



普通高等教育“十一五”国家级规划教材
高等院校计算机应用技术规划教材

C语言程序设计 (第三版)

C

恰汗·合孜尔 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



普通高等教育“十一五”国家级规划教材
高等院校计算机应用技术规划教材

C 语言程序设计

(第三版)

			恰汗·合孜尔	主 编
茹 蓓	邓沌华	安 杰	刘立君	副主编
王成虎	黄 华	匡代洪	宋艳萍	
	陈 昊	张 宇	刘莹昕	参 编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书从程序设计的实际能力培养出发,由浅入深、深入浅出,将理论与实践有机结合,融知识传授和能力培养为一体。本书内容丰富、注重实践;突出重点、分散难点;例题广泛、结合实际;图文并茂、文字流畅。本书的宗旨在于进一步巩固对基本知识的理解和掌握,提高学生的逻辑分析、抽象思维和程序设计能力,培养学生良好的程序设计风格,进而具备编写中、大型程序的能力。

本书中的程序按照模块化程序设计思想进行编写的同时,每一个程序都遵循软件工程方法学的编程风格,即采用缩进格式,程序中附有注释,以便于对程序的分析、理解和自学。

本书适合作为高等学校各专业“C 语言程序设计”课程的教材,也可供 C 语言自学者或参加各种 C 语言考试的读者及各类工程技术人员学习使用。

图书在版编目(CIP)数据

C 语言程序设计/恰汗·合孜尔主编.—3 版.

—北京:中国铁道出版社,2010

(高等院校计算机应用技术规划教材)

ISBN 978-7-113-10826-7

I. ①C… II. ①恰… 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(2009)第 223994 号

书 名: C 语言程序设计(第三版)

作 者: 恰汗·合孜尔 主编

策划编辑: 秦绪好 辛 杰

责任编辑: 黄园园

编辑部电话: (010) 63560056

封面设计: 付 巍

封面制作: 李 路

责任印制: 李 佳

出版发行: 中国铁道出版社(北京市宣武区右安门西街 8 号 邮政编码: 100054)

印 刷: 三河兴达印务有限公司

版 次: 2005 年 8 月第 1 版 2008 年 12 月第 2 版 2010 年 3 月第 3 版
2010 年 3 月第 9 次印刷

开 本: 787mm×1092mm 1/16 印张: 19.5 字数: 479 千

印 数: 10 000 册

书 号: ISBN 978-7-113-10826-7

定 价: 29.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签,无标签者不得销售

凡购买铁道版图书,如有印制质量问题,请与本社计算机图书批销部联系调换。

根据教材第二版的使用情况,这次对该套教材的内容进行了大幅度的调整和增删。第三版与第二版相比,一是删除难度较高的、对数学知识要求较高的例题,遴选了一些易理解而又典型的例题;二是在章节的编排上做了适当调整;三是增加了算法部分的内容;四是本书采用 Visual C++ 6.0 作为开发环境,所有例题均指定扩展名为.c,在 Visual C++ 6.0 环境下上机调试并通过,便于教师在上课时演示。本书的具体特点如下:

(1) C 语言的概念比较复杂,规则较多,使用灵活,不少初学者感到困难。针对 C 语言的特殊性,本书采用突出重点、分散难点的方式编写;内容组织上层次分明,由浅入深;同时从培养程序设计的实际能力出发,将理论与实践有机结合,融知识传授和能力培养为一体。

(2) 书中实例按照模块化程序设计思想进行编程的同时,每一个程序都遵循软件工程方法的编程风格。即对大部分例题用流程图方式和 N-S 图方式同时给予描述;程序采用缩进格式,程序中附有注释,便于对程序的分析、理解和自学。另外,培养学生良好的程序设计风格,进而具有编写中、大型程序的能力。

(3) 每章都有引言和本章小结,便于学生抓住要领。每章的引言中,简明扼要地介绍为什么引进本章内容及目的。本章小结中,有学习的要点和难点,便于学生抓住重点。同时,每一章末附有精心挑选和设计的多种类型的习题,有助于读者通过练习,进一步理解和巩固各章节的内容。

本书共分 10 章。第 1 章概要介绍了 C 语言以及在 Visual C++ 6.0 环境下如何运行 C 语言程序。第 2 章介绍了 C 语言的语法基础以及顺序结构程序设计的基本方法。第 3 章介绍了选择结构程序设计的基本语句以及使用选择结构编写程序。第 4 章介绍了循环结构程序设计的基本语句以及使用循环结构编写程序。第 5 章介绍了数组的概念以及使用数组编写程序。第 6 章介绍了模块化程序设计方法、函数的概念、变量的作用域和存储类别以及编译预处理命令。第 7 章介绍了指针、指针变量以及使用指针编写程序。第 8 章介绍了结构体、共用体、枚举类型以及链表及其应用。第 9 章介绍了 C 文件的基础知识以及最基本的文件操作。第 10 章介绍了各种位运算及其运算规则。

