

Broadview®
www.broadview.com.cn

基础 ▶ 案例 ▶ 实践

一本全面讲解Linux下C程序设计的学习宝典!

完全手册

Linux系统下 C程序开发详解

徐磊 等编著



程序源代码 + 电子教案 + Linux实用电子书



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



完全手册



Linux系统下 C程序开发详解

徐磊 等编著

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING



内 容 简 介

本书由浅入深、通俗易懂地讲解了 Linux 系统下的 C 程序设计。全书共 24 章,从 Linux 系统安装开始,讲解了 Linux 常用命令、Shell 编程、C 语言基础、指针与函数、C 程序网络编程、图形界面程序设计等内容。书中通过大量的实例来讲解知识要点,读者可以通过这些实例和练习对 C 程序有一个系统的学习和提高。

本书共有 7 个方面的内容。Linux 基础讲述了 Linux 系统的安装与编辑器的使用。Shell 编程介绍了 Linux 常用命令和 Shell 程序的设计。C 程序基础从 C 语言的变量和语句开始,讲解了 C 程序设计的基础语法知识。指针与函数是 C 程序设计中的重点和难点,使用了大量的实例进行讲解。文件与目录讲述了 C 语言对文件和目录的访问。网络编程讲述 C 程序网络通信的实现。图形界面设计讲述了如何使用 GTK 库开发出 Linux 系统的图形界面程序。

本书语言通俗易懂,内容丰富,注重理解与实例,知识涵盖面广。非常适合 C 语言的初学者、高校学生、程序开发人员以及 Linux 工作人员阅读和学习。

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

Linux 系统下 C 程序开发详解 / 徐磊等编著. —北京: 电子工业出版社, 2008.6

(完全手册)

ISBN 978-7-121-06399-2

I. L… II. 徐… III. ①Linux 操作系统—程序设计 ②C 语言—程序设计 IV. TP316.89 TP312

中国版本图书馆 CIP 数据核字(2008)第 052844 号

责任编辑: 高洪霞

印 刷: 北京智力达印刷有限公司

装 订: 北京中新伟业印刷有限公司

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: 33.25 字数: 813 千字

印 次: 2008 年 6 月第 1 次印刷

印 数: 5000 册 定价: 59.00 元(含光盘 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 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前言

Linux 开发的前景

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 等面向对象的语言，C 程序的程序设计方法、符号的使用、面向过程的思路和严格的语法这些特点使 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 篇，关于“Linux 开发”的网页有 5 370 000 篇，作为对比关于“Visual C++”的网页只有 6 170 000 篇。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，笔者结合多年的开发经验，编写了本书。

本书特点

1. 提供完善的售后服务：为了帮助大家学习 Linux 系统和 C 程序，我们为广大读者建立了专门的讨论区，网址为 www.rzchina.net。广大读者可以和作者进行交流和讨论。
2. 注重读者的学习与接受情况：在 Linux 系统下编程学习时，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，引导读者一步一步进入 Linux 系统，掌握与 Linux 系统编程相关的一般操作。
3. 代码注释细致，方便学习：在实例代码中，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。例如下面的代码，是循环控制语句章节中的实例。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。

```
#include <stdio.h>                                /*包含需要的头文件。*/
int main()
{
    int i,j;                                       /*定义两个循环变量。*/
    for(i=1;i<=9;i++)                             /*外层循环，乘数。*/
    {
        for(j=1;j<=i;j++)                       /*内层循环，被乘数。*/
        {
            if (j<=i)                             /*判断 i 与 j 的大小，实现排列。*/
            {
                printf("%d*%d=%d ",j , i , i*j ); /*输出乘法式与结果。*/
            }
        }
    }
    printf("\n");                                /*输出一个换行。*/
}
```

