作系统没有影响。

1.3 Fedora 20 的安装

首先，从因特网上下载 Fedora 20 桌面版安装文件，根据需要选择 DVD/CD 发行方式或者 LiveCD 或者最小 CD 等，这里选择 LiveCD 形式，大约 900MB。在下载时，应注意下载的版本和所用主机 CPU 型号（主要看是 32 位还是 64 位，是否为主机的芯片类型，例如 Intel 或者 ARM）是否一致，如果不一致，则无法安装。采用虚拟机下安装 Linux 操作系统是初学者最为常用的方式，因此这里主要介绍在虚拟机环境下的 Fedora 安装。

1.3.1 虚拟机下的 Fedora 20 的安装准备

在虚拟机软件 VMWare 安装好之后，在安装 Fedora 20 之前，首先需要为其建立一个虚拟机并准备好安装文件。主要过程如下。

(1) 运行虚拟机软件。选择“文件→新建→虚拟机”菜单，打开创建新的虚拟机向导，选择典型配置，单击“下一步”按钮，显示向导界面如图 1.2 所示。在图中选择客户机操作系统为 Linux，而版本的选择，则是根据主机的 CPU 类型选择对应的版本即可，注意查看你主机的 CPU 类型。

(2) 输入虚拟机名称及虚拟机文件存储的地方，如图 1.3 所示。每个虚拟机文件代表一个虚拟机上所安装的操作系统，例如，本例中的 H:\Virtual Machines\Fedora20.vmdx 文件就代表本次安装的 Fedora 操作系统，以后通过虚拟机打开该文件就可运行 Fedora 20 操作系统。

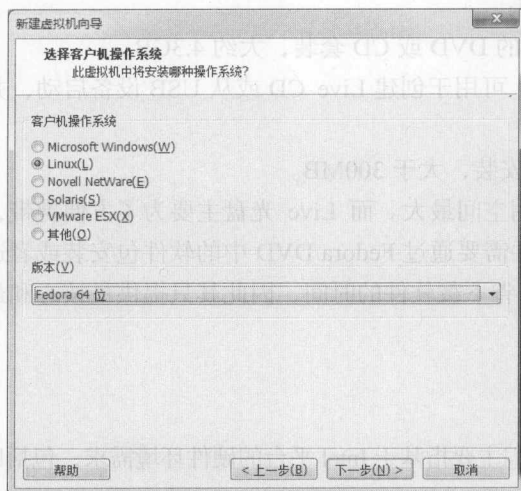


图 1.2 新建虚拟机向导

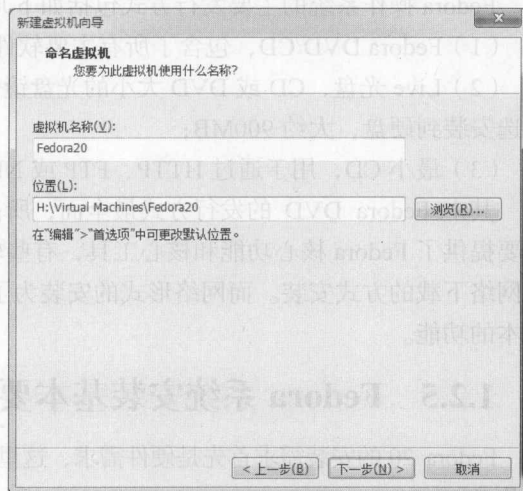


图 1.3 设置虚拟机的名称及文件保存位置

(3) 设置虚拟机的网络配置。每种网络类型对应客户机操作系统使用不同的方式与网络连接。我们这里使用第一种方式：桥接网络。桥接网络允许虚拟机中的操作系统访问主机中的网卡直接访问主机所在的网络，这是最简单的虚拟机中的操作系统访问外部网络的方法。设置虚拟机网络的配置如图 1.4 所示。