本书讲授学时数为 60~116 学时,其中包括习题、上机实验和课程设计学时数,上机实验和课程设计部分的内容可参照配套的《C 语言程序设计习题集与上机指导(第三版)》进行,各高校可根据教学课时数来确定教学内容以及相应的实验内容和课程设计内容。本书配有电子教案,以方便教学和读者自学,请到网站 <http://edu.tqbooks.net> 下载。

本书适合作为高等学校各专业“C 语言程序设计”课程的教材,也可供 C 语言自学者或参加各种 C 语言考试的读者及各类工程技术人员学习使用。

本书由恰汗·合孜尔主编,茹蓓、邓沌华、安杰、刘立君任副主编。其中,第 2、5、7、8 章以及附录由恰汗·合孜尔编写,第 1 章由邓沌华、刘立君编写,第 6 章由茹蓓编写,第 9 章由华北煤炭医学院的安杰编写,第 10 章由王成虎编写。另外,黄华、匡代洪、宋艳萍、叶尔兰、马光春、张宇、陈大春、古丽孜拉、陈昊、刘莹昕、阿尔达克等参加了编写。

在本书的编写过程中，编者广泛参阅、借鉴和吸收了国内外 C 语言程序设计方面的相关教材和资料，并吸取了这些书的优点，在此谨向这些教材和资料的作者致以诚挚的感谢。

由于作者水平有限，书中难免存在疏漏与不妥之处，恳请同行和广大读者批评指正。

编者

2010 年 1 月

C 语言是目前国内外广泛使用的一种程序设计语言,是高等院校计算机及相关专业重要的专业基础课。C 语言以其功能丰富、使用灵活、可移植性好,既可以用来编写系统程序,又可以用来编写应用程序等优点,越来越受到人们的欢迎。

根据教材第一版的使用情况及任课教师在多年教学工作中的经验和体会,对该教材进行了修订和补充。第二版与第一版相比,不仅从组织、结构、叙述、内容的筛选以及章节的编排上都有了很大变动,还新增了“算法描述”这一部分的内容。另外,为了便于学生理解,大部分例题同时用流程图方式和 N-S 图方式给予了描述,对函数和语句增加了注释,具体特点如下:

① 系统性、实用性强,内容组织上层次分明、结构清晰,全面讲授 C 语言程序设计的基本思想、方法和解决实际问题的技巧。

② C 语言的概念比较复杂,规则较多,使用灵活,容易出错,不少初学者感到困难。本书从程序设计的实际能力培养出发,由浅入深、深入浅出,将理论与实践有机结合,融知识传授和能力培养于一体。

③ 内容丰富,注重实践;突出重点,分散难点;图文并茂,文字流畅。本书的宗旨在于进一步巩固学生对基本知识的理解和掌握,提高学生的逻辑分析、抽象思维和程序设计能力,培养学生养成良好的程序设计风格,进而具备编写大型程序的能力。

④ 基于软件工程方法学的理论进行程序设计,即程序设计上完全按照模块化程序设计思想进行编程。每一个程序都遵循软件工程方法学的编程风格,即采用缩进格式;程序中附有注释,便于读者对程序的分析、理解和自学。

⑤ 对所介绍的内容都举有典型的实例,书中的每个程序都在 Turbo C 2.0 环境下经过上机调试,便于教师在上课时演示。同时,每章后都设有精心挑选的多种类型的习题,以帮助读者通过练习进一步理解和巩固所学的内容。

⑥ 每章均有引言和本章小结,便于学生抓住要领。引言简明扼要地介绍了为什么引进本章以及学习本章的目的。本章小结归纳了学习的要点和难点,以便于学生抓住要领。

⑦ 高级语言程序设计是一门实践性很强的课程,上机实验是一个十分重要的环节,为此,本书还配有辅助教材《C 语言程序设计习题集与上机指导(第二版)》,供学习时参考使用。另外,还制作了内容生动、可下载的电子教学资料。

