天信文化
Tian Xin WenHua

国家级示范性高等院校精品规划教材

C语言基础教程

C YUYAN JICHU JIAOCHENG

王兴波 刘 波/主 编
周 燕 李 娅 于昕梅/副主编

天津大学出版社
TIANJIN UNIVERSITY PRESS

C 语言基础教程

王兴波　刘　波　主　编

周　燕　李　娅　于昕梅　副主编

天津大学出版社
TIANJIN UNIVERSITY PRESS

内容提要

全书采用以知识点形成知识轮廓再形成结构性知识体系的方式编写。采用了大量的卡通插图对每个章节的重点、难点进行提示，对一些基本概念和技巧也做了大量注释；整个教材充满童趣、通俗易懂。书中内容还结合了国家计算机等级考试二级（C 语言）的具体要求。

全书共 12 章，第 1 章计算机程序设计的含义，第 2 章 C 语言程序的基本结构，第 3 章 C 语言的基本数据类型与简单 I/O，第 4 章 C 语言的语句特征，第 5 章函数，第 6 章变量的存储属性，第 7 章编译预处理与工程，第 8 章 C 语言的数组及其应用，第 9 章指针变量及其属性，第 10 章结构体、共用体数据及其应用，第 11 章 C 语言的文件 I/O 及其应用，第 12 章 C 语言的位运算与低级操作。

本书可以作为本科、高职、中专工科用计算机高级语言程序设计的教材，也可作为一般工程技术人员参考用书。

图书在版编目（CIP）数据

C 语言基础教程 / 王兴波，刘波主编. —天津：天津大学出版社，2010.11

国家级示范性高等院校精品规划教材

ISBN 978-7-5618-3777-1

Ⅰ. ①C… Ⅱ. ①王… ②刘… Ⅲ. ①C 语言—程序设计—高等学校—教材 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2010）第 225193 号

出版发行	天津大学出版社
出 版 人	杨欢
地　　址	天津市卫津路 92 号天津大学内（邮编：300072）
电　　话	发行部：022-27403647　邮购部：022-27402742
网　　址	www . tjup . com
印　　刷	昌黎太阳红彩色印刷有限责任公司
经　　销	全国各地新华书店
开　　本	185mm×260mm
印　　张	18
字　　数	450 千
版　　次	2011 年 1 月第 1 版
印　　次	2011 年 1 月第 1 次
定　　价	32.60 元

序　言

刚上研究生时，我学习的是计算机高级语言 Turbo Pascal4.0。因为在此以前没有接触过计算机，我学得如痴如醉，非常认真。这时，一个同学对我讲："你学 C 吧，C 好！"当时我对他的话不以为然，因为我觉得计算机语言只是帮助人们解决问题的工具之一，只要掌握一种就可以了。及至做硕士论文课题时，我发现自己几乎无法集中时间研究课题中的问题，而不得不将大量的时间投入到程序设计工作中。这时我认识到：只有采用一种灵活的程序设计语言才能将我解脱出来。我想起那个同学的话，开始学习 C 语言程序设计。后来我用 C 语言很成功地编写了硕士课题中的软件。

研究生毕业后，单位安排我给本科生讲授 C 语言程序设计，给了我一个系统了解 C 语言的机会。开始时，我选用了谭浩强先生编著的《C 程序设计》作为教材。教学过程中，我还编写了一个用计算机辅助 C 语言教学的软件。随着自己对 C 语言系统有了更加深入的了解，我萌发了自己写一本 C 语言教材的念头。我的第一本教材《C 语言电脑投影教学实践》由国防科技大学出版社出版，主要是针对当时方兴未艾的计算机辅助教学实践活动编写的。遗憾的是，Windows 操作系统很快将我这部以 DOS 为主要背景的教材放置在垃圾桶之中了。

