

普通高等院校

PUTONG GAODENG
YUANXIAO
JISUANJI JICHU JIAOYU
XILIE JIAOCAI

计算机基础教育系列教材

总主编 曾 一 邹显春

C 程序设计基础

主 编 熊 壮

C CHENGXU SHEJI JICHU



重庆大学出版社

<http://www.cqup.com.cn>

清华大学出版社

清华大学出版社

清华大学出版社

C 程序设计基础

清华大学出版社



普通高等院校

PUTONG GAODENG
YUANXIAO
JISUANJI JICHU JIAOYU
XILIE JIAOCAI

计算机基础教育系列教材

总主编 曾 一 邹显春

C 程序设计基础

主 编 熊 壮

C CHENGXU SHEJI JICHU

编 者 (按姓氏笔画排序)

卢 玲 向 毅 何 丽

余 嘉 胡 勇 熊 壮

重庆大学出版社

内容提要

本书从结构化程序设计技术出发,以 C 程序设计语言为载体,贯穿“基础—应用—提高”这一主线,着重讨论 C 程序设计的基础知识,突出使用方法,面向实际应用。本书的主要内容分为两部分,第一部分主要介绍计算机程序设计高级语言共性的基础知识,包含的主要内容有:程序设计的基本概念,语言及语言处理,基本数据类型的使用,表达式运算基础,顺序程序设计,结构化程序设计,数组及其简单应用,字符串处理基础,C 程序结构,函数调用中的数值参数传递,函数调用中的地址值参数传递,函数调用中的数组参数传递,变量的作用域和生存期,文件数据处理基础。第二部分主要介绍 C 程序设计语言特有的一些重要知识,包含的主要内容有:返回指针值的函数,指向函数的指针以及指向函数指针变量作函数的形式参数,数组与指针的关系,指针数组,命令行参数,用指针实现动态数组的方法,结构体数据类型,联合体数据类型,用 typedef 关键字描述复杂数据类型,位运算与枚举类型,编译预处理基础。

与本书配套的《C 程序设计基础学习指导》同时由重庆大学出版社出版,在学习指导中提供了使用 Visual C++ 6.0 开发 C 程序的基本方法、按章节设计的实验项目以及对应章节的课后习题参考答案、ASCII 码表、C 程序设计中常用的标准库函数等重要学习资料。

本书适于高等院校各专业本专科作为程序设计语言类课程教材,同时可作为计算机专业本专科学生、计算机应用开发人员、程序设计爱好者、计算机等级考试应试者在学习程序设计语言和程序设计技术时的参考教材。

图书在版编目(CIP)数据

C 程序设计基础/熊壮主编. —重庆:重庆大学出版社,2012.8

普通高等院校计算机基础教育系列教材

ISBN 978-7-5624-6818-9

I. ①C… II. ①熊… 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(2012)第 134829 号

普通高等院校计算机基础教育系列教材

C 程序设计基础

总主编 曾 一 邹显春

主 编 熊 壮

策划编辑:王 勇 李长惠 王海琼

责任编辑:文 鹏 版式设计:王 勇

责任校对:谢 芳 责任印制:赵 晟

*

重庆大学出版社出版发行

出版人:邓晓益

社址:重庆市沙坪坝区大学城西路 21 号

邮编:401331

电话:(023) 88617183 88617185(中小学)

传真:(023) 88617186 88617166

网址: <http://www.cqup.com.cn>

邮箱: fxk@cqup.com.cn (营销中心)

全国新华书店经销

万州日报印刷厂印刷

*

开本:787×1092 1/16 印张:18.25 字数:456 千

2012 年 8 月第 1 版 2012 年 8 月第 1 次印刷

印数:1—3 000

ISBN 978-7-5624-6818-9 定价:33.00 元

本书如有印刷、装订等质量问题,本社负责调换

版权所有,请勿擅自翻印和用本书

制作各类出版物及配套用书,违者必究

编审委员会

总 主 编 曾 一 邹显春

副总主编 宋文强 熊 壮

委 员 张小莉 周建丽 张高亮 甘 玲 张裔智
肖贵元 李建华 王 宇 韦鹏程 胡方霞
周丽华 张 颖 刘慧君

序言

以云计算、物联网、下一代互联网、下一代移动通信技术为代表的新一轮信息技术革命,正在成为全球社会经济发展共同关注的重点,信息技术的创新不断催生出新技术、新产品和新应用,由信息技术引发的新兴产业形态群体正逐渐形成并逐步壮大,这将给各行各业带来更广阔的发展空间,无论是现代社会的经济运行方式、企业经营管理,或是政府运转、社会建设和管理,还是高等学校的管理模式、教学模式都将面临时代革命的挑战。

