

计算机综合培训教程



入门与进阶

丛书

网络语言



中国IT培训工程编委会 编
香港恒明出版有限公司 承制
珠海网垠科技发展有限公司

珠海出版社



全国千余家著名电脑培训学校联袂推出



计算机综合培训教程

C 语言入门与进阶

中国 IT 培训工程编委会 编

珠海出版社

图书在版编目(CIP)数据

计算机综合培训教程/中国 IT 培训工程编委会编. —珠海:珠海出版社,2002.6

ISBN7-80607-823-1/TP.12

I. 计... II. 中... III. 计算机综合培训-教程 IV. TP.12

计算机综合培训教程

作 者 ■ 中国 IT 培训工程编委会

选题策划 ■ 孙建开

终 审 ■ 成 平

责任编辑 ■ 孙建开 雷良波

封面设计 ■ 非凡创意

出版发行 ● 珠海出版社

社 址 ● 珠海香洲梅华东路 297 号二层

电 话 ● 2222759 邮政编码 ● 519001

印 刷 ▲ 河南省瑞光印务股份有限公司

开 本 ▲ 787 × 1092mm 1/16

印 张 ▲ 500 字数 ▲ 8200 千字

版 次 ▲ 2002 年 6 月第 1 版

2002 年 6 月第 1 次印刷

印 数 ▲ 1 - 5000 册

ISBN7-80607-823-1/TP.12

总 定 价:625.00 元(全二十五册)

版权所有:翻印必究

《电脑综合培训教程—入门与进阶》

序言

电脑这玩意，看起来容易，用起来难。各种硬件，一拨接一拨地升级；各种软件，一版接一版地换代，往往弄得我们头昏脑胀，无从下手。

“心动不如行动”。道路只有一条：动手去用！选择您想用的软硬件和一本配套的好书，然后坐在电脑前面，开机、安装，按照书中的指示去用、去试。很快您就会发现您的电脑也有灵气了，您也能成为一名出色的舵手，自如地在电脑海洋中航行了。

现在的个人电脑虽然功能强大、用途广泛，但大多数人学电脑不外出于两种目的：一种是作为玩乐之用；另一种是将之作为谋生的工具。本丛书从读者的角度出发，引领大家在工作上善用电脑。

当工作上遇到逆境时，待业者固然应该趁此机会充实自己，就算是在职者也应多掌握一些实用技能以提升自己的竞争力。在科技发展一日千里的今天，熟悉电脑操作已成为很多工作必备的入职要求。

不过，对很多人来说，要把家中的个人电脑变为谋生工具，或好好地驾驶办公室里的电脑设备是说来容易，做起来却无从入手的一件事。因此，网垠公司基于多年策划电脑书的经验，结合中国目前的实际情况，与珠海出版社一起策划出版了这套丛书，帮助读者尽量发挥电脑的实用性能及善用互联网上的无尽资源，以达到转危为安、自强不息的目标。

为工作而学习电脑可分为两种情况来谈。由于在今天的办公室里，文秘、会计和资料库管理等工序都已电脑化，电脑已变成如从前的笔墨或算盘那样通行，不懂操作应用软件就和不懂写字、不谙计算般寸步难行。因此，对有些人来说，掌握电脑的基本概念，并能操作常用的办公室软件是首要目标。

但对另一些人来说，电脑也是提高效率的工具。当今时代步伐急促，一天 24 小时都不够用。要做到“工夫做得细，时间花得少”，就要尽量善用电脑资源，无尽延伸个人电脑的能力，使你能掌握资讯、运筹帷幄，充分发挥个人的创造和策划能力。

为能照顾上述有不同需要的读者，《电脑入门与进阶》丛书的选题范围涵盖电脑在工作上的各方面应用，帮助你使用电脑做好手头上的工作，提升自己面对逆境的适应能力，甚至利用电脑创造财富、攀登事业的高峰。