后来的科研工作使我从 C 走向了 C++。因为主要集中于科研工作和研究生教学工作，积累了的大量 C++开发经验，因此有了再出一本书的愿望。2004 年，我与杨利教授和聂勇军博士合作，执笔编写了一本 C/C++教材，由国防科技大学出版社出版。2008 年，我从部队转业到湖南农业大学，担任计算机系主任，重新开始了本科教学工作。由于教学对象的改变，我对 C/C++教材的需求有了新的认识——总期望有一种教材能把 C/C++语言的教学变成一种学生轻松学与教师轻松教的过程。基于这样的想法，我开始改写以前的教材。

本次改写的教材还是称为基础教程。这是因为我希望它能够成为一本经典教材。书中采用了很多卡通画以强调重点、难点，对一些基本概念和技巧也重新做了注释，更加通俗易懂。书中注重章节内容简繁的安排，适合课堂讲授，也适合自学。

与我的学生时代不同的是，C 语言已经成为计算机程序设计的普及型语言。现在几乎所有工科大学都开设了 C/C++语言课程，这是一件十分令人高兴的事情。曾有言："英语、数学、计算机是未来科学技术的根本！"编者深信此言极其正确！作为在信息领域工作的科技工作人员，我们向读者奉献此书，希望能够对培养未来科学技术工作者有所贡献。

书中的诸多卡通形象均系在互联网上下载，在此对这些作品的原创者表示诚挚的感谢。

王兴波

2010 年 5 月于佛山

目　录

第1章 计算机程序设计的含义

“程序设计”这一术语已经成为很多科技工作者的口头禅。那么，程序设计到底包含着什么样的意思，关联着哪些事情呢？本章通过对计算机及其工作方式的回顾，说明程序设计的含义以及我们学习程序设计时需要做的事情。本章需要掌握的内容有：

（1）计算机系统的组成与冯·诺依曼计算机系统的工作原理；

（2）软件是什么，编写计算机软件需要哪些支撑的东西；

（3）计算机语言及其发展历史，C 语言在计算机语言中的地位；

（4）程序设计的基本过程。

1.1 计算机系统及其工作原理

1.1.1 计算机系统组成简述

在学过计算机基础课程之后，我们已经知道，计算机系统由硬件系统和软件系统两大部分组成，如图 1-1 所示。

图 1-1 计算机系统的组成

1．计算机硬件系统

硬件系统是计算机系统的物理装置，即由电子线路、元器件和机械部件等构成的具体装置。通俗地说，硬件系统就是计算机上我们看得见、摸得着的那些东西的总和：显示器、鼠标、键盘、主板、内存条、显卡、硬盘、光驱等都属于硬件系统中的部件。计算机按照运算与处理的能力，可分为巨型机、大型机、中型机、小型机和微型机。微型机价格便宜，能够满足人们日常生活中绝大部分的需要。例如，人们日常生活中常见的计算机如家庭电脑、办公用电脑、手提电脑都属于微型机。其他类型的计算机通常用于专业科学研究工作。本课程的学习只用到微型机，因此，本书后文所说的计算机，除特别说明的以外，均指微型机。

人们常常把计算机的硬件系统按照其功能划分为输入设备、输出设备、存储器、运算器和控制器五大部分，各部分之间的关系如图 1-2 所示。

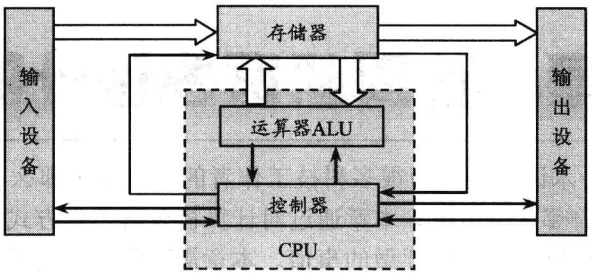


图 1-2　计算机硬件系统各部分之间的关系示意图

（1）输入设备：负责计算机处理数据的输入。例如，键盘、鼠标是常见的输入设备，硬盘是既可输入又可输出的设备，只读光驱（CDR）是输入设备。

（2）输出设备：负责计算机处理数据的输出。例如，显示器是常见的输出设备，硬盘是既可输入又可输出的设备，可读写光驱（CDRW）也是既可输入又可输出的设备。

（3）存储器：负责存储计算机处理的各种数据。存储器可分为内存储器（简称内存）与外存储器（简称外存）两个部分。而外存储器属于输入/输出设备，因此，这里的存储器仅指内存。处理器处理的所有数据都离不开内存。