高等学校作为人才培养的重要基地,理应顺应新技术革命的要求,着力改变传统的思维模式,更新教育观念、教学内容、教学手段和教学方法,着力构建学生的个性学习、终身学习能力,培养学生“面向应用、面向职业需求”的信息化素养和技能,以增强大学生就业的竞争能力和在信息化时代的生存能力。特别是21世纪计算机技术已经由专业技术发展成为通用技术,深入到社会生活的方方面面。高校计算机基础教育在很大程度上决定着学生在现代信息社会里对计算机的认知能力和应用信息化技术解决自身领域问题的能力,也直接影响学生在职业生涯中依托信息技术的协同创新能力和基于网络的学习能力。

中国高校的计算机基础教育经历了将近30年的不断发展,已从初级阶段逐步走向成熟,形成了具有鲜明中国特色的计算机基础教育的专业化教学研究队伍和日趋完善的计算机基础教育课程体系。随着新技术的迅猛发展以及国家中长期教育改革和发展规划纲要(2010—2020年)的颁布与实施,“多元化、模块化、融合化、网络化”已成为计算机基础教学的发展趋势。这就决定了高校计算机基础教育务必顺应时代的发展要求,更新教学理念,完善课程内容,借助信息技术手段加强实践教学,培养学生自主学习能力,强化学习过程、拓展考核方式,以确保高校计算机基础教学能够有效地提升学生信息素养、促进学生专业发展、培养学生实践能力。

回顾重庆市高校计算机基础教学改革之路,我市各高校按照教育部教学指导委员会的要求,结合各高校实际,在计算机基础教育的课程体系、教学手段、教学方法等方面的改革进行了有益的探索和实践,对提高计算机基础课程教学质量、提升人才培养质量发挥了重要的作用。

为顺应重庆市计算机基础课程教学改革的需要,重庆大学出版社自1996年以来一直致力于重庆市高校计算机基础教育课程教材建设,于1996年、2000年、2006年先后3次组织重庆市各高校长期在一线从事计算机基础课教学的教师编写了计算机基础教育课程系列教材,为有效推动重庆市计算机基础课程教学改革提供了有力的保障。

为适应社会信息化进程对深化重庆市高校计算机基础教育课程教学改革的挑战,构建

面向专业需求、面向学生自主创新应用为核心的多元课程体系,更好地为不同层次、不同类型高校的计算机基础教育课程新一轮教学改革保驾护航,我们将确保重庆市计算机基础教育课程教材建设的可持续性、先进性、针对性、系统性、实用性。为此,从2011年开始,重庆大学出版社与重庆计算机学会计算机基础教育专业委员会合作,以深化重庆市高校计算机基础教育课程教学改革为依据,以满足多元需求为出发点,组织编写出版“重庆市高等院校计算机基础教育系列教材”。

为完成这套教材的编写任务,重庆计算机学会计算机基础教育专业委员会成立了编审委员会。编审委员会在重庆市各高校中精心挑选了一批长期从事计算机基础课教学的一线优秀教师组成编写队伍,他们在长期的计算机基础课程教学改革实践中锤炼了较强的教学研究能力、积累了丰富的教学经验,对教学改革实践也有很深的体会。我们深信编者将借助丰富的教材编写经验,把多年来计算机基础课程教学改革的精髓融汇到教材之中,为读者奉献一套“体系新颖、内容前瞻、突出实用、面向需求”的教材,期待系列教材能够成为践行先进教学理念的生动范例。