本书为您而备

本丛书明确定位于初、中级用户。不管您以前是否使用过本丛书所述的软硬件，这套书对您都非常合适。

本丛书名中的“入门”是指，对于每个软硬件的讲解都从必备的基础知识和基本操作开始，新用户无需参照其他书即可轻松入门；老用户亦可从中快速了解新版本的新特性和新功能，自如地踏上新的台阶。至于书名中的“进阶”，则蕴涵了图书内容的重点所在。以我们的经验，当前软硬件的功能日趋复杂，不学到一定的深度和广度是难以在实际工作中应付自如的。因此本丛书在让读者快速入门之后，就以大量明晰的操作步骤和典型的应

用实例，教会读者更丰富全面的软硬件技术和应用技巧，使读者真正对所学软硬件融会贯通、熟练在手。



丛书特点:

精选新的、热门的软硬件，紧随版本更新换代，连续推出配套图书。由“入门”起步，注意“进阶”，使新手老手立马变成行家里手。语言生动、简练，详细的操作步骤和醒目可爱的屏幕画面犹如你身临现场。立足实用，侧重技术精华的分析和操作技能的掌握，触类旁通。学教两相宜，既可作为个人无师自通的资料，又可作为各类电脑学校的首选教材。



妙趣小栏目:



新知

——最新的知识，令您耳目一新。



注意

——提醒操作中应注意的有关事项，避免错误的发生，让您少一些傻眼的时刻和求救的烦恼。



提示

——提示可以进一步参见的章节，以及有关某个内容的详细信息，使您可深可浅，收放自如。



窍门

——指点一些捷径，透露一些高招，让您事半功倍，技高一筹。



练一练

——精心设计各种操作练习，您只要照猫画虎，试上一试，就不仅能在您的电脑上展现出书中的美妙画面，还能了解书中未详述的其他实现方法和可能出现的其他操作结果，随处可见的“练一练”，让您边学边用，时有所得，常有所悟。

本丛书是中国 IT 培训工程丛书的重要组成部分，该工程的“流行软件全面学习教程”、“电脑超级培训学院”、“全国计算机等级考试完全版”、“中小学信息技术标准教程”、“好孩子学电脑”等丛书一经面市，即受到全国读者的热烈欢迎。《电脑综合培训教程—入门与进阶》的推出，相信更加受到读者的热爱。

《中国 IT 工程丛书》编委会

2002 年 6 月

本书导读

C 语言是 1972 年由美国的 Dennis Ritchie 设计发明的,由早期的编程语言 BCPL (Basic Combind Programming Language) 发展演变而来。因其强大的功能,已成为最受欢迎的编程语言之一。

本书共分为十章,详尽介绍了 C 语言的基础知识、使用和编程技巧。

第一章介绍了 C 语言程序和算法

第二章介绍了 C 语言数据类型、运算符及表达式

第三章介绍了 C 语言程序控制语句

第四章介绍了 C 语言函数的使用

第五章介绍了 C 语言数组

第六章介绍了 C 语言指针技术

第七章介绍了 C 语言结构体数据类型

第八章介绍了 C 语言文件系统

第九章介绍了 C 语言编程技巧

第十章介绍了 C++ 程序基础知识

本书附常用字符与 ASCII 代码对照表, C 语言函数大全, C 常用的编辑命令, C 编译、连接和运行时的常见错误。

本书面向初、中级用户,着眼于培养读者编程的能力,讲解以实例为主,力求简明易懂,引导学生掌握基本知识点和程序设计技巧,并在理解的基础上拓展程序的功能,举一反三,进而可以参照相关实例编写程序,最后达到独立设计程序的目的。

本书适合于大专院校、培训班师生学习使用,也适合欲从事软件编程人士使用。

内 容 简 介

C 语言是1972年由美国的Dennis Ritchie设计发明的,由早期的编程语言BCPL (Basic Combind Programming Language)发展演变而来。因其强大的功能,已成为最受欢迎的编程语言之一。