（4）运算器（ALU）：是对数据信息进行加工处理的部件，其主要功能是执行算术运算、逻辑运算和信息传送。

（5）控制器：是统一指挥和有效控制计算机各部件协调工作的部件，它从存储器中逐条地取出、分析指令，并向各部件发出相应的控制信号，一步步地执行指令所规定的操作。

运算器与控制器合称为处理器，如图 1-3 所示。处理器是计算机的核心部分，主要进行计算与控制。处理器经过几代的发展，速度已经变得非常快了。

大型计算机都拥有多个处理器。早期的个人计算机（简称 PC 或 PC 机）只有一个处理器，因此人们称之为中央处理器，英文缩写为 CPU。第四代以后的个人计算机也拥有多个处理器，人们已经无法确定哪一个是“中央”的，因而统称为微处理器。

处理器总是跟内存打交道。处理器从内存取出数据进行运算，运算的结果又放回到内存，如图 1-4 所示。

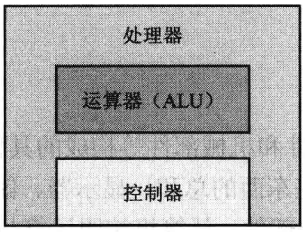


图 1-3　运算器与控制器组成处理器

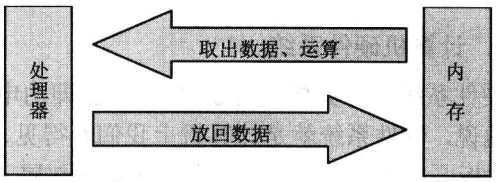


图 1-4　处理器与内存的数据流向关系示意图

硬盘、软盘、光盘、U 盘等都属于外存储器，它们只能通过内存才能与处理器建立联系。例如，计算机处理磁盘数据时，先将磁盘数据读到内存，再从内存送到处理器处理。对于学习 C 语言和低级语言的学生来说，这一点尤其要掌握。

2．计算机软件系统

计算机之所以能够处理我们日常生活中的各种数据、从事科学计算，是因为它里面装有

软件系统（software system）。没有软件系统的计算机称为裸机，它不能做任何事情。因此，计算机软件（computer software）是使计算机工作的指令系统的集合。计算机是按照指令一步一步执行计算的。软件的标准定义是：计算机系统中的程序及其文档。但是在没有学习程序设计以前，我们还难以准确地把握这个定义。学完本课程以后再回过头来理解这个定义就很容易了。因此，程序设计是跟计算机软件紧密相关的。事实上，程序设计这门课就是告诉我们开发出计算机软件的基本知识。

按照软件系统的功能，可以将软件划分为系统软件和应用软件两大类：系统软件是指控制和协调计算机及其外部设备，支持应用软件的开发和运行的软件；应用软件是专门解决某个或某些应用领域中的具体任务的软件。还可以对软件进行更细致的分类，这里不再赘述。

1.1.2　计算机的工作原理

回顾一下我们在计算机上观赏一部电影的情形。当我们用一个播放软件播放一部我们喜爱的电影时，计算机能够一幕接一幕地自动播放全部的内容。计算机为什么自动工作呢？这是因为现在的计算机基本都是按照冯·诺依曼（Von Neumann）设计原则而设计的。

冯·诺依曼是一个数学家。他对人类的最大贡献就是规划了电子计算机的一种系统结构。在分析了世界上第一台计算机 ENIAC 的工作原理的基础上，冯·诺依曼提出了计算机工作的基本原理，这就是现在被称为“程序存储+程序控制”的计算机工作原理（简称“程序存储原理”），其基本要点是：数字计算机的数制采用二进制，计算机应该按照程序顺序执行。具体解释如下。

（1）程序存储。程序存储的原理如图 1-5 所示。程序是由一条一条的指令组成的，计算机在运行时按照指令的顺序依次执行运算。运算前必须事先将程序存入存储器中，再由控制器自动读取并执行。程序与数据都采用二进制数、以字节为单位进行存储。