我们深信,这套教材的出版,将有效地深化重庆市高校计算机基础教育课程教学改革与实践,在教学观念、教学方法上,逐步形成具有重庆高校特色的计算机基础教学改革模式。

期待重庆市高校新一轮计算机基础教学改革的春风迸发出更多更新的成果。

编审委员会

2012年3月于重庆

前言

计算机程序设计语言或计算机程序设计技术课程是高等学校计算机基础教育系列课程的重要组成部分。培养学生的逻辑思维能力、抽象思维能力和基本的计算机程序设计能力,引导学生进入计算机程序设计的广阔空间,是计算机程序设计语言和计算机程序设计技术课程的主要目标。计算机程序设计的主要任务就是用合适的计算机程序设计语言对解决问题的方法进行编码处理,即编制程序。C语言功能丰富、表达能力强、程序执行效率高、可移植性好;C语言既有高级计算机程序设计语言的特点,同时又具有部分汇编语言的特点,因而C语言具有较强的系统处理能力;C语言是一种结构化程序设计语言,支持自顶向下、逐步求精的程序设计技术,通过C语言函数结构可以方便地实现程序的模块化;在C语言的基础之上发展起来的面向对象程序设计语言如C++、Java、C#等与C语言有许多的共同特征,掌握C语言对学习进而应用这些面向对象的程序设计语言有极大的帮助。

本书从结构化程序设计技术出发,以C程序设计语言为载体,通过对C语言的基本语法规则、语义以及学习C语言过程中各种常见典型问题的分析,通过对程序设计技术基础范畴内各种典型问题的求解方法描述以及相应C语言代码描述展现了在程序设计过程中如何对问题进行分析,如何组织数据和如何描述解决问题的方法,展现了在计算机应用过程中如何将方法和编码相联系的具体程序设计过程,进而向读者介绍计算机结构化程序设计的基本概念、基本技术和方法。

本书的主要内容分为相辅相成的两个部分,第一部分包括第1章至第5章,主要介绍计算机程序设计高级语言共性的基础知识,包含的主要内容有:程序设计的基本概念、语言及语言处理、基本数据类型的使用、表达式运算基础、顺序程序设计、结构化程序设计、数组及其简单应用、字符串处理基础、C程序结构、函数调用中的数值参数传递、函数调用中的地址值参数传递、函数调用中的数组参数传递、变量的作用域和生存期、文件数据处理基础。对于需要了解和掌握结构化程序设计基本思想的读者,通过这部分的学习即可较为全面地掌握结构化程序设计的基本思想。第二部分包括第6章至第10章,主要介绍C程序设计语言特有的一些重要知识,包含的主要内容有:返回指针值的函数、指向函数的指针以及指向函数指针变量作函数的形式参数、数组与指针的关系、指针数组、命令行参数、用指针实现动态数组的方法、结构体数据类型、联合体数据类型、用typedef关键字描述复杂数据类型、位运算与枚举类型、编译预处理基础。通过第二部分内容的学习可以较为全面地掌握C语言的基础知识。

与本书配套的《C程序设计基础学习指导》同时由重庆大学出版社出版,在学习指导中

提供了使用 Visual C++ 6.0 开发 C 程序的基本方法、按章节设计的实验项目以及对应章节的课后习题参考答案、ASCII 码表、C 程序设计中常用的标准库函数等重要学习资料。

本书选用 Microsoft Visual C++ 6.0 作为教学环境,书中的所有教学示例都在 Microsoft Visual C++ 6.0 集成开发环境中通过。

本书适用于高等院校各专业本专科作为程序设计语言类课程教材,同时可作为计算机专业本专科学子、计算机应用开发人员、程序设计爱好者、计算机等级考试应试者在学习程序设计语言和程序设计技术时的参考教材。

本书由熊壮、卢玲、向毅、何丽、余嘉、胡勇编著,各章节编写分工如下:熊壮(第 6 章、第 7 章),卢玲(第 8 章、第 9 章),向毅(第 5 章),何丽(第 2 章、第 3 章),余嘉(第 4 章),胡勇(第 1 章、第 10 章),全书由熊壮进行内容调整、修改,统一定稿。

本书的电子课件可在重庆大学出版社的资源网站(www.cqup.com.cn,用户名和密码:cqup)下载。

限于编者水平,书中错误和不妥之处在所难免,恳请读者不吝指教。

联系地址:重庆,重庆大学计算机学院。

E-mail:xiongz@cqu.edu.cn

编 者

2012 年 5 月

目 录

第1章 C 程序设计初步	1
1.1 程序设计语言的基本概念	2
1.2 C 语言的基本数据类型	6
1.3 C 语言基本运算符和表达式运算	12
1.4 C 语言顺序程序设计	20
习题	32
第2章 C 程序的基本控制结构	35
2.1 C 语言关系运算和逻辑运算	36
2.2 分支结构程序设计	38
2.3 循环结构程序设计	46
2.4 基本控制结构简单应用	55
习题	60
第3章 数组和字符串	67
3.1 数组的定义及数组元素的引用	68
3.2 数组的简单应用	73
3.3 字符数组和字符串	81
习题	88
第4章 函数	95
4.1 函数的定义与调用	97
4.2 函数调用中的地址值参数传递	102
4.3 函数调用中的数组参数传递	107
4.4 函数的嵌套调用和递归调用	112
4.5 变量的作用域和生存期	121
习题	130
第5章 文件数据处理基础	137
5.1 顺序存取文件处理基础	138