本书面向初、中级用户,着眼于培养读者编程的能力,讲解以实例为主,力求简明易懂,引导学生掌握基本知识点和程序设计技巧,并在理解的基础上拓展程序的功能,举一反三,进而可以参照相关实例编写程序,最后达到独立设计程序的目的。

本书适合于大专院校、培训班师生学习使用,也适合欲从事软件编程人士使用。

目 录

第1章 C语言程序和算法	1
1.1 程序设计语言发展史	2
1.2 C语言的产生和发展	3
1.3 C语言的几个特点	4
1.4 C语言的结构	5
1.4.1 入门实例	5
1.4.2 函数库	8
1.4.3 开发步骤	8
1.4.4 关键字	9
1.5 算法概述	9
1.5.1 流程图与算法的结构	11
1.5.2 N-S图算法表示法	13
1.5.3 PAD图算法表示法	14
第2章 C语言数据类型、运算符及表达式	16
2.1 基本数据类型	17
2.2 常量与变量入门	18
2.2.1 标识符	18
2.2.2 常量	19
2.2.3 变量	19
2.3 整型数据入门	19
2.3.1 整型常量	19
2.3.2 整型变量	20
2.4 实型数据入门	21
2.4.1 实型常量	21
2.4.2 实型变量	21
2.5 字符型数据入门	22
2.5.1 字符常量	22
2.5.2 字符串常量	22
2.5.3 转义字符	23
2.5.4 符号常量	23
2.5.5 字符变量	24
2.6 运算符入门	25
2.6.1 算术运算符	25
2.6.2 自增和自减运算符	25
2.6.3 关系和逻辑运算符	26

2.6.4	位操作符	27
2.6.5	问号操作	29
2.6.6	逗号操作符	30
2.6.7	操作符优先级	30
2.7	表达式入门	31
2.7.1	量的类型变换	31
2.7.2	cast 操作符	32
2.7.3	空格符与括号	32
2.7.4	简写赋值语句	32
第 3 章	C 语言程序控制语句	34
3.1	程序基本结构	35
3.2	输入输出函数	35
3.2.1	scanf () 函数	35
3.2.2	printf () 函数	37
3.2.3	getchar () 与 putchar () 函数	40
3.2.4	复数加法实例	41
3.3	条件控制语句	42
3.3.1	if 条件语句	42
3.3.2	switch 开关语句	49
3.3.3	一元二次方程解题实例	51
3.4	循环控制语句	53
3.4.1	while 当型循环	53
3.4.2	do...while 直到型循环	56
3.4.3	for 语句	57
3.4.4	break 与 continue 语句	61
3.4.5	哥德巴赫猜想验证实例	62
第 4 章	C 语言函数的使用	65
4.1	函数说明与返回	66
4.1.1	类型说明符	66
4.1.2	return() 语句	67
4.2	函数的作用域	69
4.2.1	局部变量的使用	69
4.2.2	全局变量的使用	70
4.2.3	动态存储变量的使用	72
4.2.4	静态存储变量的使用	72
4.3	函数的参数及传递	73
4.3.1	形参与实参	73

4.3.2 参数传递方法	73
4.4 递归函数	74
4.5 编写函数的要点	75
4.5.1 使用参数传递	75
4.5.2 使用内嵌代码	76
4.6 文件与函数库	76
4.6.1 文件的规模	76
4.6.2 文件的组织	77
4.6.3 函数库的建立	77
4.7 C 语言预处理程序命令	77
4.7.1 预处理程序命令	77
4.7.2 #define 命令	78
4.7.3 #error 命令	79
4.7.4 #include 命令	79
4.7.5 条件编译命令	80
4.7.6 #undef 命令	82
4.7.7 #line 命令	83
4.7.8 #pragma 命令	83
4.7.9 预定义宏	83
4.7.10 注释的使用	83
4.8 字符串显示实例	84
第 5 章 C 语言数组	88
5.1 一维数组的使用	89
5.1.1 一维数组的函数传递	89
5.1.2 一维数组的字符串用法	90
5.2 二维数组的应用	92
5.2.1 二维数组的形式	92
5.2.2 二维字符数组的建立	95
5.3 多维数组概述	96
5.4 初始化数组	97
5.4.1 数组的初始化方法	97
5.4.2 变长数组的初始化	98
5.5 比赛选手评分实例	99
第 6 章 C 语言指针技术	103
6.1 指针变量	104
6.2 定义与引用指针变量	105
6.2.1 定义指针变量	105