（2）程序控制。程序控制的原理如图 1-6 所示。计算机在工作时，控制器按照程序制定的顺序到存放程序代码的内存区域去取指令代码，在 CPU 中完成对代码的分析，然后依据分析的结果，适时地向各个部件发出完成该指令功能的控制信号，实现指令功能。

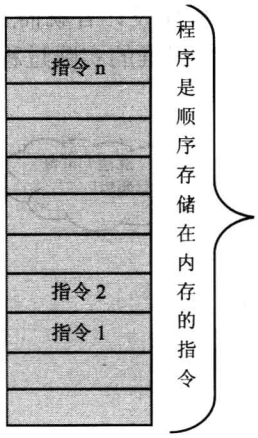


图 1-5　程序存储原理——程序在内存中存储的示意图

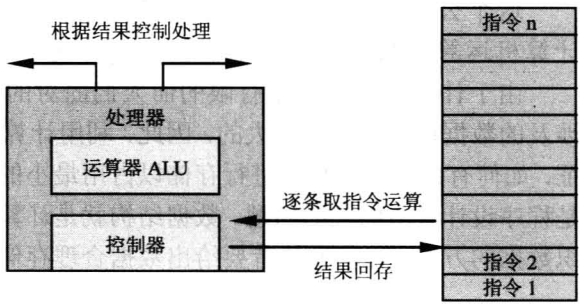


图 1-6　程序控制原理示意图

上述原理也称为计算机的冯·诺依曼体系结构。根据这个原理，电子计算机必须具备以

下功能。

（1）把需要的程序和数据送至计算机中。

（2）必须具有长期记忆程序、数据、中间结果及最终运算结果的能力。

（3）能够完成各种算术运算、逻辑运算和数据传送等数据加工处理的能力。

（4）能够根据需要控制程序走向，并能根据指令控制机器的各部件协调操作。

（5）能够按照要求将处理结果输出给用户。

图 1-7　计算机之父——冯·诺依曼

为了实现上述的功能，计算机必须具备五大基本组成部件，即运算器、控制器、存储器、输入设备和输出设备。因此人们所说的“计算机五大组成部分”实际上是冯·诺依曼体系结构的基本组成。包括 ENIAC 在内，世界上几乎所有的计算机都采用冯·诺依曼体系结构[①]。虽然计算机技术发展很快，但“存储程序原理”至今仍然是计算机内在的重要工作原理，它仍然是我们理解计算机系统功能与特征的基础。基于这样的缘故，冯·诺依曼（见图 1-7）被称为“计算机之父”。

1.2　程序设计概述

从上节对计算机工作原理的介绍，我们知道，计算机之所以能够自动工作，是因为有程序在运行和控制它。程序是由人们经过程序设计得到的。那么程序设计到底是怎样一回事呢？本节将讨论与此问题相关的内容。

1.2.1　程序设计的含义

程序设计，顾名思义，就是设计计算机工作所需要的程序。由于计算机是按照指令来工作的，因此程序设计的实质是根据计算机的工作原理，规划、编写各种指令。计算机的指令系统，通常由有限个基本指令集组成；而人们在日常生活、科学技术研究中会遇到无穷无尽个形形色色的问题。要想让计算机帮助人们解决这些问题，程序设计者就需要利用这有限个指令，去构思、设计无穷无尽个方案、方法，这是一个创造性思维的实践过程。有一本很著名的计算机书，书名为：《程序=数据结构+算法》。

这个公式高度概括了程序设计的内涵。我们知道，计算机运算的数据和指令都需要存储在内存中。

由于计算机的内存是有限的而人们面对的某些问题所涉及的数据可能是非常庞大的，因此，利用计算机的存储特征，如何有效地组织数据进行存储以占用最小的内存，永远是程序设计者所面临的问题。数据结构就是计算机存储、组织数据的方式。程序设计者要给出数据合理存储的方式，就