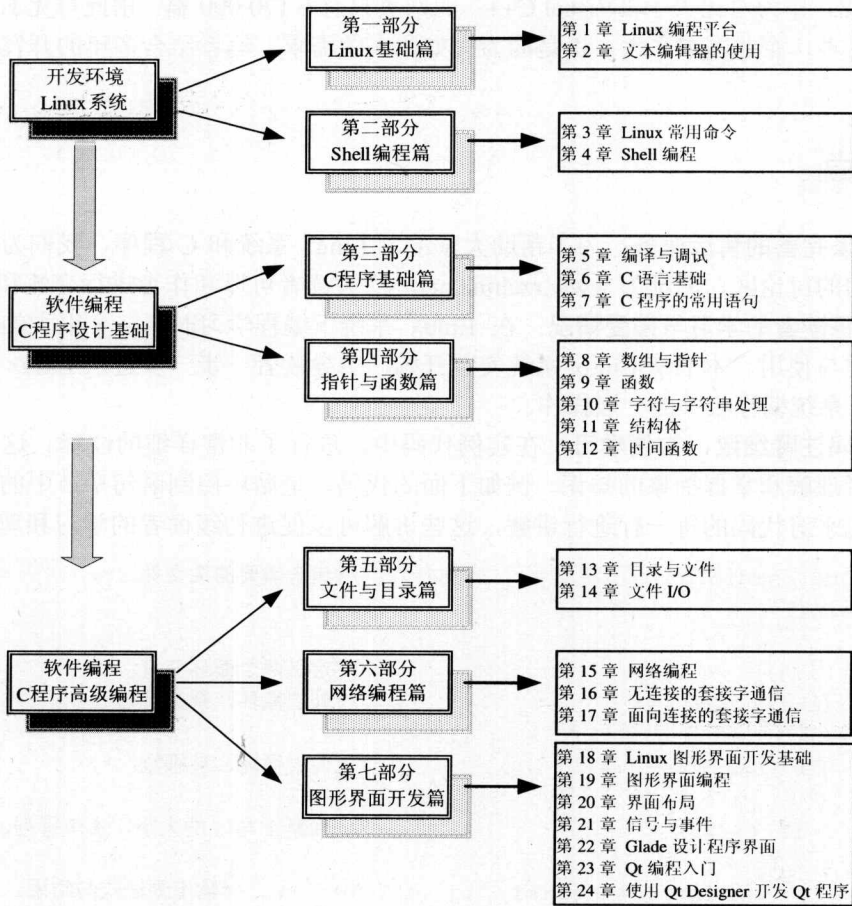
```
}  
}
```

4. **VI 编辑器**：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。

5. **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。

6. **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。

本书包括的内容



本书适合的读者

- C 程序初学者
- 高校计算机专业学生
- 高校毕业设计的学生
- 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

编 者

2008 年 1 月

第一部分 Linux 基础篇

第 1 章 Linux 编程平台

2

本章讲述了 Red Hat Linux 9.0 的安装、VMware 虚拟机的使用、简单 C 程序的编写等问题。其中需要重点理解的是 Linux 系统分区的问题。在分区时需要正确设置各个分区的格式，正确将分区挂载到系统中，并处理好与 Windows 系统的引导问题。通过 C 程序实例的练习，可以初步了解 Linux 系统下 C 程序的编程过程。

1.1 Linux 发行版的介绍	2
1.1.1 Red Hat Linux	2
1.1.2 Debian Linux	2
1.2 Linux 的安装	2
1.2.1 Linux 的分区格式	3
1.2.2 在 Windows 系统中划分 Linux 分区	3
1.2.3 Red Hat 9.0 的安装	6
1.2.4 GRUB 在 Windows 分区中引导 Linux	15
1.3 VMware 虚拟机安装 Linux 系统	17
1.3.1 VMware 软件安装	17
1.3.2 新建虚拟计算机	20
1.3.3 VMware 的使用	24
1.3.4 在 VMware 虚拟机中安装 Linux 系统	26
1.4 第一个 C 程序	28
1.4.1 在 Linux 中编写 C 程序代码	28
1.4.2 编译程序	29
1.5 小结	30

第 2 章 文本编辑器的使用

31

本章讲述了 VIM、GVIM、gedit 三种 Linux 系统下常用的文本编辑器软件。其中 VIM 是一个全字符操作的文本编辑器，学习的重点是各种命令的记忆和使用。gedit 是一个图形图面的文本编辑器，操作非常简单。GVIM 结合了 VIM 命令操作与 gedit 界面操作的两种功能。VIM 是本章学习的难点，在 Linux 全字符界面的管理中，需要灵活运用 VIM 进行文本编辑。