全书共分 13 章。第 1 章 C 语言概述,主要介绍了 C 语言的由来、特点,通过实例说明 C 语言程序的基本结构、源程序的书写风格以及 C 语言程序的运行过程,同时对 C 语言程序中使用的字符集、标识符和关键字等概念进行了具体说明。另外,还对在 Turbo C 和 Visual C++ 环境下如何运行 C 语言程序进行了介绍。第 2 章 C 语言数据类型,主要介绍了 C 语言的基本数据类型、常量和变量。第 3 章 C 语言的运算符和表达式,主要介绍了 C 语言的运算符、数据类型的自动转换、运算符的优先级和结合性以及 C 语言的表达式。第 4 章顺序结构程序设计,主要介绍了赋值表达式和赋值语句、数据的输入/输出、输入/输出函数的调用。第 5 章选择结构程序设计,主要

介绍了选择结构程序设计的思想和基本语句。第 6 章循环结构程序设计, 主要介绍了循环结构程序设计的思想、基本语句以及程序举例。第 7 章数组, 主要介绍了数组的定义、数组的维数、数组的存储、数组元素和数组下标的概念、数组的说明、数组的初始化方法、数组元素下标的取值范围、数组元素的引用、数组元素的输入/输出方法、多维数组元素的引用和初始化等问题。第 8 章函数, 主要介绍了函数的概念、函数定义与声明的基本方法、函数的传值调用、函数的嵌套调用和递归调用、变量的存储类别以及内部函数与外部函数。第 9 章预处理功能, 主要介绍了宏定义、文件包含和条件编译等。第 10 章指针, 主要介绍了指针变量的定义与初始化、指针与数组、指针与字符串、指针与函数、指针数组和指向指针的指针等。第 11 章结构体和共用体, 主要介绍了结构体、共用体、枚举类型以及链表等。第 12 章文件, 主要介绍了文件的概念、文件的打开与关闭、文件的定位、文件的读/写、文件操作的检测以及缓冲文件系统等。第 13 章位运算, 主要介绍了 C 语言中的各种位运算以及由这些运算符和相应操作构成的表达式的计算规则。

为了便于教学和学习, 与本书配套的《C 语言程序设计习题集与上机指导(第二版)》将同时出版。本书配有电子教案, 并提供程序源代码, 以方便教师教学和读者自学, 可到 <http://edu.tqbooks.net> 下载专区下载。《C 语言程序设计(第二版)》被评为普通高等教育“十一五”国家级规划教材。

本书讲授学时数为 60~116 学时, 其中包括习题、上机实验和课程设计学时数, 上机实验和课程设计部分的内容可参照配套的《C 语言程序设计习题集与上机指导(第二版)》, 各高校可根据教学课时数来确定教学内容以及相应的实验内容和课程设计内容。

本书适合作为高等学校各专业“C 语言程序设计”课程的教材, 也可供 C 语言自学者或参加各种 C 语言考试的读者及各类工程技术人员学习使用。

本书由恰汗·合孜尔主编, 其中, 第 1~8 章、10~11 章以及附录由恰汗·合孜尔编写, 第 9 章由高大利、张建民、古丽孜拉编写, 第 12 章由曹伟、加娜尔、金晓龙编写, 第 13 章由靳晟、吐尔逊、王亮亮编写。

在本书的编写过程中, 编者广泛参阅、借鉴和吸收了国内外 C 语言程序设计方面的相关教材和资料, 并吸取了这些书的优点, 在此谨向这些教材和资料的作者致以诚挚的感谢。

由于编者水平有限, 书中难免存在疏漏及不足之处, 恳请同行和广大读者批评指正。

编 者
2008 年 8 月

计算机应用能力是 21 世纪人才不可缺少的基本素质。程序设计是各专业计算机应用能力培养的重要技术基础, C 语言是目前国内外广泛使用的一种程序设计语言, 是国内外大学讲述程序设计方法的首选语言。

全国计算机等级考试、全国计算机应用技术证书考试(NIT)和全国各地区组织的大学生计算机等级考试都已将 C 语言列入了考试的范围。许多人已经用 C 语言编写应用软件, 学习 C 语言已成为广大计算机应用人员和青年学生的迫切愿望和要求。

教材是知识传授和能力培养的基础, C 语言的概念比较复杂, 规则较多, 使用灵活, 容易出错, 不少初学者感到困难。本书从程序设计的实际能力培养出发, 由浅入深、深入浅出, 将理论与实践有机结合, 容知识传授和能力培养为一体。本书内容丰富、注重实践; 突出重点、分散难点; 例题广泛、结合实际; 图文并茂、文字流畅。

本书的主要特色之一例题丰富, 每章都有丰富的例题, 并且例题涉及的面十分广泛, 力求通过实际例题的讲解, 逐步提高学生编写程序的能力。通过例题介绍了程序设计常用的各种算法和各种实际问题的处理方法, 并对各种方法进行了对比、分析, 最后给出了最优的程序。从软件工程的角度出发, 注重培养学生解决实际问题的能力, 旨在进一步巩固对基本知识的理解和掌握, 提高学生的逻辑分析、抽象思维和程序设计的能力, 培养学生良好的程序设计风格, 进而具有编写大型程序的能力。对于准备参加计算机等级考试的各类人员, 也可以通过对本书的学习, 掌握各种问题的处理方法, 以提高应试能力, 顺利通过考试。