① 世界上还有另外的计算机体系结构。例如，哈弗结构。在哈弗结构里，程序和数据分开存储。程序和数据都采用二进制存储是冯·诺依曼体系结构的基本特征。

要研究数据结构。算法是我们根据计算机工作方式所给出的解决问题的方法。需要注意的是，算法跟我们日常生活中的方法是有差别的。

例如，对于计算多项式 $y=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1+a_0$ 而言，不了解计算机算法的人，一般采用的方法是：将 x 自乘 n 次再乘以 a_n，加上 x 自乘 n–1 次乘以 a_{n-1}，以此类推。这样需要 $1+2+\cdots+(n)+(n+1)=(n+2)(n+1)/2$ 次乘法和 n+1 次加法。但若利用适当的算法，这个计算只需要 n+1 次加法和 n+1 次乘法。因此，程序设计者还需要了解算法——结合计算机工作原理与实际问题的需求，寻找快捷的计算方法。

1.2.2　程序设计语言

程序设计语言是用于编写计算机程序的语言。就像我们人类的语言是由符号和语法规则组成一样，计算机语言也是由特定的符号和语法规则组成。

程序设计语言大体上分为低级语言和高级语言两种，如图 1-8 所示。

图 1-8　程序设计语言的分类

低级语言包括机器语言和汇编语言。机器语言是计算机能够直接识别和使用的语言，用二进制数字表示。机器语言直接跟机器指令相关。不同机器的指令，其机器语言的格式是不同的。因此机器语言也称为面向机器的语言。对于人而言，机器语言烦琐、复杂、难以记忆和修改。因此人们就采用一定的助记符（帮助记忆的符号）来表示机器语言。汇编语言就是机器语言部分符号化的结果。

低级语言与特定的机器有关、功效高，但使用复杂、烦琐、费时、容易出错。低级语言适合于编写硬件开发的程序。例如操作系统软件中很多跟硬件打交道的部分就是用低级语言编写开发的。

为避免低级语言的不足，人们又开发出了高级语言。高级语言是人易于理解的计算机语言，比低级语言更符合人的认知模式，在一定程度上与具体机器无关，且易学、易用、易维护。高级语言适合于开发人性化的程序。高级语言有很多种。

高级语言的总体发展趋势是结构化、模块化、简明化、形式化、并行化和可视化。结构化、模块化是 20 世纪 60 年代中期提出的一种程序设计原则，它要求按照功能将一个大程序分解成一些独立功能的子程序（模块）。结构化、模块化的目的是为了降低程序的复杂度，使程序设计、调试和维护等变得简单。程序的形式化、并行化和可视化已经超出本课程的范

围，读者在今后的学习中会遇到，这里不予讲述。

高级语言适合于人编写计算机程序，但是任何高级语言最终都要转变成机器语言才能被机器接受。

把高级语言转变成机器语言的工具，按照其工作模式，可分为编译工具（编译器）和解释工具（解释器）两种。编译器一次将高级语言的全部都转变成一种称为目标代码的机器语言；计算机在运行程序时，一次全部将目标码加载到内存后再逐条指令执行。解释器将高级语言程序一次（或多次）加载到内存，然后逐条解释成机器语言来执行。

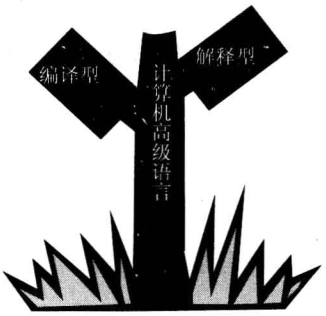
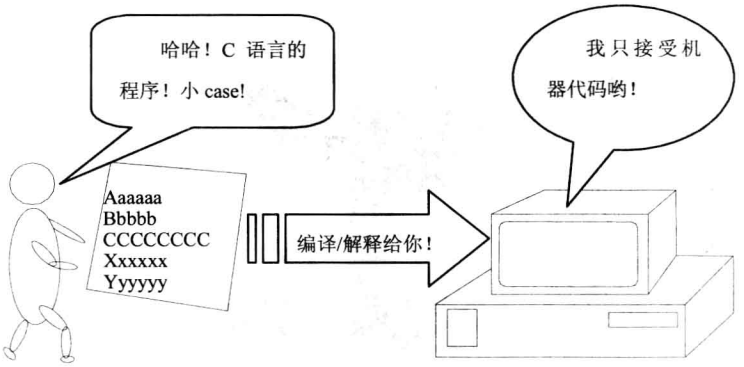


