

· 碁峯 ·

www.gotop.com.tw

The Art Of Pointer: Beautiful Star

指针的编程艺术

第二版



蔡明志 著
李强 改编

人民邮电出版社
POSTS & TELECOM PRESS



· 碁峯 ·

www.gotop.com.tw

The Art Of Pointer: Beautiful Star

指针的编程艺术

第二版



蔡明志 著
李强 改编

人民邮电出版社
北京

图书在版编目(CIP)数据

指针的编程艺术：第2版 / 蔡明志著. — 北京：
人民邮电出版社，2013.1
ISBN 978-7-115-29426-5

I. ①指… II. ①蔡… III. ①C语言—程序设计
IV. ①TP312

中国版本图书馆CIP数据核字(2012)第215763号

版 权 声 明

本书为台湾碁峰资讯股份有限公司独家授权的中文简化字版本。本书专有出版权属人民邮电出版社所有。在没有得到本书原版出版者和本书出版者书面许可时，任何单位和个人不得擅自摘抄、复制本书的一部分或全部以任何形式（包括资料和出版物）进行传播。

本书原版版权属碁峰资讯股份有限公司。

版权所有，侵权必究。

内 容 提 要

“指针”（Pointer），这是个让人又爱又难受的主题。常有人说：“指针好难啊，都不知道该怎么学……”。有许多人学到指针时，便裹足不前。基本原因是没有合适的图书来指导学习指针及其相关的问题。

本书内容精彩无比，除了对C语言的指针有深入的探讨外，同时也对与指针有异曲同工之妙的reference 详加讨论。本书以指针为核心，讲解了指针与变量、指针与数组、指针与函数、指针与字符串、指针与结构、链表、指针与文件等编程技术，可以帮助读者快速进入指针的世界，是一本难得的指导书。

本书适合程序员阅读，也适合作为大中专院校的学习用书以及培训学校的教材。

指针的编程艺术（第二版）


-
- ◆ 著 蔡明志
 - 改 编 李 强
 - 责任编辑 张 涛
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
 邮编 100061 电子邮件 315@ptpress.com.cn
 网址 <http://www.ptpress.com.cn>
 北京昌平百善印刷厂印刷
 - ◆ 开本：800×1000 1/16
 印张：27.75
 字数：785千字 2013年1月第1版
 印数：1-3 000册 2013年1月北京第1次印刷
 著作权合同登记号 图字：01-2012-4607号

ISBN 978-7-115-29426-5

定价：65.00元（附光盘）

读者服务热线：(010)67132692 印装质量热线：(010)67129223

反盗版热线：(010)67171154



序言

C 语言的特色就是“指针”(Pointer)，这是个让人又爱又难受的主题。有一句话是这么说的，学过 C，而不会指针，那只能说你看过 C。熟悉指针的人，会将指针比喻成天上那一颗最美的星星，而让不懂指针的人，头上冒星星，同样是星星，但却是完全不同的境遇。

“指针好难啊，都不知道该怎么学……”，常常有人会对我讲这句话，并问我有没有好方法可以“顿悟”它。有许多人学到指针时，便裹足不前。基本原因是没有专门的图书讨论指针及其相关的问题。基于此，我便开始规划、撰写一本以指针为中心的书，让对指针不太懂的人，可以轻易了解指针的运作原理，让懂得指针的人可以精益求精，它是你学习与深入了解指针的最佳读物。

本书内容精彩无比，除了对 C 语言的指针有深入的探讨外，同时也对与指针有异曲同工之妙的 **reference** 详加讨论，所以本书不仅有 C，还包括 C++、C++/CLR、Java、Visual C#及 Visual Basic 5 种编程语言的 **reference** 相关主题。其中要和大家分享心得的是，原本我对 Visual Basic 没什么好感，但借此机会接触它，发现它很好用的，若你的感觉和我先前一样，不妨试试看，相信你会喜欢它。

本书是以指针为核心，以 **reference** 为辅助。在 C 程序语言篇中，从内存的基本概念开始展开这一次的旅程，其中会经过指针与变量、指针与数组、指针与函数、指针与字符串、指针与结构、链表、指针与文件等，希望有效地引导读者进入指针的世界。为了验证指针与 **reference** 的相似功能，之后的语言篇着重介绍如何用 **reference** 实现指针可以完成的事项，以 **call by reference** 完成两数对调为例展开介绍。