5.2 随机存取文件处理基础	150
习题	156
第6章 指针与函数	161
6.1 返回指针值的函数	162
6.2 指向函数的指针变量	165
习题	171
第7章 指针与数组	179
7.1 指针与数组的关系	180
7.2 指针数组与命令行参数	191
7.3 用指针构成动态数组(*)	196
习题	203
第8章 结构体和联合体数据类型	209
8.1 结构体类型的定义和使用	210
8.2 结构体数组	218
8.3 结构体数据类型与指针的关系	222
8.4 结构体类型简单应用——单链表基本操作(*)	226
8.5 联合体数据类型	232
习题	238
第9章 枚举类型和位运算	245
9.1 枚举类型及其简单应用	246
9.2 位运算及其应用	253
习题	260
第10章 编译预处理基础	265
10.1 宏定义预处理命令及其简单应用	266
10.2 文件包含预处理命令及其简单应用	270
10.3 条件编译预处理命令及其简单应用	271
习题	275
参考文献	281

C 程序设计初步

计算机应用本质上就是以计算机作为工具来解决人类社会中的实际问题,其中一个关键的步骤就是让计算机能够理解和执行人类解决某种问题的方法,从而达到解决实际问题的目的。为了能够使计算机理解人的意图,就必须解决人类和计算机相互交流的问题,将人解决问题的思路、方法和手段通过某种计算机能够理解的形式告诉计算机,使得计算机能够根据人的指令去一步一步地工作进而完成某种特定的任务。

1.1 程序设计语言的基本概念

程序设计语言是人与计算机进行信息交流的工具。程序设计是在一定的语言环境下进行的设计活动。

1.1.1 程序设计语言概述

从计算机诞生到今天,程序设计语言随着计算机技术的进步而不断发展。根据计算机技术发展的不同阶段和应用场合的不同,程序设计语言可分为三类。

1) 机器语言

机器语言是以二进制指令代码表示的指令集合,是计算机能直接识别和执行的语言。用机器语言编写的程序运行效率高,占用内存少,但缺点是面向机器,因机器不同而有差异,不同机型的代码不一样,通用性差,而且程序不直观,编程、维护都很困难。

2) 汇编语言

汇编语言是用助记符来表示指令的符号语言。每一条汇编指令和机器指令都有一一对应的关系,这些助记符通常是指令功能的英文缩写,相对机器语言而言,使用汇编语言编程比较容易,缺点是仍然面向机器,针对不同的机型,其汇编指令不同,故其通用性差。

机器语言和汇编语言都是因 CPU 不同而不同,故将它们称为面向机器的语言,亦称为低级程序设计语言。程序员用它们编程时,不仅要考虑解题思路,还要熟悉机器的内部结构。

3) 高级语言

所谓高级语言,是一种接近于人们的自然语言与数学语言的程序设计语言。用高级语言编程简单、方便、直观、易读、不易出错。

高级语言分为面向问题语言、面向结构语言和面向对象语言三种类型。所谓面向问题语言,是指用这类高级语言编程时,编程人员无须考虑机器的构造,而且无须去构造“怎么做”的解题过程,用户只需告诉计算机“做什么”,而不必告诉它“怎么做”,机器便会替你完成解题过程,比如你只要告诉计算机“找出平均成绩 90 分以上的学生名单”,计算机便会按你的意图去完成这项工作。所谓面向结构语言,又称为算法语言,是指用这类高级语言编程时,程序设计人员必须了解解决问题的全过程,不仅要告诉计算机“做什么”,还要告诉计算机“怎么做”,这种程序的执行是由程序设计人员预先设定的。所谓面向对象的语言,是指用这类高级语言来编程时要根据所要解决的问题进行数据抽象,并针对该数据的操作进行绑定,形成类,对应的实体称为对象,解决问题的算法是基于抽象数据类型的,是作用在抽象数据类型实体化的程序中的行为序列。这种程序执行的流程不是由程序设计人员预先设定的,而是由事件来驱动的。