(4) 设置虚拟机操作系统所占用的磁盘容量大小, 一般而言, 越大越好。可根据具体情况将操作系统放在一个文件中或拆分为多个文件。设置方式如图 1.5 所示。

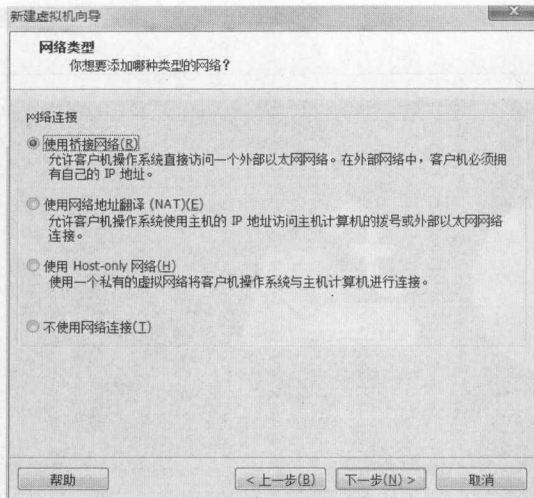


图 1.4 虚拟机网络设置

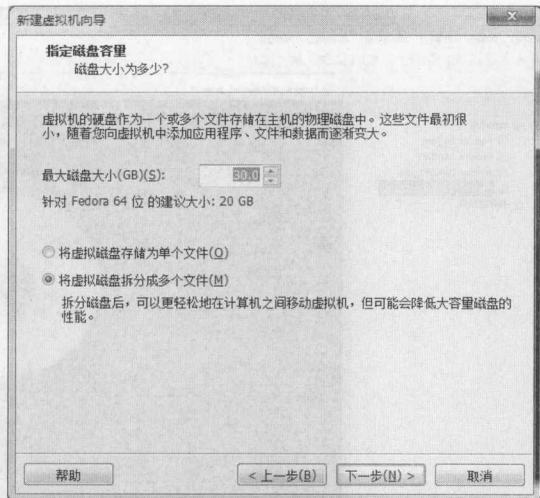


图 1.5 设定虚拟机操作系统的磁盘容量

(5) 最后以显示虚拟机操作系统的配置信息作为虚拟机操作系统参数配置向导的结束, 如图 1.6 所示。

(6) 虚拟机光驱加载安装文件。只有执行了这一步, 启动虚拟机才能开始安装操作系统。在新建虚拟机的名称上单击鼠标右键, 在弹出的快捷菜单中选择“属性”选项, 然后双击光驱, 弹出如图 1.7 所示的界面, 在该界面中右边的“使用 ISO 镜像”文件选项中可设置虚拟机安装文件的位置。