本书的第七篇是比较篇，将 C、C++、C++/CLR、Java、Visual C#及 Visual Basic 6 种编程语言在程序语言基本架构上的主题做一比较，希望此篇可以让你一窥编程语言之美。


注：附录 A 和附录 B 及源程序见光盘。

写本书的心情不知为什么，觉得好愉快，因为读者可以通过本书轻轻松松地了解什么是指针，并告诉我，指针是天上那一颗最闪亮的星星。

喜欢我的读者也不在少数，有些人常会跟我说：“C 先生，我是看你的书长大的”。真诚地希望读者可以从书中获取一些知识。再一次谢谢你们，一路的陪伴、支持与鼓励，让我有动力再出发。本书联系邮箱 mjtsai168@gmail.com，编辑联系邮箱 zhangtao@ptpress.com.cn。

Dedicated to

*My wife Linda,
My daughters Amy and Jennifer*



目 录

第一篇 C 语言篇

第一章 内存的基本概念	3
1-1 变量	4
1-2 一维数组	5
1-3 二维数组	9
1-4 结构	11
1-5 练习	12
第二章 指针与变量	15
2-1 一颗星星：表示一把钥匙	16
2-2 两颗星星：表示两把钥匙	19
2-3 三颗星星：表示三把钥匙	20
2-4 练习	22
第三章 指针与数组	24
3-1 指针与一维数组	25
3-2 指针与二维数组	30
3-3 数组指针	33
3-4 为什么 <code>parr</code> 等同于 <code>*parr</code> ?	35
3-5 指向数组的指针	37
3-6 多重指针	42
3-7 命令行参数	49
3-8 改错题	52
3-9 练习	53
3-10 程序实战	60
第四章 指针与函数	61
4-1 函数简介	62
4-2 两数对调	62
4-3 地址调用	64
4-4 函数指针	69

4-5	指针函数	71
4-6	改错题	72
4-7	练习	76
4-8	程序实战	76
第五章	指针与字符串	78
5-1	字符串指针	79
5-2	常用的字符串库函数	82
5-3	改错题	94
5-4	程序实战	95
第六章	指针与结构	96
6-1	指向结构变量的指针	97
6-2	结构指针与自增运算符	101
6-3	结构与函数	104
6-4	自引用结构	107
6-5	改错题	114
6-6	练习	122
第七章	链表——C 程序实战	125
7-1	单链表	126
7-2	双链表	131
7-3	链表的应用——堆栈的实战	136
7-4	链表的应用——队列的实战	140
7-5	程序实战	144
第八章	指针与文件	145
8-1	fprintf 与 fscanf 函数	146
8-2	fwrite 与 fread 函数	148
8-3	当前文件的指针指向何处	151
8-4	fgetpos 与 fsetpos 函数	154
8-5	rewind	156
8-6	程序实战	157

第二篇 C++ 语言篇

第一章	Visual C++ 2010 使用简介	161
------------	-----------------------------	------------

第二章 C++的 reference	165
2-1 变量与 reference	166
2-2 两数对调	167
2-3 结构与 reference	171
2-4 类与 reference	175
2-5 练习	180
2-6 程序实战	181
第三章 C++的 this 指针	182
3-1 this 指针	183
3-2 股票买卖范例程序	186
3-3 程序实战	190
第四章 C++的 new 与 delete	191
4-1 new、delete 与变量	192
4-2 new、delete 与数组	194
4-3 new、delete 与结构	196
4-4 new、delete 与类	198
4-5 程序实战	201
第五章 C++的复制构造函数	202
5-1 复制构造函数	203
5-2 重载指定运算符	208
5-3 程序实战	212
第六章 虚函数、指针与 reference	213
6-1 虚函数	214
6-2 程序实战	218
第七章 链表——C++程序实战	219
7-1 单链表	220
7-2 双链表	227
7-3 堆栈	234
7-4 程序实战	238
第三篇 C++/CLR 语言篇	
第一章 Visual C++/CLR 使用简介	241