图 1-9　高级语言的类型

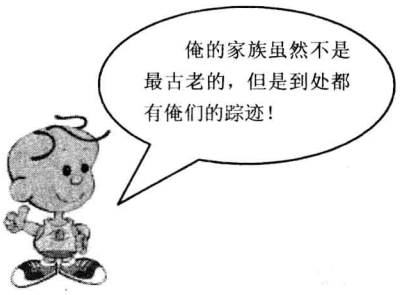
高级语言按照其生成机器代码的方式，可分为编译型高级语言和解释型高级语言，如图 1-9 所示。由于事先全部转变成了机器代码，编译型高级语言编写的程序比解释型的运行得要快。



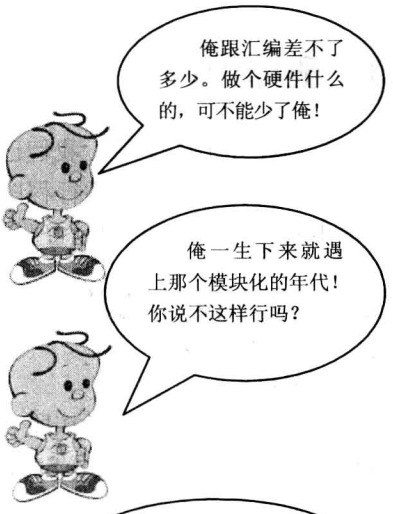
1.2.3　C 语言的特点

学习 C 语言，首先应该了解 C 语言。C 语言是一种高级语言，具有以下特点。

（1）不是历史最悠久的语言，却是生命力最旺盛的高级语言之一。历史最悠久的语言是机器语言，其后是汇编语言；C 语言是在其后产生的。事实上，C 语言是从 B 语言发展起来的。B 语言是贝尔实验室开发的一种通用的程序设计语言，它是于 1969 年前后由美国贝尔实验室的电脑科学家肯·汤普森（Ken Thompson）在丹尼斯·利奇（Dennis Ritchie）的支持下设计出来的。后来，丹尼斯·利奇以 B 语言为基础开发出 C 语言。自从被 C 语言取代之后，B 语言几乎已遭弃置，而 C 语言则成为目前世界上最常用的程序设计语言之一。



（2）是高级语言却具有汇编语言的功能。C 语言创造者创造 C 语言的初衷是希望 C 语言能够取代汇编语言来编写操作系统。因此，C 语言具有开发硬件的功能。事实上，今天的许多与硬件相关联的程序，都是用 C 语言写的。



（3）符合结构化、模块化程序设计规范。结构化、模块化是 20 世纪 60 年代中期提出的一种程序设计规则，它要求程序按照模块进行结构设计，每个模块具有相对独立性。按照这种要求设计的程序可读性好、移植性好。由于 C 语言是在 60 年代中后期发展起来的，因此，它一问世就必然满足这种规则。

（4）基本数据简单，数据处理内容丰富、范围广泛。一种语言的数据处理能力确定了这种语言的应用范围。C 语言具有很强的数据处理能力。尽管 C 语言直接处理的基本数据很简单（只有整数型、实数型、字符型、指针型和 void 类型），但是它可以通过一些规则将这些数据扩充成更加丰富的类型，这与一些其他语言不一样。有些高级语言如 PASCAL、BASIC 等的基本数据类型多，但扩充处理的灵活性不如 C 语言。

（5）运算丰富，支持简单明了的运算符。与其他语言相比，C 语言的运算符和表达式更加简单明了；C 语言支持的运算有 34 种之多。

（6）功能强大，能进行多种低级操作。C 语言是除了汇编语言以外功能最强大的语言。它能实现许多低级操作，如对磁盘、打印机、I/O 端口、鼠标等的直接操作。利用 C 语言能编写控制硬件的程序对一些硬件进行操作。例如，用 C 语言编制一些对磁盘操作，如创建磁盘目录、删除磁盘文件、修改文件属性、读写磁盘物理扇区等的程序是非常方便的。