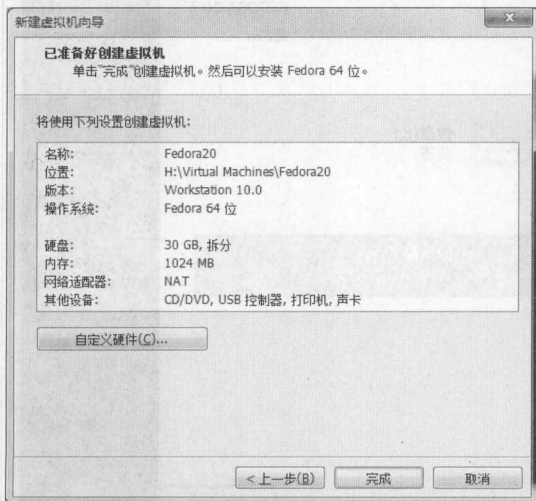


图 1.6 虚拟机配置信息



图 1.7 虚拟机光驱加载安装文件

1.3.2 Fedora 20 安装

在完成虚拟机的初始配置之后, 单击“启动”按钮, 此时, 虚拟机会自动读取虚拟机光驱上

所指定的 Fedora 20 安装文件并进行安装。主要安装过程如下所示。

(1) 安装启动画面如图 1.8 所示，这里选择安装到本地硬盘。

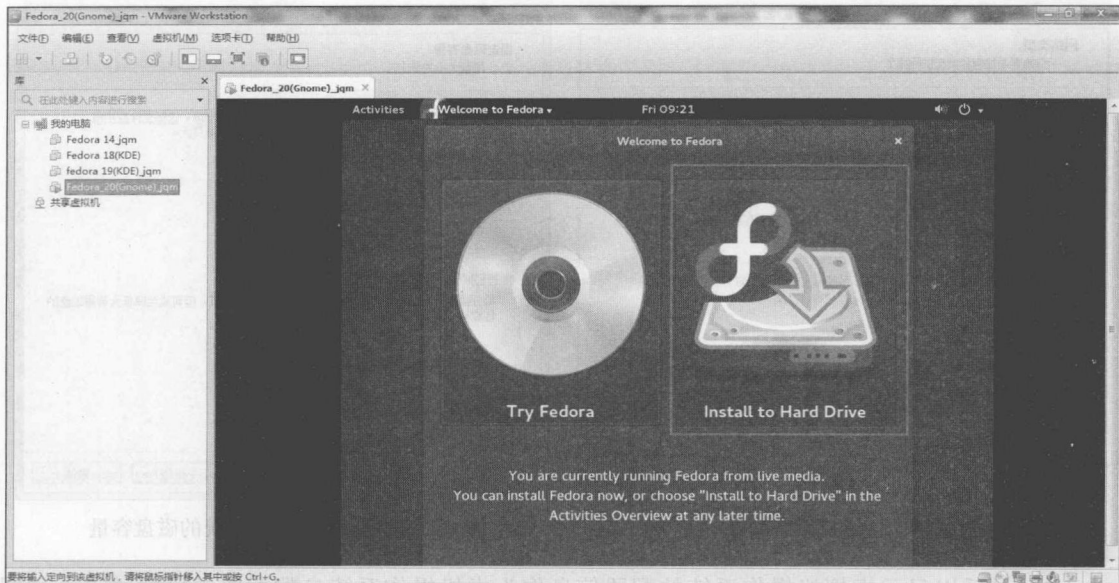


图 1.8 Fedora 安装界面

(2) 进行一系列主机基本参数的配置，如日期、时间、网络等情况，一般根据需要进行配置。如果不需要配置，则单击右下角的“开始安装”按钮即可，如图 1.9 所示。