本书由工学博士恰汗·合孜尔教授组织编著, 第 1~7、9~11 章由单洪森编写; 第 8 章由田晓东、陈大春编写。感谢广大朋友对作者完成本书所给予的帮助。

为了便于教学和学习, 与本书配套的《C 语言程序设计上机指导与习题集》将同时出版。

本书可作为高等学校各专业 C 语言程序设计课程的正式教材, 也可作为计算机等级考试教材、计算机爱好者自学用书及各类工程技术人员的参考资料。

由于作者水平有限, 书中难免有错误和不妥之处, 恳请同行和广大读者批评指正。

编者

2005 年 5 月

第1章 C语言概述.....1	2.4.4 字符串常量.....26
1.1 程序设计语言的发展历程.....1	2.4.5 符号常量.....27
1.2 C语言的发展历程.....3	2.5 变量.....28
1.3 C语言的主要特点.....4	2.5.1 变量的概念.....28
1.4 C语言程序的基本组成.....4	2.5.2 变量的定义与初始化.....28
1.5 C语言程序的书写风格.....7	2.5.3 整型变量.....29
1.6 C语言程序的编译及运行.....8	2.5.4 实型变量.....29
1.7 程序及算法.....9	2.5.5 字符变量.....30
1.7.1 程序.....9	2.6 C语言的运算符和表达式概述.....32
1.7.2 算法的概念.....9	2.6.1 运算符.....32
1.7.3 算法的描述.....10	2.6.2 表达式.....32
1.8 程序设计方法.....12	2.6.3 运算符的优先级和结合性.....32
1.8.1 结构化程序设计方法.....12	2.7 C语言中最基本的运算符和表达式.....34
1.8.2 面向对象程序设计方法.....13	2.7.1 算术运算符和算术表达式.....34
1.9 Visual C++ 6.0 的运行环境及基本操作.....14	2.7.2 赋值运算符和赋值表达式.....36
1.10 程序举例.....18	2.7.3 关系运算符和关系表达式.....38
本章小结.....19	2.7.4 逻辑运算符和逻辑表达式.....38
习题.....19	2.7.5 条件运算符和条件表达式.....40
第2章 C语言基础及顺序结构程序设计.....21	2.7.6 逗号运算符和逗号表达式.....41
2.1 C语言的字符集.....21	2.7.7 强制类型转换运算符.....42
2.2 C语言的关键字和标识符.....21	2.7.8 其他运算符.....43
2.2.1 关键字.....22	2.8 C语言的基本语句.....44
2.2.2 标识符.....22	2.9 数据的输入与输出.....45
2.3 C语言的数据类型.....22	2.9.1 字符输入/输出函数.....46
2.4 常量.....23	
2.4.1 整型常量.....24	
2.4.2 实型常量.....24	
2.4.3 字符常量.....25	

2.9.2 格式输出函数	47	5.2.3 一维数组的初始化	119
2.9.3 格式输入函数	51	5.2.4 一维数组程序举例	120
2.10 顺序结构程序设计	54	5.3 多维数组	128
2.11 程序举例	56	5.3.1 二维数组的定义	128
本章小结	59	5.3.2 二维数组元素的引用	129
习题	59	5.3.3 二维数组的初始化	129
第 3 章 选择结构程序设计	64	5.3.4 多维数组的定义	130
3.1 选择结构的概念	64	5.3.5 多维数组程序举例	130
3.2 if 语句	66	5.4 字符数组	135
3.2.1 if 语句的 3 种形式	66	5.4.1 字符数组的定义	135
3.2.2 if 语句的嵌套	72	5.4.2 字符数组的初始化	135
3.3 switch 语句	75	5.4.3 字符串及其字符串的 结束标志	137
3.4 程序举例	79	5.4.4 字符数组的输入/输出	137
本章小结	85	5.4.5 常用的字符串处理函数	138
习题	85	5.4.6 字符数组程序举例	142
第 4 章 循环结构程序设计	90	5.5 程序举例	145
4.1 循环的概念	90	本章小结	147
4.2 while 语句	91	习题	148
4.3 do...while 循环	94	第 6 章 函数与编译预处理	151
4.4 for 循环	96	6.1 函数概述	151
4.5 break 语句和 continue 语句	100	6.1.1 模块化程序设计方法	151
4.5.1 break 语句	100	6.1.2 函数的分类	152
4.5.2 continue 语句	101	6.1.3 函数的定义	153
4.6 多重循环	102	6.2 函数的调用	154
4.7 程序举例	105	6.2.1 函数的调用方式	154
4.8 3 种循环语句的比较	111	6.2.2 对被调函数原型的声明	155
本章小结	112	6.3 函数的参数传递方式与函数 的返回值	156
习题	112	6.3.1 函数的参数传递方式	156
第 5 章 数组	117	6.3.2 函数的返回值	161
5.1 数组及数组元素的概念	117	6.4 函数的嵌套调用与递归调用	162
5.2 一维数组的定义和引用	117	6.4.1 函数的嵌套调用	162
5.2.1 一维数组的定义	117		
5.2.2 一维数组元素的引用	118		