C 语言是一种结构化程序设计语言,与其他程序设计语言相比,C 语言具有下列基本特点:

①简洁、紧凑,使用方便、灵活。C 语言一共只有 32 个保留字(C99 标准中增加了若干

保留字),9 种控制语句,区分大小写,程序书写形式自由。

②可以实现部分低级语言功能。C 语言允许直接访问地址,能进行位(bit)运算,能实现汇编语言的大部分功能,可以直接对计算机硬件进行操作。

③采用自顶向下、逐步求精的结构化程序设计方法,使用函数结构把整体程序分割成为若干个相对独立的功能模块,并且为程序模块之间的相互调用以及数据传递提供便利。

④数据类型丰富。C 语言的数据类型有:整型、实型、字符型、数组类型、指针类型、结构体类型、联合体(共用体)类型以及枚举类型等。C 语言可以实现各种复杂的数据结构,因而具有较强的数据处理能力。

⑤运算符丰富。C 语言运算符包含范围广泛,共有 34 种之多。除一般高级语言使用的四则运算(+,-,*,/)及与(&&)、或(||)、非(!)等逻辑运算功能外,还可以实现以二进制位为单位的位与(&)、位或(|)、位非(~)、位异或(^)以及移位(<<,>>)等位运算,并且具有++、--等单目运算和+=、-=、*=、/=等复合运算功能。

⑥C 系统中使用预处理语句,可以实现宏定义、文件包含以及条件编译的预处理功能。

⑦可移植性好。C 语言本身并不依赖计算机硬件系统,从而便于硬件结构不同的机型间和各种操作系统间实现程序的移植。

由于具有上述众多特点,C 语言目前仍然得到广泛普及和应用。在不同软硬件环境下的 C 语言编译系统种类有多种,它们的语句功能基本一致,但不同版本之间也有某些差异,主要体现在标准函数库中的函数种类、格式和功能上稍有差别。

1.1.2 语言处理程序概述

用汇编语言和高级语言编写的程序都称为源程序,计算机不能直接识别并执行源程序,需要通过语言处理程序将它们翻译成计算机能直接识别并执行的机器语言。对于汇编语言编写的程序,计算机系统使用称之为“汇编程序”的语言处理程序将它翻译成为计算机系统能够理解并执行的机器语言。对于高级语言编写的程序,翻译方式有解释方式和编译方式两种。

1) 解释方式

语言处理程序用解释方式处理源程序时,每读入一句高级语言的语句,就对该语句的语法、语义进行检查,若没发现错误则将其翻译成对应的执行代码并立即执行;一条高级语言的语句处理完成后,再次读入下一条语句进行处理,直至整个程序源代码处理完成为止。

2) 编译方式

语言处理程序用编译方式处理源程序时,将由高级语言编写的源程序代码一次性完全翻译成计算机能理解的目标程序。完成这个工作的语言处理程序称为“编译程序”,每种不同的计算机程序设计语言设计的程序源代码都需要使用不同的编译程序来进行处理。编译后形成的目标程序还需要通过“连接程序”的处理,将程序中所用的函数调用、系统功能调用等嵌入目标程序中,构成一个可以连续执行的二进制执行文件。用编译方式处理高级程序设计语言源程序的简单过程如图 1.1 所示。

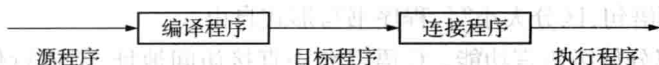


图 1.1 高级语言源程序处理过程

1.1.3 C 程序的基本结构和处理过程

任何一种计算机程序设计语言都有自己特定的字符集、语法规则、语义和一定的表现形式,程序的书写格式和构成规则是程序语言表现形式的一个重要方面。下面通过示例展示和讨论 C 程序的总体框架形式以及 C 程序的主要组成成分。

例 1.1 本示例展示的是由一个函数构成的 C 程序,为了讨论方便给程序加上了行号。

```

1  /* Name: e0101. cpp
2     程序的作用是展示 C 程序的构成成分。 */
3  #include<stdio. h>
4  void main()
5  {
6     int a,b,s; /* 定义 3 个整型变量 a,b 和 s */
7     a=12; //将 12 赋值给变量 a,即让变量 a 的值为 12
8     b=23;
9     s=a+b; //计算 a,b 两个变量的和并赋值给变量 s
10    printf("s=%d\n",s); //输出变量 s 的值(即内容)
11 }
```

