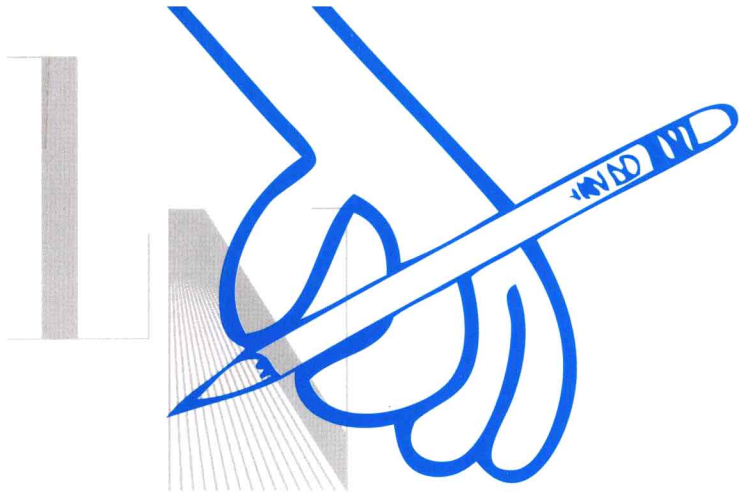




从零开始，手把手教会你在Linux中用C语言开发程序

本书特色：

熟悉系统→编程知识→实用案例→深入核心→项目实践

179个典型实例、80个练习题、1个项目开发案例

超值、大容量DVD

12小时多媒体视频教学

本书源代码、本书教学PPT

本书技术支持

论坛：<http://www.rzchina.net>

• 从零开始学编程 •

从零
开始学

Linux编程



12小时多媒体教学视频

◎ 罗利民 王浩 编著

循序渐进·由浅入深 内容充实·层次清楚 实例丰富·步骤清晰 对比讲解·理解深刻 习题指导·巩固学习 案例精讲·深入剖析



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

•从零开始学编程•

从零 开始学

Linux编程



编著

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

Linux 系统可以免费应用到多种服务器上,所以开发人员都在往 Linux 系统上转移,且很多开发都是使用 C 语言来完成的,这也是本书推出的关键所在。

本书共分为 5 篇:第 1 篇是 Linux 系统知识,讲述 Linux 系统的安装和使用、常用命令、文本编辑器及程序的编译与调试;第 2 篇是 C 语言基本知识,讲述 C 语言的基本语法、指针、结构体、系统库函数等知识;第 3 篇是 Linux 系统编程,讲述 Shell 编程和文件处理方法;第 4 篇是 Linux 网络编程,讲解 TCP 和 UDP 网络编程的知识;第 5 篇是 Linux C 项目实战,通过案例的学习,既复习了前面介绍的知识,也了解了项目实践的流程和技巧,从而增加读者的编程经验。

全书配合大量的图例及代码说明,内容精练,重点突出,实例丰富。如果你一点都不了解 Linux 和 C 语言编程,那么本书将是你不错的选择。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

从零开始学 Linux 编程 / 罗利民, 王浩编著. —北京: 电子工业出版社, 2012.6

(从零开始学编程)

ISBN 978-7-121-17026-3

I. ①从… II. ①罗… ②王… III. ①Linux 操作系统—程序设计 IV. ①TP316.89

中国版本图书馆 CIP 数据核字 (2012) 第 096185 号

策划编辑: 胡辛征

责任编辑: 葛 娜

印 刷: 北京东光印刷厂

装 订: 三河市皇庄路通装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 21.25 字数: 542 千字

印 次: 2012 年 6 月第 1 次印刷

印 数: 4000 册 定价: 49.80 元 (含 DVD 光盘 1 张)

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。



微软并不是魔鬼，只是它的操作系统实在太蹩脚了。

——Linux 之父 Linus Torvalds

Linux 系统是一个非常优秀的系统，在服务器解决方案方面有着巨大的优势。近年来国内无数大型企业使用 Linux 系统作为服务器解决方案，对服务器软件开发人员的需求大量增加。而这些工作通常是使用 C 语言来完成的。虽然 C 程序有着很好的先进性，有着非常重要的用途，但是相对于 C#、Java 等面向对象语言，C 程序的程序设计方法、符号的使用、面向过程的思路 and 严格的语法等这些特点使 C 语言并不容易学习和掌握。