2.1	VIM 的使用	31
2.1.1	VIM 的启动	31
2.1.2	在桌面上创建 VIM 启动器	31
2.1.3	VIM 的工作模式	32
2.1.4	保存与打开文件	33
2.1.5	移动光标	34
2.1.6	插入	36
2.1.7	删除	36
2.1.8	取消	37
2.1.9	保存	37
2.1.10	退出	37
2.1.11	查找	38
2.1.12	替换	38
2.1.13	选项设置	38
2.1.14	调用 shell 命令	39
2.2	VIM 使用实例	39
2.2.1	字符的插入与删除	39
2.2.2	字符的查找与替换	40
2.3	GVIM	41
2.3.1	文件的新建与保存	41
2.3.2	查找与替换	42
2.4	gedit	43
2.4.1	gedit 的启动与打开文本	44
2.4.2	gedit 编辑文件	44
2.4.3	gedit 打印文件	46
2.4.4	gedit 的首选项	47
2.5	小结	50

第二部分 Shell 编程篇

第 3 章 Linux 常用命令

52

本章主要介绍了 Linux 中常用的文件操作命令和系统帮助文档的使用。在以后的编程学习中，需要使用这些命令对文件进行各种操作，其中文件权限的设置是一个难点，在使用这个命令时，需要注意命令参数的作用与含义。读者有学习中遇到的问题，可以使用 man 帮助手册查看相关的帮助信息。

3.1	目录与文件命令	52
3.1.1	查看目录命令 pwd	52
3.1.2	打开目录命令 cd	52
3.1.3	查看目录命令 ls	53
3.1.4	建立文件命令 touch	54
3.1.5	建立与删除文件夹命令 mkdir, rmdir	54
3.1.6	文件复制命令 cp	55
3.1.7	文件移动和更名命令 mv	56
3.1.8	删除文件与目录命令 rm	56
3.1.9	创建链接命令 ln	57
3.1.10	查看文件信息命令 wc	58
3.1.11	文件查看命令 cat	58
3.1.12	查看文件视图命令 head 与 tail	59
3.1.13	文件权限设置命令 chmod	59
3.1.14	查找文件命令 find	61
3.2	man 系统帮助命令	61
3.2.1	man 手册的使用方法	61
3.2.2	帮助文档的结构	62
3.3	info 手册	62
3.4	小结	63

第 4 章 Shell 编程

64

本节讲述了 Shell 程序的基本语法。通过本章的学习，可以编写出简单的 Shell 程序完成各种系统功能。在 Shell 程序的学习中，需要注意局部变量和环境变量的使用、各种符号的使用方法。测试语句与流程控制语句是本章的难点，需要对程序的各种变量进行逻辑判断，执行不同的程序。

4.1	bash 的使用	64
4.1.1	bash 的内部命令与外部命令	64
4.1.2	命令补齐功能	64
4.1.3	命令通配符	64
4.1.4	使用命令的历史记录	65
4.1.5	命令的别名	66
4.2	shell 程序的基本结构	66
4.3	局部变量	67
4.4	环境变量	68
4.4.1	环境变量的查看	68
4.4.2	环境变量的访问	68

4.4.3	环境变量的定义	69
4.4.4	在系统配置文件中定义环境变量	69
4.4.5	位置变量	69
4.5	shell 的运算符	70
4.5.1	变量赋值	70
4.5.2	算术运算符	71
4.6	输入和输出	72
4.6.1	echo 命令输出结果	72
4.6.2	read 命令读取信息	74
4.6.3	文件重定向	75
4.7	引号的使用方法	76
4.7.1	双引号	76
4.7.2	单引号	77
4.7.3	反引号	77
4.7.4	反斜线	77
4.8	测试语句	78
4.8.1	文件状态测试	78
4.8.2	数值测试	79
4.8.3	字符串测试	79
4.8.4	逻辑测试	80
4.9	流程控制结构	80
4.9.1	if 语句	81
4.9.2	if 语句应用实例	82
4.9.3	for 语句	84
4.9.4	for 循环应用实例	85
4.9.5	until 语句	86
4.10	Shell 编程实例	87
4.10.1	程序的功能	87
4.10.2	编写程序的代码	88
4.11	小结	89