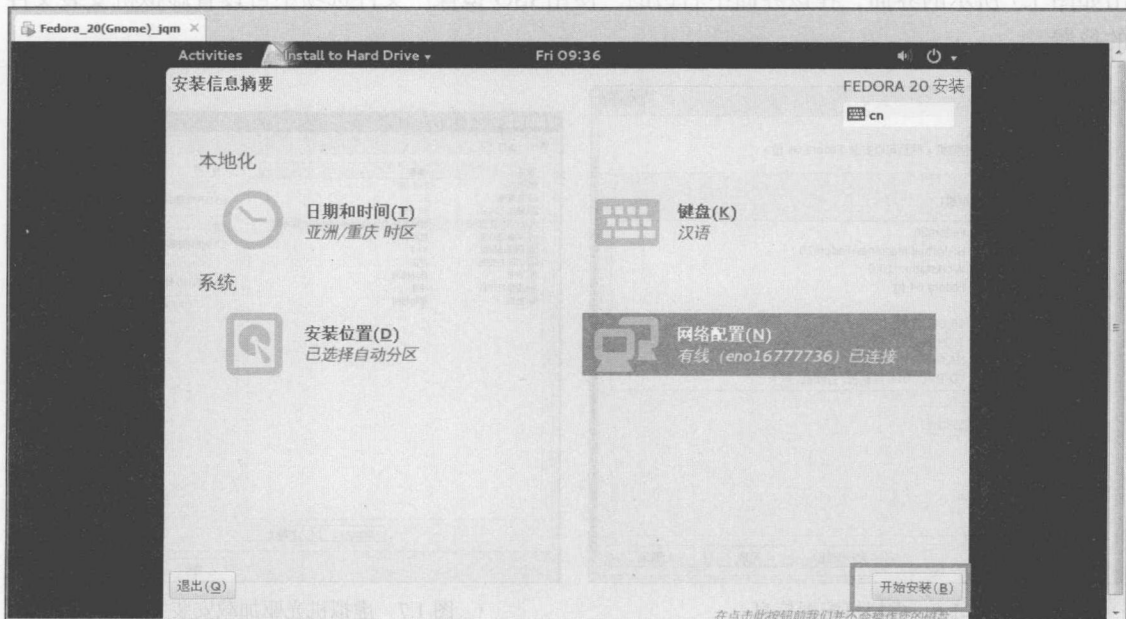


图 1.9 Fedora 基本参数配置

(3) 在安装过程中，需要配置根用户 root 的密码，如图 1.10 所示。这里一般可创建其他用户，不建议直接采用 root 用户账号登录，防止误操作导致系统崩溃。

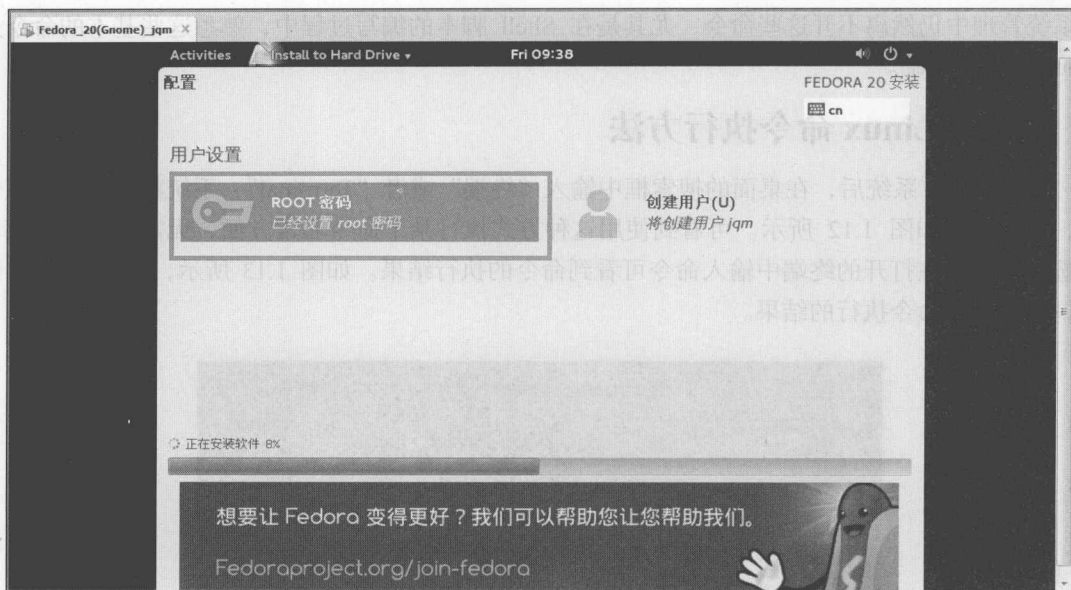


图 1.10 配置 root 用户密码

(4) 等待系统安装完成，完成后重启 Fedora 之后登录系统的界面如图 1.11 所示。首先单击左上角的“活动”，其次单击左下角方框中的地方，出现右边的界面，其中，右边方框中就是常用工具。