在 Baidu 上搜索“Linux 编程”的网页有 9 250 000 个，关于“Linux 开发”的网页有 5 370 000 个，关于“Visual C++”的网页只有 6 170 000 个。由此可见，Linux 编程关注人数之多，企业需求之大。正是因为有如此大的需求，所以笔者结合多年的开发经验，编写了本书。

本书特点

1. 注重读者的学习与接受情况

在 Linux 系统下学习编程时，一个很大的困难就是平台的搭建与使用。本书从 Linux 系统安装开始，引导读者一步步进入 Linux 系统，掌握与 Linux 系统编程相关的一般操作。

2. 代码注释细致，方便学习

对实例代码进行了非常详细的注释，这些注释可以引导读者理解和掌握抽象的知识。在实例中几乎对代码的每一行进行讲解，这些讲解可以促进初级读者的学习和理解。

3. 用 VI 编辑器作为开发平台

在 Linux 下开发没有统一的开发平台，这也是很多开发者郁闷的地方。笔者经过多年的比较，并和同行讨论，认为 VI 编辑器是其中的佼佼者，因此把它详细地介绍给大家。

4. 内容集中，有的放矢

对于 Windows 平台下的 C 语言开发，市场上已经有很多同类书籍了，因此对这部分内容，笔者使用极少的篇幅帮助读者入门，然后快速过渡到 Linux 开发。

5. 内容深入

在注重基础和实际的同时，注重知识的扩充。网络编程和图形界面编程是 C 程序学习中比较深入的知识，本书对这些知识的讲述可以使读者对 C 程序学习有一个系统的提高。

本书结构

本书共分为 5 篇 18 章，主要章节规划如下所示。

第 1 篇（第 1~4 章） Linux 系统知识

讲述 Linux 系统的安装和使用、常用命令、文本编辑器的使用及程序的编译与调试。

第 2 篇（第 5~12 章） C 语言基本知识

讲述 C 语言的基本语法、常用语句，以及各种类型的数据、数组和指针及结构体等知识。同时对系统库函数的使用进行了介绍。在本篇的最后，介绍了常见的编程规范。

第 3 篇（第 13~14 章） Linux 系统编程

讲述 Linux 系统中的 Shell 编程和文件（目录）处理方法。

第 4 篇（第 15~17 章） Linux 网络编程

介绍在 Linux 上如何进行网络编程，同时对基于 TCP 和 UDP 的网络编程进行实例讲解。

第 5 篇（第 18 章） Linux C 项目实战

通过案例的学习，既复习了前面介绍的知识，也了解了项目实践的流程和技巧，从而增加读者的编程经验。

本书由浅入深，由理论到实践，尤其适合初级读者逐步学习和完善自己的知识结构。

适合阅读本书的读者

- ☐ C 程序初学者
- ☐ 高校计算机专业的学生
- ☐ 高校做毕业设计的学生
- ☐ Linux 爱好者和工作人员
- ☐ Linux 相关学习和研究的研究生
- ☐ Linux 程序员
- ☐ 嵌入式系统开发入门人员

本书作者与技术支持

本书第 1~10 章由罗利民编写，第 11~18 章由王浩编写。在编写的过程中得到了出版社各位编辑的协助，在此表示感谢！

读者在阅读本书的过程中如果有疑问，请到本书技术支持论坛提问，会有专人答复。
论坛网址：www.rzchina.net。

编著者

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010)88254396；(010)88258888

传 真：(010)88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱 电子工业出版社总编办公室

邮 编：100036

关于本书用纸说明

亲爱的读者朋友：您所拿到的这本书使用的是**环保轻型纸**！

环保轻型纸在制造过程中添加化学漂白剂较少，颜色更接近于自然状态，具有纸质轻柔、光反射率低、保护读者视力等优点，其成本略高于胶版纸。为给您带来更好的阅读体验并与读者共同支持环保，我们在没有提高图书定价的前提下，使用这种纸张。愿我们共同分享纸质图书的阅读乐趣！