第二章	浅谈 C++/CLR 程序设计	244
2-1	可追踪的 reference	245
2-2	可追踪的 handle	245
2-3	数组	247
2-4	Interior 指针	250
2-5	函数	251
2-6	程序实战	259
第三章	Visual C++/CLR 中 链表的使用	260
3-1	用单链表处理数据	261
3-2	用双链表处理数据	265
3-3	使用链表操作堆栈	269
3-4	程序实战	273
第四篇 Java 语言篇		
第一章	JCreator 使用简介	277
1-1	安装 JCreator 编译程序	278
1-2	编写、编译与执行程序	280
1-3	程序调试(debug)	282
1-4	新建项目文件	283
第二章	对象、字符串与 reference	285
2-1	对象与 reference	286
2-2	字符串与 reference	287
2-3	常用的字符串库函数	290
2-4	程序实战	294
第三章	数组与 reference	295
3-1	数组简介	296
3-2	数组复制	301
3-3	如何传递数组	304
3-4	二维数组	309
3-5	程序实战	312
第四章	类与 reference	313
4-1	类与 reference	314
4-2	this 指针	317

4-3	复制构造函数	319
4-4	程序实战	325
第五章 Java 程序中链表的使用		326
5-1	单链表	327
5-2	双链表	332
5-3	堆栈的使用	338
5-4	程序实战	344

第五篇 Visual C# 2010 语言篇

第一章	Visual C#使用简介	347
第二章	变量与 reference	350
2-1	值调用	351
2-2	call by reference	352
2-3	特殊的参数 out	353
2-4	程序实战	354
第三章	数组与 reference	355
3-1	找出一维数组的最大值	356
3-2	一维数组的加和	357
3-3	二维数组的加和	358
3-4	不规则数组的加和	359
3-5	程序实战	360
第四章	Visual C#中链表的使用	361
4-1	数据的转换	362
4-2	单链表	363
4-3	双链表	368
4-4	使用链表操作堆栈	374
4-5	程序实战	378

第六篇 Visual Basic 2010 语言篇

第一章	Visual Basic 使用简介	381
第二章	函数与 reference	384
2-1	call by value	385

2-2	call by reference	385
2-3	程序实战	386
第三章 数组与 reference		387
3-1	找出数组的最大值	388
3-2	数组的加和	389
3-3	不规则的数组	392
3-4	程序实战	393
第四章 Visual Basic 中链表的使用		394
4-1	链表的基本概念	395
4-2	单链表	398
4-3	双链表	404
4-4	使用链表操作堆栈	409
4-5	程序实战	415

第七篇 比较篇

附录 A	答案 (见光盘)	433
附录 B	术语 (见光盘)	557

PART

1

C 语言篇

C 语言最早是在 AT&T(American Telephone and Telegraph, 美国电话与电报公司)内部使用的语言, 但为了发布 UNIX 操作系统, 不得不公开这一程序语言。主导这语言的两个灵魂人物, 分别是 Dennis Ritchie 与 Brian Kernighan。ANSI C 已于 1989 年公布。Dennis Ritchie 生于 1941 年, 哈佛大学数学博士, 是 C 语言之父。Brian Kernighan 目前在普林斯顿大学计算机科学系(Department of Computer Science, Princeton University)任教。

C 语言(1971 年)是受到 CPL(1962 年)、BCPL(在 1969 年由 Martin Richards 所编写的)和 B(在 1970 年由 Ken Thompson 所编写的)等程序语言的直接或间接的影响和演化而产生的。

下图分别是这两位大师所编写的 C 语言教材《The C Programming Language》。想要了解更多的读者, 可访问他们的网站:

Dennis Ritchie: <http://plan9.bell-labs.com/cm/cs/who/dmr/>

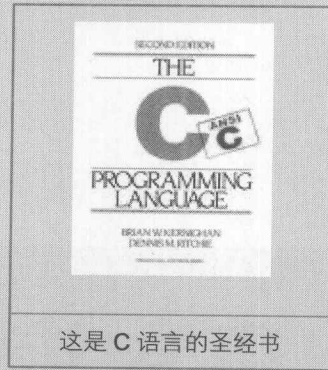
Brian Kernighan: <http://www.cs.princeton.edu/~bwk/>

1983 年, Ritchie 和 Thompson 获得图灵奖 (Turing Award), 这是计算机界的诺贝尔奖。不幸的是 Ritchie 于 2011 年 10 月 12 日逝世, 享年 70 岁; 他的丰功伟绩值得世人敬仰。