6.2.2 指针变量引用形式	106
6.3 指针运算符及表达式	107
6.3.1 入门实例	107
6.3.2 指针变量的参数应用	109
6.4 数组的指针应用	110
6.4.1 一维数组的指针表示	110
6.4.2 二维数组的指针表示	113
6.4.3 指针变量作函数的参数	116
6.4.4 字符数组的指针表示	124
6.5 指针的动态内存分配	126
6.6 指针数组的定义	128
6.7 双重指针的引用	135
6.8 带参 main 函数的使用	138
第 7 章 C 语言结构体数据类型	143
7.1 结构体类型变量概述	144
7.1.1 定义	145
7.1.2 引用	146
7.1.3 初始化	147
7.2 结构体数组的定义及引用	148
7.3 结构体类型变量的指针引用	156
7.3.1 指向结构体类型变量的使用	156
7.3.2 指向结构体数组的指针使用	157
7.4 链表的使用	160
7.4.1 单链表的创建	160
7.4.2 单链表的插入与删除	163
7.5 共用体类型变量	171
7.5.1 定义	171
7.5.2 引用	173
第 8 章 C 语言文件系统	176
8.1 缓冲文件系统	177
8.1.1 打开与关闭文件	177
8.1.2 读写文件的方式	179
8.1.3 文件的随机读写	188
8.2 非缓冲文件系统概述	191
8.3 文件处理应用实例	193
第 9 章 C 语言编程技巧	196
9.1 图形应用技巧	197

9.1.1	测试显示适配器类型.....	197
9.1.2	存取屏幕图像.....	208
9.1.3	控制屏幕显示格式.....	210
9.1.4	编译脱离 BGI 的图形软件.....	212
9.1.5	硬拷贝屏幕图形.....	212
9.1.6	任意更改 VGA 显示器显示颜色.....	215
9.1.7	用随机函数产生动画.....	217
9.1.8	用 putimage 函数完成动画显示.....	219
9.1.9	实现图像的位移变换.....	221
9.2	菜单设计技巧.....	223
9.2.1	设计下拉式菜单.....	223
9.2.2	设计选择式菜单.....	226
9.2.3	设计阴影输出窗口.....	227
9.3	音响设计技巧.....	230
9.3.1	设计音乐程序.....	230
9.3.2	自动识谱程序.....	234
9.3.3	后台演奏程序.....	237
第 10 章	C++程序基础知识.....	239
10.1	面向对象入门.....	240
10.1.1	面向对象的程序.....	240
10.1.2	C++类的结构.....	241
10.2	输入输出方式.....	243
10.3	类.....	244
10.3.1	类的定义.....	244
10.3.2	构造函数与析构函数.....	247
10.3.3	函数的重载.....	252
10.3.4	友元函数.....	253
10.4	对象指针.....	257
10.5	派生类.....	264
10.5.1	单继承的派生.....	264
10.5.2	多继承的派生.....	273
附录 A	常用字符与 ASCII 代码对照表.....	279
附录 B	C 语言函数大全.....	280
附录 C	C 常用的编辑命令.....	322
附录 D	C 编译、连接和运行时的常见错误.....	324

第 1 章 C 语言程序和算法



你 将 会 学 到



1. 程序设计语言发展史
2. C 语言的产生和发展
3. C 语言的几个特点
4. C 语言的结构
5. 算法概述

1.1 程序设计语言发展史

自1946年世界上第一台电子计算机问世以来,计算机科学及其应用的发展十分迅猛,计算机被广泛地应用于人类生产、生活的各个领域,推动了社会的进步与发展。特别是随着国际互联网(Internet)日益深入千家万户,传统的信息收集、传输及交换方式正被革命性地改变,我们已经难以摆脱对计算机的依赖,计算机已将人类带入了一个新的时代——信息时代。