电子工业出版社博文视点

目 录

第 1 篇 Linux 系统知识

第 1 章 认识 Linux 系统	1
1.1 Linux 发行版的介绍	1
1.1.1 Red Hat Linux	1
1.1.2 Debian Linux	1
1.2 Linux 的安装	2
1.2.1 Linux 的分区格式	2
1.2.2 在 Windows 系统中划分 Linux 分区	2
1.2.3 Red Hat Linux 9.0 的安装	4
1.2.4 GRUB 在 Windows 分区中引导 Linux	13
1.3 VMware 虚拟机安装 Linux 系统	15
1.3.1 VMware 软件安装	15
1.3.2 新建虚拟计算机	17
1.3.3 VMware 的使用	21
1.3.4 在 VMware 虚拟机中安装 Linux 系统	23
1.4 第一个 C 程序	25
1.4.1 在 Linux 中编写 C 程序代码	25
1.4.2 编译程序	26
1.5 小结	27
1.6 小测试	27
第 2 章 常用的 Linux 操作命令	28
2.1 目录与文件命令	28
2.1.1 查看目录命令 pwd	28

2.1.2 打开目录命令 cd	28
2.1.3 查看目录命令 ls	29
2.1.4 建立文件命令 touch	30
2.1.5 建立与删除文件夹命令 mkdir、rmdir	30
2.1.6 文件复制命令 cp	31
2.1.7 文件移动和更名命令 mv	32
2.1.8 删除文件与目录命令 rm	32
2.1.9 创建链接命令 ln	33
2.1.10 查看文件信息命令 wc	33
2.1.11 文件查看命令 cat	34
2.1.12 查看文件视图命令 head 与 tail	34
2.1.13 文件权限设置命令 chmod	35
2.1.14 查找文件命令 find	36
2.2 man 系统帮助命令	36
2.2.1 man 手册的使用方法	37
2.2.2 帮助文档的结构	37
2.3 info 手册	38
2.4 小结	39
2.5 小测试	39
第 3 章 文本编辑器的使用	40
3.1 VIM 的使用	40
3.1.1 VIM 的启动	40
3.1.2 在桌面上创建 VIM 启动器	40
3.1.3 VIM 的工作模式	41
3.1.4 保存与打开文件	42
3.1.5 移动光标	43

3.1.6	常用操作	43	4.5.1	gdb 简介	65
3.1.7	选项设置	45	4.5.2	在程序中加入调试信息	65
3.1.8	调用 shell 命令	46	4.5.3	启动 gdb	65
3.2	VIM 使用实例	46	4.5.4	在 gdb 中加载需要调试 的程序	66
3.2.1	字符的插入与删除	46	4.5.5	在 gdb 中查看代码	66
3.2.2	字符的查找与替换	47	4.5.6	在程序中加入断点	66
3.3	带菜单的 GVIM	48	4.5.7	查看断点	67
3.3.1	文件的新建与保存	48	4.5.8	运行程序	67
3.3.2	查找与替换	49	4.5.9	查看变量	68
3.4	图形化的 gedit	50	4.6	程序调试实例	69
3.4.1	gedit 的启动与打开文件	50	4.6.1	编写一个程序	69
3.4.2	gedit 编辑文件	50	4.6.2	编译文件	70
3.4.3	gedit 打印文件	51	4.6.3	程序调试	70
3.4.4	gedit 的首选项	52	4.6.4	gdb 帮助的使用	72
3.5	小结	54	4.7	gdb 常用命令	72
3.6	小测试	55	4.8	编译程序常见的错误与问题	74
第 4 章	编译与调试	56	4.8.1	逻辑错误与语法错误	74
4.1	编译的概念和理解	56	4.8.2	C 程序中的错误与异常	74
4.1.1	程序编译的过程	56	4.8.3	编译中的警告提示	74
4.1.2	编译器	57	4.8.4	找不到包含文件的错误	75
4.2	gcc 编译器	57	4.8.5	错误地使用逗号	75
4.2.1	gcc 编译器简介	57	4.8.6	括号不匹配错误	75
4.2.2	gcc 对源程序扩展名的支持	57	4.8.7	小括号不匹配错误	76
4.3	C 程序的编译	58	4.8.8	变量类型或结构体声明错误	76
4.3.1	编写第一个 C 程序	58	4.8.9	使用不存在的函数错误	76
4.3.2	用 gcc 编译程序	59	4.8.10	大小写错误	76
4.3.3	查看 gcc 的参数	59	4.8.11	数据类型错误	77
4.3.4	设置输出的文件	60	4.8.12	赋值类型错误	77
4.3.5	查看编译过程	61	4.8.13	循环或判断语句中多加分号	77
4.3.6	设置编译的语言	61	4.9	小结	78
4.4	编译过程的控制	62	4.10	小测试	78
4.4.1	编译过程简介	62			
4.4.2	控制预处理过程	63			
4.4.3	生成汇编代码	63			
4.4.4	生成目标代码	64			
4.4.5	链接生成可执行文件	64			
4.5	gdb 调试程序	65			