C 语言最大的特色是, 它拥有功能非常强大的指针(pointer), 这是让很多学习者又爱又恨的功能。很多人学到这个功能时, 感到非常困惑, 因而对程序语言不感兴趣。但有些人却将指针看作是天上闪亮的星星, 从而爱上程序设计这玩意儿。我不晓得你是哪一种情况, 希望你是后者。

若你是前者, 也没关系, 让这本书为你解开指针的神秘面纱。本书涵盖了与指针相关的主题, 包括指针与变量、指针与数组、指针与函数、指针与字符串、指针与结构、指针与文档等,

由浅入深，并配合图示加以说明，以期你能了解指针的精髓，从而将指针看作是天上最灿烂的星星。这也是笔者写这本书的最大动机。



同时也要告诉读者，C 除了有强大的指针之外，也是 UNIX 与 Linux 操作系统的核心语言，也就是说，这些操作系统是由 C 所编写而成的。所以有人说，C 语言是系统程序语言 (system programming language)。不仅如此，它也是编写有关网络、通信及嵌入式程序的最佳程序语言。

总而言之，好好地学习 C 语言，对你的未来一定会有加分的效果。

内存的基本概念

1-1 变量

1-2 一维数组

1-3 二维数组

1-4 结构

1-5 练习

当程序定义一个变量时，编译程序根据其数据类型分配适当大小的内存给它，以便存储数据。这就如同地址一般，邮差根据收件人的地址，将信件投入正确的信箱。

1-1 变量

程序以变量名称代表某一事项，就好比每一个人都有姓名一般。如果有人喊“蔡明志”，那我一定会说“到”，因为这个名字就代表我的意思。我们就从变量开始谈起。请参阅范例 address-3。

范例 address-3

```
/* address-3.c */
#include <stdio.h>
#include <conio.h>
int main()
{
    int x=100, y=200, z=300;
    printf("x 变量的地址是 %p\n", &x);
    printf("y 变量的地址是 %p\n", &y);
    printf("z 变量的地址是 %p\n\n", &z);

    printf("x 变量的内容是 %d\n", x);
    printf("y 变量的内容是 %d\n", y);
    printf("z 变量的内容是 %d\n", z);
    getch();
    return 0;
}
```

输出结果

```
x 变量的地址是 0012FF60
y 变量的地址是 0012FF54
z 变量的地址是 0012FF48

x 变量的内容是 100
y 变量的内容是 200
z 变量的内容是 300
```

程序中有关 x、y、z 变量的地址，如下图所示。

0022FF42	0012FF48	0012FF54	0012FF60	0012FF66
	300	200	100	

如果你想要知道变量所在内存的地址，只要在变量前加上地址运算符(&)即可。我们一定要确实了解每一表达式(expression)所表达的意思是什么，是变量所在的内存地址(address)，还是变量值(value)。

如果要使用 scanf 函数，从键盘输入数据给一个变量，需要注意哪些问题呢？我们以范例

address-5 来说明。

范例 address-5

```
/* address-5.c*/
#include <stdio.h>
#include <stdlib.h>

int main()
{
    double x, y, average;
    printf("请输入浮点数 x 与 y: ");
    scanf("%lf %lf", &x, &y);
    average = (x+y)/2;
    printf("x 与 y 的平均数为: %.2f\n", average);
    system("PAUSE");
    return 0;
}
```

输出结果

```
请输入浮点数x与y: 12.34 45.67
x与y的平均数为: 29.01
```

利用 `scanf` 函数，从键盘输入数据给变量，要特别注意的是，`scanf` 函数的参数是要赋值变量的地址，因此，会用到地址运算符(&)。这好比邮差一定要根据收件人的地址，才能将这封信放到收件人的信箱。这么打比方，是希望你能更好地了解地址的概念。程序中变量 `x` 与 `y` 的数据类型是 `double`，所以其对应的格式都是 `%lf`。

在此要特别提醒读者的是，本书所输出的内存地址会与你所得到的结果不同，因为我们使用不同的计算机设备。使用地址的目的是让你了解变量存储位置。

1-2 一维数组

我们把焦点转到一维数组，范例 `addressArray-5` 展示了如何调用 `scanf` 函数，并从键盘向数组输入数据。

范例 addressArray-5

```
/* addressArray-5.c */
#include <stdio.h>
#include <conio.h>
int main()
{
    int x[5];
    int i;

    for(i=0; i<5; i++)
    {
```