新的时代对于我们的基本要求之一是:自觉地、主动地学习和掌握计算机的基本知识和基本技能,并把它作为自己应该具备的基本素质。要充分认识到,缺乏计算机知识,就是信息时代的“文盲”。

对于理工科的大学生而言,掌握一门高级语言及其基本的编程技能是必需的。大学学习,除了掌握本专业系统的基础知识外,科学精神的培养、思维方法的锻炼、严谨踏实的科研作风养成,以及分析问题、解决问题的能力训练,都是日后工作的基础。学习计算机语言,正是一种十分有益的训练方式,而语言本身又是与计算机进行交互的有力的工具。

一台计算机是由硬件系统和软件系统两大部分构成的,硬件是物质基础,而软件可以说是计算机的灵魂,没有软件,计算机是一台“裸机”,是什么也不能干的,有了软件,才能灵动起来,成为一台真正的“电脑”。所有的软件,都是用计算机语言编写的。

计算机程序设计语言的发展,经历了从机器语言、汇编语言到高级语言的历程。

1. 机器语言

电子计算机所使用的是由“0”和“1”组成的二进制数,二进制是计算机的语言的基础。计算机发明之初,人们只能降贵纡尊,用计算机的语言去命令计算机干这干那。一句话,就是写出一串串由“0”和“1”组成的指令序列交由计算机执行,这种语言,就是机器语言。

使用机器语言是十分痛苦的,特别是在程序有错需要修改时,更是如此。而且,由于每台计算机的指令系统往往各不相同,所以,在一台计算机上执行的程序,要想在另一台计算机上执行,必须另编程序,造成了重复工作。但由于使用的是针对特定型号计算机的语言,故而运算效率是所有语言中最高的。机器语言,是第一代计算机语言。

2. 汇编语言

为了减轻使用机器语言编程的痛苦,人们进行了一种有益的改进:用一些简洁的英文字母、符号串来替代一个特定的指令的二进制串,比如,用“ADD”代表加法,“MOV”代表数据传递等等,这样一来,人们很容易读懂并理解程序在干什么,纠错及维护都变得方便了,

这种程序设计语言就称为汇编语言,即第二代计算机语言。然而计算机是不认识这些符号的,这就需要一个专门的程序,专门负责将这些符号翻译成二进制数的机器语言,这种翻译程序被称为汇编程序。

汇编语言同样十分依赖于机器硬件,移植性不好,但效率仍十分高,针对计算机特定硬件而编制的汇编语言程序,能准确发挥计算机硬件的功能和特长,程序精炼而质量高,所以至今仍是一种常用而强有力的软件开发工具。

3. 高级语言

从最初与计算机交流的痛苦经历中,人们意识到,应该设计一种这样的语言,这种语言



接近于数学语言或人的自然语言，同时又不依赖于计算机硬件，编出的程序能在所有机器上通用。经过努力，1954 年，第一个完全脱离机器硬件的高级语言—FORTRAN 问世了，40 多年来，共有几百种高级语言出现，有重要意义的有几十种，影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、LISP、SNOBOL、PL/1、Pascal、C、PROLOG、Ada、C++、VC、VB、Delphi、JAVA 等。

高级语言的发展也经历了从早期语言到结构化程序设计语言，从面向过程到非过程化程序语言的过程。相应地，软件的开发也由最初的个体手工作坊式的封闭式生产，发展为产业化、流水线式的工业化生产。