第 2 篇 C 语言基本知识

第 5 章	C 语言基础	79
5.1	C 程序的基本概念	79
5.1.1	C 程序的基本结构	79

5.1.2	C 程序的一般格式	80	6.3.2	块语句	112
5.1.3	C 程序中的注释	81	6.4	小结	113
5.2	数据类型	82	6.5	小测试	113
5.2.1	整型 (int)	82	第 7 章 数组与指针		114
5.2.2	浮点型 (float)	83	7.1	数组的理解与操作	114
5.2.3	字符型 (char)	84	7.1.1	什么是数组	114
5.2.4	变量名	85	7.1.2	数组的定义与访问	115
5.2.5	字符 NULL	86	7.1.3	数组使用实例	116
5.3	变量赋值与输出	86	7.2	指针	117
5.3.1	变量赋值	86	7.2.1	对指针的理解	117
5.3.2	printf 函数输出变量	86	7.2.2	指针操作符	117
5.3.3	scanf 函数从键盘读入变量	88	7.2.3	指针的定义与访问	118
5.4	运算符	89	7.2.4	指针使用实例	119
5.4.1	算术运算符	89	7.3	数组与指针	120
5.4.2	关系运算符	91	7.3.1	数组与指针的关系	120
5.4.3	逻辑运算符	91	7.3.2	指针的算术运算	120
5.5	小结	92	7.3.3	字符数组与字符串	121
5.6	小测试	93	7.4	二维数组与多维数组	122
第 6 章 C 程序的常用语句		94	7.4.1	对二维数组的理解	122
6.1	流程控制语句	94	7.4.2	二维数组与指针	123
6.1.1	if 条件语句	94	7.5	实例	125
6.1.2	if 语句嵌套	96	7.5.1	学生成绩统计实例	125
6.1.3	switch 选择执行语句	98	7.5.2	冒泡法排序实例	126
6.1.4	for 循环语句	99	7.5.3	统计字符串中的字符	127
6.1.5	for 循环嵌套	101	7.5.4	小写字母转换成大写字母	129
6.1.6	for 循环应用实例: 输出九九乘法口诀表	102	7.5.5	用指针访问数组	130
6.1.7	while 循环语句	103	7.6	常见问题	131
6.1.8	do while 循环语句	104	7.6.1	错误的写操作	131
6.1.9	转移控制语句: continue	105	7.6.2	指针的错误赋值	132
6.1.10	转移控制语句: break	106	7.6.3	数组指针的越界错误	133
6.1.11	转移控制语句: return	107	7.7	小结	134
6.2	流程控制语句实例	107	7.8	小测试	134
6.2.1	三个数字的排序	108	第 8 章 函数		135
6.2.2	解一元二次方程	109	8.1	对函数的理解	135
6.3	两种特殊语句结构	111	8.1.1	什么是函数	135
6.3.1	三元操作符	111	8.1.2	系统函数 (库函数) 与 用户自定义函数	136