6.4.2 函数的递归调用	163	7.6 指向指针的指针	213
6.5 变量的作用域与存储类别	165	7.7 函数的指针	215
6.5.1 局部变量和全局变量	165	7.7.1 函数的指针和指向函数的 指针变量	215
6.5.2 变量的动态和静态存储 方式	168	7.7.2 函数的指针作为函数 参数	216
6.5.3 局部变量的存储类别	168	7.8 返回指针的函数	217
6.5.4 全局变量的存储类别	170	7.9 main()函数的返回值和参数	219
6.6 内部函数和外部函数	172	7.9.1 main()函数的返回值	219
6.6.1 内部函数	172	7.9.2 main()函数的参数	219
6.6.2 外部函数	172	7.10 程序举例	220
6.7 编译预处理	173	本章小结	222
6.7.1 宏定义	173	习题	223
6.7.2 文件包含	178	第 8 章 结构体和共用体	226
6.7.3 条件编译	179	8.1 结构体	226
6.8 程序举例	182	8.1.1 结构体类型的定义	226
本章小结	189	8.1.2 结构体变量的说明	227
习题	189	8.1.3 结构体变量的引用	228
第 7 章 指针	194	8.1.4 结构体变量的赋值	228
7.1 地址和指针的概念	194	8.1.5 结构体变量的初始化	229
7.2 指向变量的指针变量	195	8.1.6 结构体数组	230
7.2.1 指向变量的指针变量的 定义	195	8.1.7 指向结构体变量的指针 变量	233
7.2.2 指针运算符	195	8.2 动态存储分配与链表	234
7.2.3 指针变量的初始化	197	8.2.1 链表的概念	235
7.2.4 指针变量的运算	198	8.2.2 动态存储分配	235
7.2.5 指针变量作为函数参数	199	8.2.3 建立和输出链表	237
7.3 指针与数组	201	8.2.4 链表的基本操作	237
7.3.1 一维数组和指针	201	8.3 共用体类型	244
7.3.2 二维数组和指针	206	8.3.1 共用体类型概述	244
7.4 字符串的指针和指向字符串的 指针变量	209	8.3.2 共用体变量的定义	244
7.4.1 字符串的指针	209	8.3.3 共用体变量的引用和 赋值	245
7.4.2 字符串作为函数参数	211	8.4 枚举类型	246
7.5 指针数组	212		

8.4.1 枚举类型的定义.....	246	9.5 文件的随机读/写.....	271
8.4.2 枚举类型变量的赋值和 使用	247	9.6 文件检测函数.....	273
8.5 用户自定义类型.....	249	9.7 程序举例	273
8.6 程序举例	251	本章小结	275
本章小结	253	习题	275
习题	254		
第 9 章 文件	257	第 10 章 位运算	278
9.1 文件的基本概念.....	257	10.1 位运算符和位运算	278
9.1.1 文件的分类.....	257	10.1.1 位运算符	278
9.1.2 文件的操作过程	258	10.1.2 位运算符的运算功能... ..	279
9.1.3 文件缓冲区.....	258	10.2 位段结构	285
9.2 文件类型指针.....	259	10.2.1 位段的概念	285
9.3 文件的打开与关闭	259	10.2.2 位段结构的定义和 使用	285
9.3.1 文件的打开函数	259	10.3 程序举例	287
9.3.2 文件的关闭函数	261	本章小结	287
9.4 文件的读/写	261	习题	288
9.4.1 文件的写函数.....	261	附录 A 部分字符的 ASCII 码对照表	289
9.4.2 文件的读函数.....	265	附录 B Visual C++ 6.0 库函数.....	290
9.4.3 文件的读/写函数程序 举例	269	附录 C Visual C++ 6.0 编译错误信息....	296
		参考文献	300