60 年代中后期，软件越来越多，规模越来越大，而软件的生产基本上是人自为战，缺乏科学规范的系统规划与测试、评估标准，其恶果是大批耗费巨资建立起来的软件系统，由于含有错误而无法使用，甚至带来巨大损失，软件给人的感觉是越来越不可靠，以致几乎没有不出错的软件。这一切，极大地震动了计算机界，史称“软件危机”。人们认识到：大型程序的编制不同于写小程序，它应该是一项新的技术，应该像处理工程一样处理软件研制的全过程。程序的设计应保证正确性，也便于验证正确性。1969 年，提出了结构化程序设计方法，1970 年，第一个结构化程序设计语言—Pascal 语言出现，标志着结构化程序设计时期的开始。

80 年代初开始，在软件设计思想上，又产生了一次革命，其成果就是面向对象的程序设计。在此之前的高级语言，几乎都是面向过程的，程序的执行是流水线似的，在一个模块被执行完成前，人们不能干别的事，也无法动态地改变程序的执行方向。这和人们日常处理事物的方式是不一致的，对人而言是希望发生一件事就处理一件事，也就是说，不能面向过程，而应是面向具体的应用功能，也就是对象（object）。其方法就是软件的集成化，如同硬件的集成电路一样，生产一些通用的、封装紧密的功能模块，称之为软件集成块，它与具体应用无关，但能相互组合，完成具体的应用功能，同时又能重复使用。对使用者来说，只关心它的接口（输入量、输出量）及能实现的功能，至于如何实现的，那是它内部的事，使用者完全不用关心，C++、VB、Delphi 就是典型代表。

高级语言的下一个发展目标是面向应用，也就是说：只需要告诉程序你要干什么，程序就能自动生成算法，自动进行处理，这就是非过程化的程序语言。

1.2 C 语言的产生和发展

C 语言是 1972 年由美国的 Dennis Ritchie 设计发明的，并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。它由早期的编程语言 BCPL (Basic Combined Programming Language) 发展演变而来。在 1970 年，AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出较先进的并取名为 B 的语言，最后导致了 C 语言的问世。

随着微型计算机的日益普及，出现了许多 C 语言版本。由于没有统一的标准，使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况，美国国家标准研究所 (ANSI) 为 C 语言制定了一套 ANSI 标准，成为现行的 C 语言标准。

C 语言入门与进阶

1.3 C 语言的几个特点

1. 中级计算机语言

C 语言通常称为中级计算机语言。中级语言并没有贬义，不意味着它功能差、难以使用、或者比 BASIC、Pascal 那样的高级语言原始，也不意味着它与汇编语言相似，会给使用者带来类似的麻烦。C 语言之所以被称为中级语言，是因为它把高级语言的成分同汇编语言的功能结合起来。表 1-1 表明了 C 语言在计算机语言中所处的地位。

表 1-1 C 语言在计算机语言中的地位

高级	Ada、Modula-2、Pascal、COBOL、FORTRAN、BASIC
中级	C、FORTH、Macro-assembler
低级	Assembler

作为中级语言，C 允许对位、字节和地址这些计算机功能中的基本成分进行操作。C 语言程序非常容易移植。可移植性表示为某种计算机写的软件可以用到另一种机器上去。举例来说，如果为苹果机写的一个程序能够方便地改为可以在 IBMPC 上运行的程序，则称为是可移植的。

所有的高级语言都支持数据类型的概念。一个数据类型定义了一个变量的取值范围和可在其上操作的一组运算。常见的数据类型是整型、字符型和实数型。虽然 C 语言有五种基本数据类型，但与 Pascal 或 Ada 相比，它却不是强类型语言。C 程序允许几乎所有的类型转换。例如，字符型和整型数据能够自由地混合在大多数表达式中进行运算。这在强类型高级语言中是不允许的。

C 语言的另一个重要特点是它仅有 32 个关键字，这些关键字就是构成 C 语言的命令。和 IBMPC 的 BASIC 相比，后者包含的关键字达 159 个之多。

2. 结构化语言

虽然从严格的学术观点上看，C 语言是块结构 (block-structured) 语言，但是它还是常被称为结构化语言。这是因为它在结构上类似于 ALGOL、Pascal 和 Modula-2 (从技术上讲，块结构语言允许在过程和函数中定义过程或函数。用这种方法，全局和局部的概念可以通过“作用域”规则加以扩展，“作用域”管理变量和过程的“可见性”。因为 C 语言不允许在函数中定义函数，所以不能称之为通常意义上的块结构语言)。