8.1.3	函数的返回值	137	9.1.4	大小写字母测试函数 islower()和 isupper()	160
8.1.4	无参函数	138	9.1.5	数字测试函数 isdigit()	161
8.1.5	有参函数	138	9.1.6	符号测试函数 ispunct()	162
8.1.6	函数参数实例	138	9.1.7	其他字符测试函数	163
8.2	自定义函数	140	9.2	字符测试函数综合实例	163
8.2.1	自定义函数的编写	140	9.2.1	统计字符串中各类 字符的个数	163
8.2.2	函数中调用函数	141	9.2.2	判断电话号码与姓名 是否正确	165
8.2.3	函数声明	143	9.3	字符串转换	167
8.2.4	递归函数	144	9.3.1	C 程序中的字符串	167
8.2.5	main 函数的参数 argc 与 argv	145	9.3.2	字符串转换成浮点型函数 atof()	167
8.3	函数与指针	145	9.3.3	字符串转换成整型 函数 atoi()	168
8.3.1	值调用与引用调用	146	9.3.4	字符串转换成成长整型 函数 atol()	169
8.3.2	引用调用与指针	146	9.3.5	将浮点型数转换成字符串 函数 ecvt()	170
8.3.3	指针参数实例	147	9.3.6	字母的大小写转换函数 tolower()和 toupper()	170
8.3.4	自定义函数中指针 使用实例	148	9.3.7	其他字符串转换函数	171
8.3.5	数组作为参数	149	9.4	字符串比较	172
8.3.6	数组作为函数参数实例	150	9.4.1	字符串比较函数 bcmp()	172
8.4	返回值	151	9.4.2	字符串大小比较函数 memcmp()	173
8.4.1	函数返回值的类型	151	9.4.3	忽略大小写比较字符串 函数 strncasecmp()	174
8.4.2	函数返回指针	152	9.5	字符串复制	175
8.4.3	函数返回指针实例	153	9.5.1	字符串复制函数 bcopy()	175
8.4.4	函数返回结果到参数	154	9.5.2	字符串复制函数 memcpy()	175
8.5	库函数	154	9.5.3	字符串复制函数 strcpy()	176
8.5.1	库函数的种类	154	9.5.4	字符串复制函数 strncpy()	177
8.5.2	库函数包含文件	155	9.6	字符串的清理与填充	177
8.5.3	头文件使用实例	156	9.6.1	字符串清理函数 bzero()	177
8.6	小结	156			
8.7	小测试	156			
第 9 章	字符与字符串处理	158			
9.1	字符测试函数介绍	158			
9.1.1	数字或字母测试函数 isalnum()	158			
9.1.2	字母测试函数 isalpha()	159			
9.1.3	可打印字符测试函数 isgraph()	160			

9.6.2 字符串填充函数 <code>memset()</code>	178	第 11 章 时间函数	200
9.7 字符串查找	179	11.1 常用的时间函数	200
9.7.1 字符串查找函数 <code>index()</code> 与 <code>rindex()</code>	179	11.1.1 返回时间函数	200
9.7.2 字符串查找函数 <code>memchr()</code> ..	180	11.1.2 取当前时间函数	201
9.7.3 字符串查找函数 <code>strchr()</code> 与 <code>strrchr()</code>	180	11.1.3 字符串格式时间函数	203
9.8 字符串的连接与分割	181	11.1.4 另一种字符串格式 时间函数	203
9.8.1 字符串连接函数 <code>strcat()</code>	181	11.1.5 取得当地时间函数	204
9.8.2 字符串分割函数 <code>strtok()</code>	182	11.1.6 将时间转换成秒数函数	205
9.9 其他字符串函数	182	11.1.7 取得当前时间函数	206
9.9.1 字符串长度函数 <code>strlen()</code>	183	11.1.8 设置当前时间函数	207
9.9.2 允许出现字符查找函数 <code>strspn()</code>	183	11.2 时间函数使用实例	208
9.9.3 不允许出现字符查找 函数 <code>strcspn()</code>	184	11.2.1 运行程序所需要的时间	208
9.10 小结	185	11.2.2 两次输入之间的时间间隔	209
9.11 小测试	185	11.2.3 设置系统时间	210
第 10 章 结构体	186	11.3 小结	211
10.1 结构体操作	186	11.4 小测试	211
10.1.1 对结构体的理解	186	第 12 章 C 语言的编程规范	212
10.1.2 结构体的定义	186	12.1 命名规范	212
10.1.3 结构体的访问	188	12.2 格式规范	214
10.1.4 结构体数组	189	12.3 函数规范	215
10.1.5 结构体使用实例	190	12.4 其他规范	216
10.2 结构体与指针	191	第 3 篇 Linux 系统编程	
10.2.1 结构体指针的定义	192	第 13 章 Shell 编程	218
10.2.2 结构体指针的访问	192	13.1 bash 的使用	218
10.2.3 结构体作为函数的参数	193	13.1.1 bash 的内部命令与 外部命令	218
10.2.4 结构体指针作为 函数的参数	195	13.1.2 命令补齐功能	218
10.2.5 结构体作为函数的 返回值	196	13.1.3 命令通配符	219
10.2.6 结构体指针作为函数的 返回值	197	13.1.4 使用命令的历史记录	219
10.3 小结	199	13.1.5 命令别名	220
10.4 小测试	199	13.2 Shell 程序的基本结构	220
		13.3 局部变量	221
		13.4 环境变量	221
		13.4.1 查看环境变量	222
		13.4.2 访问环境变量	222