C 语言是一种国内外广泛流行的、已经得到普遍应用的程序设计语言，它既可以用来编写系统软件，又可以用来编写应用软件。本章主要介绍 C 语言的发展过程、特点、结构、运行过程和算法描述以及程序设计的基本概念。

1.1 程序设计语言的发展历程

计算机是由人来指挥的，人们为了用计算机来解决实际问题，就需要编制程序。所谓程序，是指用某种程序设计语言为工具编制出来的指令序列，它表达了人们解决问题的思路，用于指挥计算机进行一系列操作，从而实现预定的功能。程序设计语言（也称为计算机语言）就是用户用来编写程序的语言，它是人与计算机之间交换信息的工具。

程序设计语言是计算机软件系统的重要组成部分，而相应的各种语言处理程序属于系统软件。程序设计语言就其发展过程和特点，一般可以分为机器语言、汇编语言和高级语言。

（1）机器语言（machine language）

计算机硬件系统只能执行由 0 和 1 二进制代码构成的操作指令。每一类计算机都有一套指令系统，指令系统中的每一条指令称为机器指令。

每一类型的计算机都有它特有的机器指令系统，这样的机器指令集合称为机器语言。用这种语言编写的程序，称为机器语言程序，计算机可以直接识别并执行机器语言程序。

由于机器语言程序直接针对计算机硬件进行操作，因此，机器语言程序的执行效率比较高，即占用空间少，执行时间短。但是，用机器语言编写程序难度比较大，容易出错，而且程序的直观性比较差，也不容易移植，即通用性差，编程效率低。例如，“10111000111010000000011”的功能仅仅表示将 1000 送入寄存器 AX 中。

（2）汇编语言（assemble language）

为了方便记忆和编制程序，人们用一些符号和简单的较接近自然语言的语句来表示二进制形式的机器指令。例如，用符号 ADD 表示加法，用符号 MOV 表示数据传递等，这样就构成了一种与机器语言对应的符号语言，这种符号化的语言称为汇编语言。例如，“10111000111010000000011”用汇编语言表示就是“MOV AX,1000”。即表示“将 1000 送入寄存器 AX 中”。从中可以看出其可读性要好于机器指令。

汇编语言编写的程序称为汇编语言源程序。这种程序计算机是不能直接识别和执行的,必须通过一个专门的程序(汇编程序)将这些符号翻译成二进制数的机器语言才能执行。这种“汇编程序”就是汇编语言的翻译程序。汇编语言程序的执行过程如图 1-1 所示。

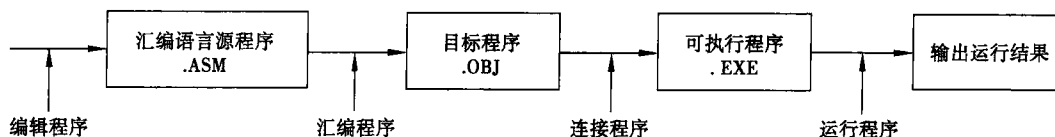


图 1-1 汇编语言程序的执行过程

汇编语言与机器语言是一一对应的关系,特别相似。它仅仅改进了指令表示方法,比机器语言易记、易读、易写。汇编语言和机器语言都是面向机器的程序设计语言,一般称为低级语言。

(3) 高级语言 (advanced language)

20 世纪 50 年代,出现了高级语言——FORTRAN。高级语言使用接近人类自然语言的语句代码来编写计算机程序,如 FORTRAN 语言、C 语言等。每种高级语言均规定了专门的词法和语法规则,使用特定的英文单词和数学符号来编写程序。例如下列语句段:

```

if (x>y)
    max=x;
else
    max=y;
  
```

以上语句段表示的是“如果 x 大于 y , 则 $\max=x$, 否则 $\max=y$ ”。对于有一定英语基础的人来说,很容易理解语句的含义,也便于记忆。

由于高级语言与具体的计算机指令系统无关,因而高级语言是一种面向操作者(用户)的语言。用高级语言编写的程序能在不同类型的计算机上运行,通用性好,这大大地促进了计算机应用的普及。

用高级语言编写的程序称为高级语言源程序。计算机也不能直接识别和执行这种程序,必须经过翻译,才能将其转换成机器语言程序执行。翻译的方法有两种,一种是解释方式,另一种是编译方式。解释方式是对高级语言编写的程序翻译一句执行一句,执行过程如图 1-2 所示;而编译方式是将高级语言编写的程序文件全部翻译成机器语言,生成可执行文件以后再执行,执行过程如图 1-3 所示。