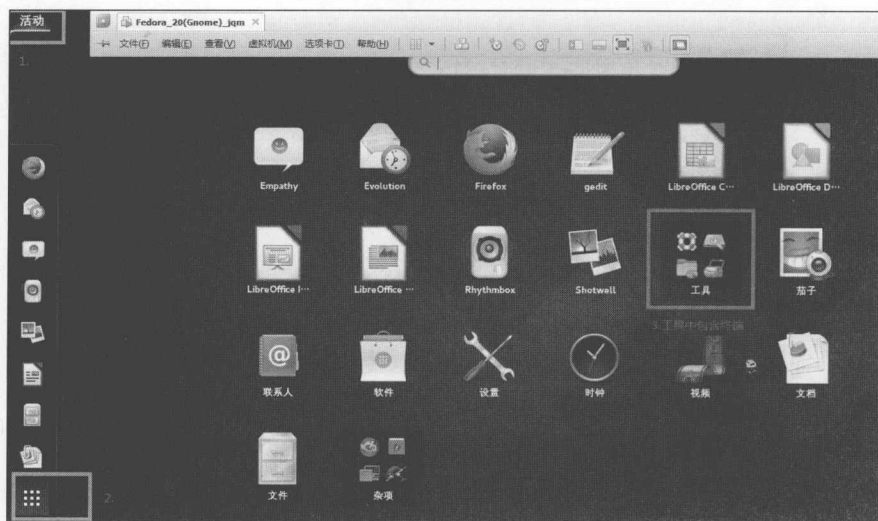


图 1.11 系统登录后的界面

1.4 Linux 常用命令

在使用 Linux 的过程中，我们需要使用大量的 Linux 命令，因为 UNIX 最初没有图形界面，很多任务需要用户输入执行命令来完成，这些命令具有强大的功能，有些功能没有图形界面。因

此系统管理中仍然离不开这些命令，尤其是在 Shell 脚本的编写过程中，熟悉这些基本的命令是 Linux 操作的基础。

1.4.1 Linux 命令执行方法

登录 Linux 系统后，在桌面的搜索框中输入“终端”或者“Terminal”，系统将自动显示该命令，执行过程如图 1.12 所示。可看到使用这种方式执行某个命令非常方便，无需通过多重菜单去选择命令。在打开的终端中输入命令可看到命令的执行结果。如图 1.13 所示，执行 `uname` 命令，可查看到命令执行的结果。

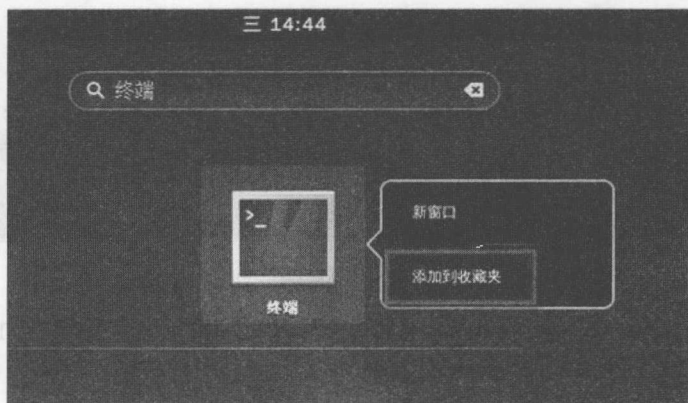


图 1.12 Linux 终端命令的执行

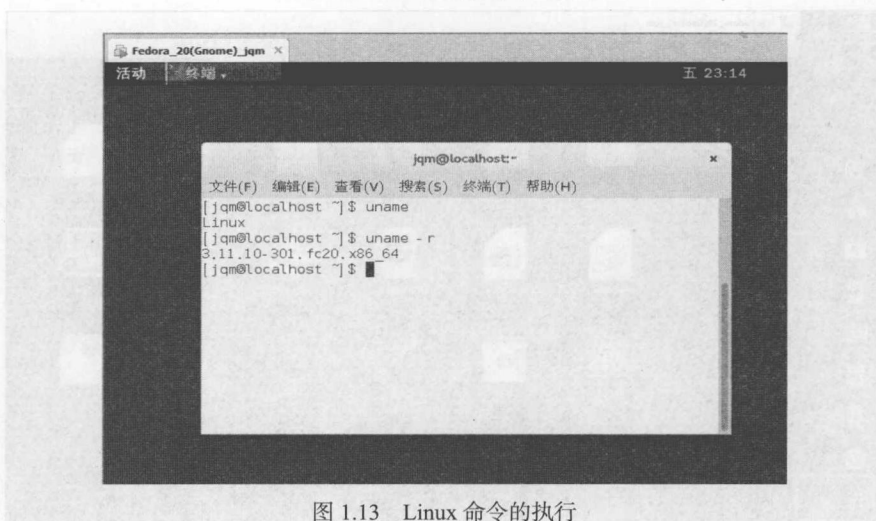


图 1.13 Linux 命令的执行

1.4.2 Linux 常用各种命令

1. man 及 info 帮助命令

使用 Linux 操作系统时，可以使用 `man` 命令获得帮助。首先我们获得关于 `man` 如何使用的帮助。在终端中输入命令：

```
[cosmos@localhost ~]$ man man
```