15.1.2	端口	270	16.1.1	建立 socket	287
15.1.3	域名	270	16.1.2	取得 socket 状态	289
15.1.4	TCP 与 UDP	270	16.1.3	设置 socket 状态	290
15.2	套接字	271	16.2	面向无连接的套接字通信	291
15.2.1	什么是套接字	271	16.2.1	工作流程	292
15.2.2	套接字相关的数据类型	271	16.2.2	信息发送函数 sendto()	292
15.2.3	套接字类型	272	16.2.3	信息接收函数 recvfrom()	293
15.3	域名与 IP 地址	272	16.3	无连接的套接字通信实例	294
15.3.1	用域名取得主机的 IP 地址	272	16.3.1	无连接的套接字通信 客户端	294
15.3.2	用 IP 地址返回域名	274	16.3.2	无连接的套接字通信 服务器	295
15.4	网络协议	275	16.3.3	测试 UDP 通信程序	297
15.4.1	由协议名称取得 协议数据	275	16.4	小结	298
15.4.2	由协议编号取得 协议信息	276	16.5	小测试	298
15.4.3	取得系统支持的 所有协议	277	第 17 章	面向连接的套接字通信	299
15.5	网络服务	278	17.1	面向连接的套接字通信 工作流程	299
15.5.1	取得系统支持的 网络服务	278	17.2	绑定端口	299
15.5.2	用名称取得系统支持 的服务	279	17.2.1	绑定端口函数 bind()	299
15.5.3	由端口取得服务名称	281	17.2.2	使用 bind() 函数绑定 端口实例	301
15.6	网络 IP 地址的转换	282	17.3	监听与连接	302
15.6.1	将网络地址转换成 长整型数	282	17.3.1	等待监听函数 listen()	302
15.6.2	将长整型 IP 地址转换成 网络地址	283	17.3.2	listen() 函数使用实例	303
15.6.3	主机字符顺序与网络字符 顺序的转换	284	17.3.3	接受连接函数 accept()	304
15.7	错误处理	285	17.3.4	accept() 函数使用实例	304
15.7.1	herror() 函数显示错误	285	17.3.5	请求连接函数 connect()	306
15.7.2	捕获错误编号	285	17.3.6	connect() 函数使用实例	307
15.8	小结	286	17.4	数据的发送与接收	308
15.9	小测试	286	17.4.1	数据接收函数 recv()	308
第 16 章	无连接的套接字通信	287	17.4.2	recv() 函数使用实例	309
16.1	socket 套接字	287	17.4.3	数据发送函数 send()	310
			17.4.4	数据传输函数 write() 与 read()	311
			17.4.5	read() 函数接收数据实例	311

17.5	面向连接的套接字通信实例	313
17.5.1	服务器程序	313
17.5.2	客户端程序	316
17.5.3	实例程序测试	317
17.6	小结	318
17.7	小测试	318

第 5 篇 Linux C 项目实战

第 18 章	电话本项目的开发	319
18.1	程序功能分析	319
18.2	程序的函数	319
18.3	包含文件	320

18.4	数据的定义	320
18.5	菜单函数	321
18.6	显示电话信息函数	322
18.7	添加电话号码函数	322
18.8	按姓名查找函数	322
18.9	删除电话号码函数	323
18.10	保存到文件函数	324
18.11	从文件导入信息函数	324
18.12	主函数	325
18.13	程序的运行	326
18.14	小结	327

第 1 篇 Linux 系统知识

第 1 章 认识 Linux 系统

Windows 操作系统相信很多读者都不陌生，但谈到 Linux 系统，大家可能就不太熟悉了。为此，在本章中，笔者将对 Linux 系统的相关知识进行简单的说明和介绍。