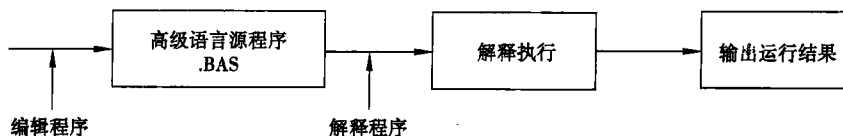


图 1-2 高级语言解释执行方式



图 1-3 高级语言编译执行方式

高级语言几乎在每一种机器上都有自己的编译程序。目前,一般的高级语言都提供了集成开发环境,它集源程序编辑、编译(解释)和执行为一体,非常方便用户使用。

随着计算机技术的发展,目前又出现了各种面向对象的计算机语言,也称为第四代语言。其特点是将实际问题中处理的事物抽象成对象和对象之间的关系,而与该事物有关的数据及对数据的具体操作都与对象封装在一起。编写程序时只要考虑如何认识问题中的对象和描述对象,而不必具体说明对象中的数据操作,只要根据对对象的描述,计算机就可以自动完成相应的数据处理工作。这种方式简化了程序设计过程,更加符合人类对客观事物的认识过程及思维方式。例如, Visual C++、Java 语言等。

1.2 C语言的发展历程

随着计算机应用的迅速发展,各种功能强大、使用方便的高级语言相继出现,高级语言使用方便,可移植性好。但高级语言一般难以实现低级语言能够直接操作计算机硬件的特点(如对内存地址的操作等)。在这样的情况下,人们希望有一种语言既有高级语言使用方便的优点,又有低级语言能够直接操作计算机硬件的优点,因此,C语言就应运而生了。

C语言的起源可以追溯到 ALGOL 60。1963年英国的剑桥大学在 ALGOL 60的基础上推出了 CPL 语言,但是 CPL 语言难以实现。1967年英国剑桥大学的 Martin Richards 对 CPL 语言进行了简化和改进,推出了 BCPL 语言。1970年美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础,又进行了进一步的简化,设计出了很简单且接近硬件的 B 语言(取 BCPL 的第一个字母),并且用 B 语言写了第一个 UNIX 操作系统,在 DEC PDP-7 型计算机上实现。1971年在 DEC PDP-11 上实现了 B 语言。1972年由美国的 Dennis M.Ritchie 在 B 语言的基础上设计出了 C 语言(取 BCPL 的第二个字母),并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。

后来,C语言进行多次改进,但主要还是在贝尔实验室内部使用。1977年 D.M.Ritchie 发表了不依赖于具体机器的 C 语言编译文本《可移植 C 语言编译程序》,使 C 语言移植到其他机器时所需要的工作大大简化了,这也推动了 UNIX 操作系统迅速地在各种机器上实现。随着 UNIX 操作系统的日益广泛使用,C语言也迅速得到推广。1978年以后,C语言先后移植到大、中、小、微型计算机上,迅速成为世界上应用最广泛的程序设计语言。

1978年 B.W.Kernighan 和 D.R.Ritchie 两人出版了 C 语言白皮书,书中给出了 C 语言的详细定义。1983年美国国家标准化协会(ANSI)根据 C 语言问世以来各种版本对 C 语言的发展和扩充,制定了 ANSI C 标准。1987年 ANSI 公布了 C 新标准 87ANSI C。后来流行的各种 C 语言编译系统的版本大多数都是以此为基础的,但是它们彼此又有不同。此后在微机上使用的 C 语言编译系统多为 Turbo C、Visual C++等,它们都是按标准 C 语言编写的,相互之间略有差异。每一种编译系统又有着不同的版本,版本越高的编译系统所提供的函数越多,编译能力越强,使用越方便,用户界面越友好。本书将以 Visual C++ 6.0 作为软件开发环境。

1.3 C 语言的主要特点

C 语言发展十分迅速, 目前已成为最受欢迎的语言之一, 主要是因为它具有强大的功能。许多著名的系统软件, 如 UNIX 操作系统就是由 C 语言编写的。另外, C 语言还成功地用于数值计算、文字处理、图形处理、数据库、计算机网络和多媒体等。归纳起来 C 语言具有下列特点:

(1) 语言简洁, 结构紧凑, 使用方便、灵活

C 语言仅有 32 个关键字和 9 种控制语句, 且源程序书写格式自由。

(2) 具有丰富的运算符和数据结构

C 语言把括号、赋值、数据类型转换等都作为运算符处理, 从而使 C 语言的运算类型极其丰富, 表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以描述的运算, 并具有现代程序设计语言的各种数据结构, 尤其是指针类型数据, 使用十分灵活和多样化。