（7）目标码质量高，可移植性好。C 语言的编译程序是高效编译系统，生成的目标代码质量高。经过实验统计，C 语言编译的目标码只比汇编目标码的效率略低而高于其他所有高级语言的目标码。C 语言生成的目标码可以与其他语言链接生成可执行文件，其移植性能极好。

（8）拥有丰富多彩的系统工具和库函数。所有的 C 语言系统都有一个共通的特点：提供大量的库函数。现在的 C 系统拥有数百个库函数。利用这些库函数，程序设计者能够开发丰富多彩的软件。

1.3 程序设计

通过前面几节的学习，我们已经了解了程序设计的含义以及高级语言、C 语言的一些大概。本小节主要介绍如何进行程序设计。先介绍程序设计的基本术语，再介绍程序设计的三个基本步骤，然后介绍从事程序设计必备的工具。

1.3.1 程序设计的基本术语

1．源程序

源程序就是程序设计者按照一种语言的规范编写的原始程序代码。源程序是程序最基本的数据。所有的源程序都是 ASCII 码文本，是珍贵的数据资料。源程序又叫源代码或源文件，通常是我们从键盘一个字符一个字符地输入的。

2．编译/解释、编译型/解释型语言系统

前面已经介绍，源程序需要经过编译/解释才能变成机器代码，编译/解释是通过编译器/解释器实现的，而编译器/解释器是由高级语言系统提供的。通过编译的方法处理源程序的语言系统称为编译型语言系统，通过解释的方法处理源程序的语言系统称为解释型语言系统。目前的高级语言系统大多数为编译型语言系统。

3．编译程序（器）、目标程序

所有的编译型程序设计系统都提供一个编译程序。这个编译程序就是前面所说的编译器。编译过程是靠编译程序实现的。当用户把源程序设计好以后，需要利用编译程序对源程序进行编译。在编译的过程中，编译程序对源程序进行扫描，首先检查分析源程序中的错误（语法错误、拼写错误）并把这些错误告诉用户。当用户清除所有错误后，编译程序就把源文件编译成一个机器可以理解的二进制程序（称为目标程序或目标文件，后缀为.OBJ）。

源程序、编译程序、编译过程及目标文件的关系如图 1-10 所示。

4．库/开发包

图 1-10　程序编译关系

自从有了程序设计以来，我们的历代前辈已经积累了许多程序。有些程序是可以为我们所借鉴和利用的。以数学计算为例，先人们已经把三角函数、对数函数、指数函数等常用的数学计算的程序写好了，我们不必再去写这些程序。如果需要，在编写程序时直接借用它们就可以。为了方便，先辈们把那些可以留给后辈们直接使用的程序分类后集中在一起编译好后形成开发库（Library）或开发包（Package），库和开发包是程序设计不可缺少的工具。我们一般通过文档来了解库和包的功

能与使用方法。一般的高级语言系统都带有一些基本的库，我们可以直接使用。

5．链接程序（器）与链接

能够在机器上运行的程序称为可执行程序（文件）。对编译型系统而言，链接是生成可执行文件的最终环节。我们在编写程序的过程中不可避免地要用到一些库。如前所述，这些库都是前人编译好后分类存放在外存储器上的。链接程序就是帮助我们把我们程序中需要的库与我们自己的目标代码整合起来。通俗地讲，链接程序将编译生成的目标程序链接成可执行文件。链接程序、链接过程及可执行文件的关系如图 1-11 所示。

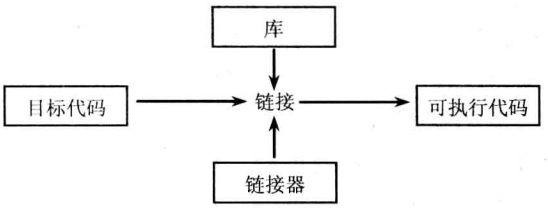


图 1-11　程序链接关系

6．解释程序与解释

解释程序是解释型语言系统提供的一种翻译程序。解释程序把源程序解释成机器能理解的形式。与编译程序不同的是，解释程序并不生成源程序的目标程序，而是生成一种另外的可执行的中间代码。因此，解释程序每解释一条指令就立即执行它。