然后下面是关于 `man` 命令的帮助：

```

man (1)                                man (1)

NAME
    man - format and display the on-line manual pages

SYNOPSIS
    man [-acdfFhkKtWw] [--path] [-m system] [-p string] [-C config_file]
    [-M pathlist] [-P pager] [-B browser] [-H htmlpager] [-S section_list]
    [section] name ...

DESCRIPTION
    man formats and displays the on-line manual pages. If you specify section, man only looks in that section of the manual. name is normally the name of the manual page, which is typically the name of a command, function, or file. However, if name contains a slash (/) then man interprets it as a file specification, so that you can do man ./foo.5 or even man /cd/foo/bar.1.gz.

    See below for a description of where man looks for the manual page files.

MANUAL SECTIONS
    The standard sections of the manual include:

```

关于 man 命令，我们常用的格式如下：

```
man [ 参数选择 ] [节号] 命令
```

这里的节号指从哪一节查找该命令的相关信息。节号及其含义主要见表 1.1。

表 1.1 man 文档的章节

节 号	含 义
1	用户命令
2	系统调用及内核服务
3	库函数
4	特殊文件、设备驱动及硬件
5	文件格式及其惯例
6	游戏等
7	其他杂集
8	系统管理及维护工具

其中，参数-a 表示从所有的 man 文档章节中查找，然后从找到的第一个开始显示。-k 参数表示从所有的 man 文档中按照所指定的关键字进行查找。下面介绍几种 man 命令使用实例。

例 1.1 (1) man write 命令。在终端中输入如下命令：

```
[cosmos@localhost ~]$ man write
```

得到如下结果：

```

WRITE(1)                                Linux Programmer's Manual                                WRITE(1)

NAME
    write - send a message to another user

SYNOPSIS

```



```
write user [ttyname]
```

DESCRIPTION

Write allows you to communicate with other users, by copying lines from your terminal to theirs.

When you run the write command, the user you are writing to gets a message of the form:

```
Message from yourname@yourhost on yourtty at hh:mm ...
```

Any further lines you enter will be copied to the specified user's terminal. If the other user wants to reply, they must run write as well.

When you are done, type an end-of-file or interrupt character. The other user will see the message EOF indicating that the conversation is over.

从中可以看出该命令从第一节找到关于 write 用户命令的帮助信息。

(2) man 2 write 命令。在终端下输入 man 2 write 命令：

```
[cosmos@localhost ~]$ man 2 write
```

得到如下结果：

```
WRITE(2)      Linux Programmer's Manual      WRITE(2)
NAME
  write - write to a file descriptor
SYNOPSIS
  #include <unistd.h>

  ssize_t write(int fd, const void *buf, size_t count);
DESCRIPTION
  write() writes up to count bytes from the buffer pointed buf to the
  file referred to by the file descriptor fd.

  The number of bytes written may be less than count if, for example,
  there is insufficient space on the underlying physical medium, or the
  RLIMIT_FSIZE resource limit is encountered (see setrlimit(2)), or the
  call was interrupted by a signal handler after having written less than
  count bytes. (See also pipe(7).)

  For a seekable file (i.e., one to which lseek(2) may be applied, for
  example, a regular file) writing takes place at the current file off-
  set, and the file offset is incremented by the number of bytes actually.
```

该命令返回了系统调用 write 的帮助内容。第一行结果中 write(2)括号中的 2 表示从 man 文档的第二节关于系统调用的文档中获得的信息。

(3) man -a write 命令，执行该命令后，man 返回文档中所有的关于该命令 write 的信息，从第一节到所有的包含该命令的节号。当第一个显示完后，输入 q 命令，系统自动显示下一个该命令的信息。

(4) man -k write 命令。在终端下输入 man -k write 命令：

```
[cosmos@localhost ~]$ man -k write
```