第三部分 C 程序基础篇

第 5 章 编译与调试

92

程序的编译和调试是编程的一个重要环节。本章讲解了 Linux 系统中 C 编程的编译器 gcc 和编译器 gdb 的使用。使用 gcc 时,需要对编译进行各种设置,需要理解 gcc 各项参数的作用。gdb

的学习重点是 gdb 单步运行程序的理解，通过程序的单步运行发现程序中的问题。

5.1	编译的概念和理解	92
5.1.1	程序编译的过程	92
5.1.2	编译器	92
5.2	gcc 编译器	93
5.2.1	gcc 编译器简介	93
5.2.2	gcc 对源程序扩展名的支持	93
5.3	C 程序的编译	94
5.3.1	编写第一个 C 程序	94
5.3.2	用 gcc 编译程序	95
5.3.3	查看 gcc 的参数	96
5.3.4	设置输出的文件	97
5.3.5	查看编译过程	97
5.3.6	设置编译的语言	98
5.3.7	-ansi 设置 ANSIC 标准	98
5.3.8	g++编译 C++程序	98
5.4	编译过程的控制	99
5.4.1	编译过程简介	100
5.4.2	控制预处理过程	100
5.4.3	生成汇编代码	101
5.4.4	生成目标代码	102
5.4.5	链接生成可执行文件	102
5.5	gdb 调试程序	103
5.5.1	gdb 简介	103
5.5.2	在程序中加入调试信息	103
5.5.3	启动 gdb	103
5.5.4	在 gdb 中加载需要调试的程序	104
5.5.5	在 gdb 中查看代码	104
5.5.6	在程序中加入断点	105
5.5.7	查看断点	105
5.5.8	运行程序	105
5.5.9	变量的查看	106
5.6	程序调试实例	107
5.6.1	编写一个程序	107
5.6.2	编译文件	108
5.6.3	程序的调试	108

5.6.4	gdb 帮助的使用	111
5.7	gdb 常用命令	111
5.8	编译程序常见的错误与问题	112
5.8.1	逻辑错误与语法错误	113
5.8.2	C 程序中的错误与异常	113
5.8.3	编译中的警告提示	113
5.8.4	找不到包含文件的错误	114
5.8.5	错误地使用逗号	114
5.8.6	括号不匹配错误	114
5.8.7	小括号不匹配错误	115
5.8.8	变量类型或结构体声明错误	115
5.8.9	使用不存在的函数的错误	115
5.8.10	大小写错误	116
5.8.11	数据类型的错误	116
5.8.12	赋值类型错误	116
5.8.13	循环或判断语句中多加分号	116
5.9	小结	117

第 6 章 C 语言基础

118

本章讲解了数据类型、变量赋值与输出、常用运算符等知识。这些知识是编写 C 程序的基础，通过这些知识的学习，可以理解 C 程序的一些概念，编写简单的 C 语言程序。在本章的学习中，数据类型与运算符的使用是重点，需要详细地理解数据类型的概念和输出方式。

6.1	C 程序的基本概念	118
6.1.1	C 程序的基本结构	118
6.1.2	C 程序的一般格式	119
6.1.3	C 程序中的注释	120
6.2	数据类型	120
6.2.1	整型 (int)	121
6.2.2	浮点型 (float)	122
6.2.3	字符型(char)	123
6.2.4	变量名	124
6.2.5	字符 NULL	124
6.3	变量的赋值与输出	124
6.3.1	变量的赋值	125
6.3.2	printf 函数输出变量	125
6.3.3	scanf 函数从键盘读入变量	127

6.4 运算符.....	128
6.4.1 算术运算符.....	128
6.4.2 关系运算符.....	130
6.4.3 逻辑运算符.....	130
6.5 小结.....	131