1.3.2　程序设计的四部曲

程序设计是我们针对某一（类）具体问题，按照特定计算机语言的语法规则将解决方法（算法）用该语言表达出来并生成机器语言，以实现计算机对问题的求解。任何一个程序设计都包括四个阶段：设计阶段、编辑阶段、调试阶段和编写文档阶段。

1.3.2.1　设计阶段

程序设计是要解决问题的。显然，不同的问题乃至同一问题的不同需求，都决定了解决问题的方法的差异。因此，程序设计的第一步就是要了解问题及其需求，规划好解决问题的方法。这就是“设计”的意义所在。程序员要设计一个好的程序，需要大量的学习和实践，除了学习书本知识以外，还要学习前人的经验，并在实践中锻炼、发挥自己。

1.3.2.2　编辑阶段

规划好算法之后，就要进行编码。这是利用编辑器编写源代码（源程序、源文件）的过程。编码需要按照语言的语法规则进行。每种语言都有自己编码的规则，我们学习计算机语言，本质上就是学习编码规则。一般的语言都包含以下几大部分：

（1）数据类型及其派生规则；

（2）语句规则；

（3）流程控制规则；

（4）子程序及其调用规则；

（5）文件的输入输出规则。

因此，学习计算机语言一定要掌握好上述规则。

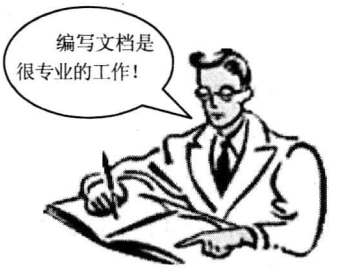
1.3.2.3 调试阶段

完成了源程序的编码后，就要调试我们编写的程序了。所谓调试，就是通过反复运行我们编写的程序，找出程序中存在的问题。一般说来，我们刚开始编写的源程序里面，总是会存在这样或者那样的错误：有些是语法错误，有些是逻辑关系上的错误，有些是算法上的不足。任何一个错误都将使程序无法产生正确的结果。需要说明的是，对于编译型程序设计语言，在调试前还需要进行编译和链接生成可执行程序后才能进入调试阶段。源程序中的一些明显的语法错误可以在编译过程中被发现，但是逻辑错误和算法上的不足，往往是编译过程无法发现的，需要通过实验数据调试才可能找到。

程序的调试过程是一个非常辛苦和痛苦的过程，程序员工作时间的 90%都是在调试程序中度过的。对于一个程序员而言，通宵达旦地调试一个小毛病，不是什么骇人听闻的事，通常都是事实。因此，学习计算机语言，需要良好的钻研精神。

1.3.2.4 编写文档阶段

我们编写的程序通常是要发布的——把程序传播出去以便他人利用或者借鉴。因此，需要对我们的程序进行详细的说明。从技术层面介绍和说明程序的文章称为程序的文档。一个完整的文档包含以下几个方面的内容：



（1）程序的作用；

（2）程序设计的思想和实现的方法；

（3）程序中应用到的技术点及其说明；

（4）程序产生的结果及其展示。

回顾软件的定义，我们可以看出，文档是软件不可分割的一部分。因此，程序员需要学会编写文档。编写文档是软件工程领域一个很专业的工作，其内容已经超出本书的范围，这里不予讲述。

1.3.3 程序设计必备的工具——程序设计三剑客

现在，我们已经知道了程序设计的含义及其基本过程，按道理就可以开始学习并进行程序设计了。但是，编者在这里还要说一些次要的东西，那就是程序设计的必备工具：合适的语言、合适的开发环境及合适的实验环境。

1.3.3.1 合适的语言

前面已经介绍，语言是生成程序的基本工具。进行程序设计，必须选择一个适合学习者自己的语言。目前，在各种各样的高级语言中，有些适合于初学者学习，有些适合于熟练者