(3) C 语言是结构化语言

C 语言是一种结构化程序设计语言, 适合于大型程序的模块化设计。

(4) C 语言允许直接访问物理地址

C 语言能够直接对内存地址进行访问操作, 可以实现汇编语言的大部分功能。所以, 它既有高级语言的功能, 又兼有汇编语言的大部分功能。有时, 也称它为“中级语言”。

(5) 生成的目标代码效率高

C 语言仅比汇编程序生成的目标代码执行效率低 10%~20%。这在高级语言中已是出类拔萃的了。

(6) C 语言适用范围广, 可移植性好

C 语言还有一个突出的优点就是基本上不做修改就能用于各种型号的计算机和各类操作系统。

1.4 C 语言程序的基本组成

【例 1.1】输出: Let's study the C language!。

```
/* This is the first C program */
#include <stdio.h>                /* 程序需要使用 C 语言提供的标准函数库 */
void main()                      /* 主函数 */
{ printf("Let's study the C language!\n"); /* 调用库函数 printf() 显示字符串 */
}
```

程序运行结果:

```
Let's study the C language!
```

程序解释:

① 程序的第 1 行是注释信息。C 语言程序中, 注释是程序员为了增加程序的可读性和易读性, 人为增加的说明性信息。其主要功能是用来说明程序的功能、用途、符号的含义或程序实现的方法等。C 语言中的注释不影响程序的功能和执行, 注释的多少都无关紧要。在 C 语言程序中, 注释由“/*”开始, 由“*/”结束, 在“/*”和“*/”之间放置注释的内容, 但注释不能嵌套, 例如, /* This is the /* first */ C program */, 则是错误的。

② 程序的第2行是C语言中的文件包含预处理命令。C语言的预处理命令都是以“#”开头。stdio.h 是一个头文件，通过文件包含命令#include，把 stdio.h 包含在用户编写的程序中，stdio.h 是关于标准输入/输出的头文件，其中包含使用标准输入/输出的许多信息。在本程序中由于第4行使用 printf()函数，因此需要用预处理命令将必要的信息包含进来。关于预处理命令将在后面做详细的介绍。

③ 程序的第3~5行是C语言程序的主函数 main()的定义。main 是主函数名，是开发系统提供的特殊函数，一个C语言程序有且仅有一个 main()函数。C语言程序执行时就是从 main()函数开始，具体讲就是从“{”开始，到“}”结束。花括号中间的部分就是 main()函数的具体实施部分，称为函数体。C语言中的函数其实就是代表实现某种功能并可重复执行的一段程序，每个函数都有一个名字，并且不能与其他的函数同名。执行一个函数称为函数调用，函数可以带参数，也可以不带参数。main 的后面跟着空的()，表明 main()函数没有参数。函数可以有返回值，也可以没有返回值，在 main()函数前加上 void (空类型)，表明 main()函数没有返回值。

④ 程序的第4行是 main()函数仅包含的一个语句，该语句仅由 printf()输出函数构成，语句后面有一个分号，表示该语句结束。C语言规定，语句以分号结束。

【例 1.2】求 a、b 两个数的和。

```
/* This is the second C program */
#include <stdio.h>                /* 程序需要使用 C 语言提供的标准函数库 */
void main()                      /* 主函数 */
{   int a,b,sum;                /* 定义 a,b,sum 三个整型变量 */
    a=123;                      /* 把常数 123 赋给变量 a */
    b=456;                      /* 把常数 456 赋给变量 b */
    sum=a+b;                    /* 将 a、b 的和赋给变量 sum */
    printf("sum is:%d\n",sum);   /* 在屏幕上输出 sum 的值 */
}
```

程序运行结果：

```
sum is: 579
```

程序解释：

① 程序的第4行是变量的定义。C语言是用变量来代表数据的。因此如果程序要处理数据(通常是这样)，就必须先定义变量。这里以整型符号 int 的方式定义了 a、b、sum 三个整型变量，变量之间以逗号分隔。

② 程序的第5~7行是赋值语句。“=”是赋值运算符，它的作用是把赋值运算符“=”右边表达式的值赋给左边的变量。语句 a=123;和 b=456;是一个赋值语句，把常数 123 和 456 分别赋给变量 a、b。语句 sum=a+b;为计算 a、b 之和，并把和值赋给变量 sum。

③ 程序的第8行是打印语句。在屏幕上输出 sum 的值，输出格式用格式字符串%d 指定。%d 表示整型格式，在输出时，该位置用对应变量 sum 的值代替。

【例 1.3】从键盘输入两个整数，在屏幕上输出它们的最大值。

```
/* This is the third C program */
#include <stdio.h>                /* 程序需要使用 C 语言提供的标准函数库 */
int max(int x,int y)             /* 定义函数 max(), 形参 x、y 为整型，返回整型值 */
```