第 7 章 C 程序的常用语句

132

本章讲述了 C 程序中的流程控制语句。流程控制语句实现了程序的逻辑功能，可以完成各种复杂的程序运算，是编程开发的基础和重点。其中，循环语句和条件语句是本章的重点，需要理解这些语句的结构和执行流程。

7.1 流程控制语句.....	132
7.1.1 if 条件语句.....	132
7.1.2 if 语句的嵌套.....	134
7.1.3 switch 选择执行语句.....	136
7.1.4 for 循环语句.....	138
7.1.5 for 循环的嵌套.....	140
7.1.6 for 循环应用实例：输出九九乘法口诀表.....	141
7.1.7 while 循环语句.....	142
7.1.8 do while 循环语句.....	144
7.1.9 转移控制语句：continue.....	145
7.1.10 转移控制语句：break.....	146
7.1.11 转移控制语句：return.....	146
7.2 流程控制语句实例.....	147
7.2.1 三个数字的排序.....	147
7.2.2 解一元二次方程.....	149
7.3 两种特殊语句结构.....	151
7.3.1 ?三元操作符.....	151
7.3.2 块语句.....	152
7.4 小结.....	152

第四部分 指针与函数篇

第 8 章 数组与指针

154

数组与指针的知识是 C 程序中的重要内容，可以实现很多复杂的功能和算法。数组与指针这两个概念的理解是个难点，需要建立起直观形象的空间概念。在实际开发时各种对象与算法常常

通过指针与数组建立起来的,需要使用各种复杂的指针和数组。在程序设计练习时需要注意数组与指针的应用技巧。

8.1	数组的理解与操作	154
8.1.1	什么是数组	154
8.1.2	数组的定义与访问	155
8.1.3	数组使用实例	156
8.2	指针	157
8.2.1	指针的理解	157
8.2.2	指针操作符	157
8.2.3	指针的定义与访问	158
8.2.4	指针使用实例	159
8.3	数组与指针	160
8.3.1	数组与指针的关系	160
8.3.2	指针的算术运算	160
8.3.3	字符数组与字符串	161
8.4	二维数组与多维数组	162
8.4.1	二维数组的理解	163
8.4.2	二维数组与指针	164
8.5	实例	165
8.5.1	学生成绩统计实例	165
8.5.2	冒泡法排序实例	166
8.5.3	统计字符串中字符	168
8.5.4	小写字母转换成大写字母	170
8.5.5	指针访问数组	171
8.6	常见问题	172
8.6.1	错误的写操作	172
8.6.2	指针的错误赋值	173
8.6.3	数组指针的越界错误	174
8.7	小结	175

第9章 函数

176

本章讲述了C程序中的函数操作。学习难点是自定义函数的理解和使用。函数可以把一个功能或模块封装起来,程序可以方便地使用这个已经封装的功能。在自定义函数中,需要注意函数的参数与返回值。指针与数组作为一种特殊的数据类型可以作为函数的参数和返回值,这一知识比较难以理解,读者在学习时需要根据实例理解函数中指针和数组的作用。

9.1	函数的理解	176
-----	-------	-----

9.1.1	什么是函数	176
9.1.2	系统函数（库函数）与用户自定义的函数	177
9.1.3	函数的返回值	178
9.1.4	无参函数	179
9.1.5	有参函数	180
9.1.6	函数参数实例	180
9.2	自定义函数	181
9.2.1	自定义函数的编写	181
9.2.2	函数中调用函数	183
9.2.3	函数的声明	184
9.2.4	递归函数	186
9.2.5	main 函数的参数 argc 与 argv	187
9.3	函数与指针	188
9.3.1	值调用与引用调用	188
9.3.2	引用调用与指针	189
9.3.3	指针参数简单实例	190
9.3.4	自定义函数中指针使用实例	191
9.3.5	数组作为参数	192
9.3.6	数组作为函数参数实例	194
9.4	返回值	195
9.4.1	函数返回值的类型	195
9.4.2	函数返回指针	196
9.4.3	函数返回指针实例	197
9.4.4	函数返回结果到参数	198
9.5	库函数	198
9.5.1	库函数的种类	198
9.5.2	库函数包含文件	199
9.5.3	头文件使用实例	200
9.6	小结	201