结构化语言的显著特征是代码和数据的分离。这种语言能够把执行某个特殊任务的指令和数据从程序的其余部分分离出去、隐藏起来。获得隔离的一个方法是调用使用局部 (临时) 变量的子程序。通过使用局部变量，我们能够写出对程序其它部分没有副作用的子程序。这使得编写共享代码段的程序变得十分简单。如果开发了一些分离很好的函数，在引用时我们仅需要知道函数做什么，不必知道它如何做。切记：过度使用全局变量 (可以被全部程序访问的变量) 会由于意外的副作用而在程序中引入错误。

结构化语言比非结构化语言更易于程序设计，用结构化语言编写的程序的清晰性使得它们更易于维护。这已是人们普遍接受的观点了。C 语言的主要结构成分是函数 C 的独立子程序。在 C 语言中，函数是一种构件 (程序块)，是完成程序功能的基本构件。函数允许一个



程序的诸任务被分别定义和编码，使程序模块化。可以确信，一个好的函数不仅能正确工作且不会对程序的其它部分产生副作用。

3. 可移植性

也许你会问“所有的计算机语言不都是程序员使用的吗？”，回答是断然的“否”。我们考虑典型的非程序员的语言 COBOL 和 BASIC。COBOL 的设计使程序员难以改变所编写代码的可靠性，甚至不能提高代码的编写速度。

然而 COBOL 设计者的本意却是打算使非程序员能读程序（这是不大可能的事）。注意，这并不是攻击 COBOL 的优点，而是想指出，它没有被设计成为程序员的理想语言。BASIC 的主要目的是允许非专业程序员在计算机上编程解决比较简单的问题。与其形成鲜明对照的是 C 语言，由于程序生成、修改和现场测试自始至终均由真正的程序员进行，因而它实现了程序员的期望：很少限制、很少强求、块结构、独立的函数以及紧凑的关键字集合。用 C 语言编程，程序员可以获得高效机器代码，其效率几乎接近汇编语言代码。

C 语言被程序员广泛使用的另一个原因是可以用它代替汇编语言。汇编语言使用的汇编指令，是能够在计算机上直接执行的二进制机器码的符号表示。汇编语言的每个操作都对应为计算机执行的单一指令。虽然汇编语言给予程序员达到最大灵活性和最高效率的潜力，但开发和调试汇编语言程序的困难是难以忍受的。非结构性使得汇编语言程序难于阅读、改进和维护。也许更重要的是，汇编语言程序不能在使用不同 CPU 的机器间移植。

最初，C 语言被用于系统程序设计。一个“系统程序”是一大类程序的一部分，这一大类构成了计算机操作系统及实用程序。通常被称为系统程序的有：

- 操作系统。
- 翻译程序。
- 编辑程序。
- 汇编程序。
- 编译程序。
- 数据库管理程序。

随着 C 语言的普及，加之其可移植性和高效率，许多程序员用它设计各类程序。几乎所有的计算机上都有 C 语言编译程序，这使我们可以很少改动甚至不加改动地将为一种机器写的 C 语言源程序在另一种机器上编译执行。可移植性节省了时间和财力。

C 语言不仅在速度和结构上有它的优势，而且每个 C 语言系统都提供了专门的函数库，程序员可以根据不同需要对其进行剪裁，以适应各种程序的设计。由于它允许（更准确地说是鼓励）分别编译，所以 C 语言可使程序员方便地管理大型项目，最大限度地减少重复劳动。

1.4 C 语言的结构

1.4.1 入门实例

任何一种程序设计语言都具有特定的语法规则和规定的表达方法。一个程序只有严格按照语言规定的语法和表达方式编写，才能保证编写的程序在计算机中能正确地执行，同时也