本章主要涉及的内容有：

- ❑ Linux 发行版，介绍两种常见的 Linux 发行版本。
- ❑ Linux 的安装，介绍安装 Linux 系统需要注意的事项，以及如何安装 Red Hat 9.0。
- ❑ 虚拟机安装，介绍如何安装虚拟机软件及其使用，同时在虚拟机中安装 Linux 系统。
- ❑ C 程序，介绍如何在 Linux 中编写 C 程序代码，同时进行简单的编译。



1.1 Linux 发行版的介绍

所谓 Linux 发行版，指的是以 Linux 内核为中心，集成各种系统管理软件或应用软件，组成一套完整的操作系统。这种在内核上开发出的可供用户使用的 Linux 系统就是 Linux 发行版。Linux 内核指的是可提供硬件抽象层、磁盘及文件系统控制、多任务等功能的系统软件。内核不是一套完整的操作系统，在计算机上还无法使用，需要发行版中的软件支持。本节将介绍两种常见的发行版本。

1.1.1 Red Hat Linux

Red Hat 是最成熟的一种 Linux 发行版，在服务器和个人桌面、嵌入式开发方面有着广泛的应用。Red Hat 的第一个版本发布于 1994 年，到现在为止已经发布了数十个版本。本书使用 Red Hat Linux 9.0 进行开发和讲解。

在使用上，Red Hat Linux 有着非常友好的界面，Linux 系统不再是字符界面的操作，而是和 Windows 一样有着非常便捷的图形操作。在应用软件和开发工具等方面，Red Hat Linux 9.0 提供了 OpenOffice.org 办公软件、Mozilla 浏览器、网站和数据库服务器、QT 开发平台等开源工具。

Red Hat Linux 在安装系统时允许用户根据自己的不同需求自主选择安装环境。例如，可以安装家庭台式机、工作站、服务器以及自定义等不同的应用环境。在使用个人台式电脑学习和开发时，需要选择安装开发编程环境。

1.1.2 Debian Linux

Debian 是一套自由开发的 Linux 操作系统，在个人计算机和商业领域有着广泛的应用。Debian 最大的特点是集成了数万个软件包，用户可以在网络中非常方便地管理和升级所需要的软件。

Debian 的软件管理有着良好的集成度，所有的软件都是由官方编译成 deb 包，然后发布到升级服务器中的。deb 的软件管理方式很好地解决了复杂的软件依赖性问题，使 Debian Linux 更加稳定、强健。apt-get 这种软件管理方式，使得软件的下载和安装非常简单，只需要一个简单的命令就可以完成软件下载与安装的全部操作。

1.2 Linux 的安装

Linux 系统的安装与 Windows 系统的安装相似，用安装光盘可以方便地安装 Linux 系统。与 Windows 系统安装不同的是，Linux 系统的安装需要注意系统的引导与分区。特别是与 Windows 一起安装双系统时，更需要注意分区和引导的问题。本节将讲解 Linux 分区、Linux 安装等问题。

1.2.1 Linux 的分区格式

Linux 的分区格式与 Windows 是完全不相同的。Windows 使用的是 FAT32 或 NTFS 格式的分区系统，而 Linux 使用 Ext2 或 Ext3 格式的分区系统。除了文件分区以外，Linux 还需要一个 Swap 交换分区。

- ❑ Ext2: Ext2 是 Linux 系统中标准的文件系统，也是 Linux 中使用最多的一种文件系统。Ext2 是专门为 Linux 设计的分区格式，有极快的访问速度和极小的 CPU 占用率。Ext2 既可以应用于硬盘，也可以应用在软盘、移动硬盘、闪存等移动存储设备上。
- ❑ Ext3: Ext3 是一种日志式文件系统（Journal File System），是 Ext2 的下一代分区格式，在 Ext2 的格式上再加上日志功能。Ext3 的最大特点是将整个磁盘的写入动作以日志的形式完整地记录在磁盘的某个区域上，以便在需要时回溯追踪。当文件分区中的某个操作过程出现异常时，可以根据所记录的日志直接找回被中断的部分。这种日志式的分区格式使文件的存储非常稳定，被广泛应用在 Linux 系统中。
- ❑ Swap: Swap 是 Linux 系统专门用于内存交换的硬盘分区，相当于 Windows 系统中的虚拟内存。Swap 交换分区的大小一般是内存的两倍。在内存不够时，Linux 会将不活动的内存数据存储在交换分区中。