第 10 章 字符与字符串处理

202

本章讲述了 C 程序中的字符测试、字符串转换、字符串操作三方面的知识。在 C 程序中常常需要使用这些知识对字符和字符串进行测试和操作。

10.1	字符测试函数介绍	202
10.1.1	数字或字母测试函数 isalnum	202

10.1.2	字母测试函数 isalpha	203
10.1.3	可打印字符测试函数 isgraph	204
10.1.4	大小写字母测试函数 islower 和 isupper	205
10.1.5	数字测试函数 isxdigit	206
10.1.6	符号测试函数 ispunct	207
10.1.7	其他字符测试函数	208
10.2	字符测试函数综合实例	208
10.2.1	统计字符串中各类字符的个数	208
10.2.2	判断电话号码与姓名是否正确	210
10.3	字符串转换	212
10.3.1	C 程序中的字符串	212
10.3.2	字符串转换成浮点型函数 atof	213
10.3.3	字符串转换成整型函数 atoi	214
10.3.4	字符串转换成长整型函数 atol	215
10.3.5	将浮点型数转换成字符串函数 ecvt	216
10.3.6	字母的大小写转换函数 tolower 和 toupper	217
10.3.7	其他字符串转换函数	218
10.4	字符串比较	218
10.4.1	字符串比较函数 bcmp	219
10.4.2	字符串大小比较函数 memcmp	220
10.4.3	忽略大小写比较字符串函数 strncasecmp	221
10.5	字符串复制	222
10.5.1	字符串复制函数 bcopy	222
10.5.2	字符串复制函数 memccpy	223
10.5.3	字符串复制函数 strcpy	223
10.5.4	字符串复制函数 strncpy	224
10.6	字符串的清理与填充	225
10.6.1	字符串清理函数 bzero	225
10.6.2	字符串填充函数 memset	226
10.7	字符串查找	227
10.7.1	字符查找函数 index 与 rindex	227
10.7.2	字符查找函数 memchr	228
10.7.3	字符查找函数 strchr 与 strrchr	228
10.8	字符串的连接与分割	229
10.8.1	字符串连接函数 strcat	229
10.8.2	字符串分割函数 strtok	230

10.9 其他字符串函数.....	231
10.9.1 字符串长度函数 strlen.....	231
10.9.2 允许出现字符查找函数 strspn.....	232
10.9.3 不允许出现字符查找函数 strcspn.....	233
10.10 小结.....	234

第 11 章 结构体

235

本章讲解了 C 程序的结构体操作。结构体是对现有数据类型的扩充，可以把有实际含义的事物定义为一个结构体数据类型。程序中使用结构体数据类型可以方便地处理有实际含义的数据。本章的内容中，结构体指针和与结构体相关的函数是个难点，要掌握用结构体指针访问结构成员变量的方法，需要仔细理解结构体指针与结构体函数的作用。本章中的最后一个实例是结构体的一个综合运用，处理了复杂的数据与功能，建议学习时完成这个程序的编写。

11.1 结构体的操作.....	235
11.1.1 结构体的理解.....	235
11.1.2 结构体的定义.....	235
11.1.3 结构体的访问.....	237
11.1.4 结构体数组.....	238
11.1.5 结构体使用实例.....	239
11.2 结构体与指针.....	241
11.2.1 结构体指针的定义.....	241
11.2.2 结构体指针的访问.....	242
11.2.3 结构体作为函数的参数.....	244
11.2.4 结构体指针作为函数的参数.....	245
11.2.5 结构体作为函数的返回值.....	247
11.2.6 结构体指针作为函数的返回值.....	248
11.3 结构体实例.....	250
11.3.1 程序的需求分析.....	250
11.3.2 程序中的函数.....	251
11.3.3 程序中的结构体与全局变量.....	251
11.3.4 头文件和函数声明.....	252
11.3.5 定义结构体和全局变量.....	252
11.3.6 显示学生信息的函数.....	252
11.3.7 程序的选择菜单.....	253
11.3.8 学生信息输入函数.....	254
11.3.9 按姓名查找函数.....	255