1.2.2 在 Windows 系统中划分 Linux 分区

在安装 Linux 系统时，一般需要将 Linux 与 Windows 系统安装在同一台计算机中，这就需要在已经安装好的 Windows 系统中为 Linux 划分分区。使用 Norton PartitionMagic 分区软件，可以方便地在 Windows 系统的图形界面上对硬盘进行分区。下面的步骤是使用 Norton PartitionMagic 在硬盘中划分 Linux 分区以供 Linux 安装。

① 从网络上下载“Norton PartitionMagic”分区软件并且安装。打开 Norton PartitionMagic 进行硬盘分区，Norton PartitionMagic 的界面如图 1.1 所示。在进行硬盘分区时，需要关闭其他正在运行的软件。

② 查看计算机中的分区。在 Norton PartitionMagic 的分区列表中，显示了计算机中的硬盘分区情况和分区使用情况，如图 1.2 所示。

③ 删除最后一个分区。在删除分区之前，需要备份最后一个分区中的文件。例如，图 1.2 显示的最后分区是“F:”盘，需要将“F:”盘中的数据转移到其他分区。

④ 在分区列表中，选择最后一个分区。在如图 1.1 所示的界面中，单击“分区操作”标签中的“删除分区”工具，弹出“删除分区”对话框，如图 1.3 所示。

⑤ 单击“确定”按钮删除分区。需要注意的是，对分区进行删除操作，会删除一个分区中的所有数据，删除以后的数据是不能再恢复的。所有的操作需要谨慎，不可以出现错误。

⑥ 如图 1.4 所示，删除一个分区以后，硬盘中有了 15GB 的未分配空间。



图 1.1 使用Norton PartitionMagic进行硬盘分区

分区	类型	容量 MB	已使用 MB	未使用 MB	状态	主/逻辑
磁盘 1						
本地磁盘 (C:) (★)	FAT32	14,998.2	4,253.2	10,744.9	活动	主
(★)	扩展	63,522.6	63,5...	0.0	无	主
本地磁盘 (D:) (★)	FAT32	20,002.8	2,789.4	17,213.4	无	逻辑
本地磁盘 (E:) (★)	FAT32	28,498.1	12,6...	15,834.5	无	逻辑
SOFT (F:) (★)	FAT32	15,021.7	10.6	4,373.0	无	逻辑
(★)	未分配	7.8	0.0	0.0	无	主

图 1.2 在Norton PartitionMagic中查看计算机的硬盘分区情况和分区使用情况

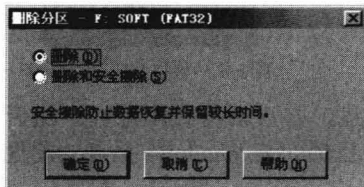


图 1.3 “删除分区”对话框

分区	类型	容量 MB	已使用 MB	未使用 MB	状态	主/逻辑
磁盘 1						
本地磁盘 (C:) (★)	FAT32	14,998.2	4,253.2	10,744.9	活动	主
(★)	扩展	63,522.6	48.5...	15,021.7	无	主
本地磁盘 (D:) (★)	FAT32	20,002.8	2,789.4	17,213.4	无	逻辑
本地磁盘 (E:) (★)	FAT32	28,498.1	12,6...	15,834.5	无	逻辑
(★)	未分配	15,021.7	0.0	0.0	无	逻辑
(★)	未分配	7.8	0.0	0.0	无	主

图 1.4 删除分区以后的硬盘分区列表

⑦ 新建一个 Swap 交换分区。在分区列表中，选择未分区的硬盘空间。在如图 1.1 所示的界面中，单击“分区操作”标签中的“创建分区”工具，弹出“创建分区”对话框，如图 1.5 所示。

⑧ 在“创建为”下拉菜单中，选择“逻辑分区”选项，在“分区类型”下拉菜单中，选择“Linux Swap”选项；在“大小”文本框中输入分区的大小 1024（Swap 交换分区的大小应该是内存大小的两倍）。然后单击“确定”按钮完成设置。

⑨ 创建一个引导分区。在 Linux 系统中，需要由引导程序来引导系统。引导程序常常存放在一个独立的分区中，大小一般是 100~200MB。按照步骤 7，新建一个分区，“创建分区”对话框如图 1.6 所示。