一个 C 程序的主要组成部分有:合适的注释语句、编译预处理语句以及一到若干个函数,例 1.1 是由一个 C 函数构成的 C 程序。依照在程序中出现的先后次序,其主要构成成分有:

1) 注释语句

注释语句在程序中并不起功能性的作用,一个程序中是否有注释语句对其功能没有任何影响。注释语句的书写方法有两种:

①/* 字符序列 */:单/多行注释方式,即这种注释既可写在一行上,又可以写成多行的形式。例 1.1 程序中第 6 行使用的是单行注释,第 1—2 行是多行注释的使用示例,程序中往往使用多行注释来实现那些需要大量文字描述的注释内容。

②//字符序列:单行注释方式,即这种注释只能写成为单行的形式。例 1.1 程序中第 7 行、第 9 行和第 10 行使用的都是这种单行注释方式。

注释语句的作用主要有两点:一是用来做功能描述或者解释工作,帮助阅读程序的人理解程序的功能或者某些语句的意思;二是在程序的调试过程中注释掉那些暂时不需要的语句。例 1.1 程序中,如果认为语句第 8 行的语句是不需要的,则只需要用下面两种形式之一对其注释,就可以让其在程序中不起功能性的作用:

```
/* b=23; */ 或者 // b=23;
```

当需要第 8 行的语句在程序中起作用时,只需要去掉该语句的注释标记就可以恢复其在程序中原有的作用。

注释语句可以写在 C 程序任何需要书写描述性信息的地方,但需要特别注意的是,注释语句不能插入在语法上是一个基本整体的程序构成元素中,例如不能将例 1.1 程序中的第 10 行书写成:pr/* 输出变量 s 的值(即内容) */intf("s=%d\n",s);。

2) 编译预处理语句

编译预处理语句是使用#号开头的语句,如例 1.1 程序中的第 3 行。所谓编译预处理,就是 C 编译系统在对 C 源程序进行编译之前对源程序进行的一些预加工。第 3 行所示编译预处理语句的意思是将名为“stdio. h”的文件内容包含到程序中来(即逻辑上嵌入书写编译预处理语句处)以对程序中用到的输入输出标准库函数进行声明。需要提醒读者的是,由于 C 程序都要使用输入输出标准库函数实现程序中的数据输入输出,所以每个 C 程序都应该有这一行代码。除了第 3 行所示的文件包含编译预处理语句外,C 系统中还有许多其他的编译预处理语句,本书在第 10 章中将会讨论最常用的编译预处理语句使用方法。

3) 函数

C 程序由一到若干个函数构成,其中有且仅有一个主函数(即名字为“main”的函数)。主函数是 C 程序执行的入口(即执行开始的地方),一般也应该是出口(即执行完主函数中所有能够执行的语句,程序的执行过程就会结束)。由于例 1.1 是由一个函数构成的 C 程序,所以函数只能是主函数,其名字也只能是“main”。

函数由函数首部(函数头)和函数体组成。如例 1.1 程序中第 4 行即是主函数的函数首部;第 5 行的左花括号和第 11 行的右花括号构成了函数体两端的界限,在这对花括号中间的 C 语句描述了函数用到的数据和对数据的处理方法。在例 1.1 程序中,第 6—10 行代码描述了程序的功能,具体功能参见每行代码后面用注释语句对它的解释。

C 程序主函数的框架形式除了例 1.1 程序代码中展示的形式外,还常常使用另外一种返回整型的形式,而且这种返回整型的形式是很多开发环境指定的 C 程序主函数书写形式。图 1.2 同时展示了这两种最常用的主函数框架,本教材选用的是返回 void 类型的主函数框架形式。

<pre>void main() { //主函数 C 代码 }</pre>	<pre>int main() { //主函数 C 代码 return 0; }</pre>
(a) 返回Void型	(b) 返回int型

图 1.2 C 程序主函数两种最常用框架

如前所述,使用 C 语言编写的源程序也需要通过编译、连接等处理后才能被计算机系统执行,C 程序处理的基本过程和步骤如图 1.3 所示。

一个 C 程序的完整处理过程主要有程序的设计和录入编辑、编译、连接、运行等步骤。本教材推荐使用微软的 Microsoft Visual C++ 6.0 集成开发环境,使用 VC++6 开发 C 程序的方法请参照与本书配套的《C 程序设计基础学习指导》。如图 1.3 所示,开发 C 程序各阶

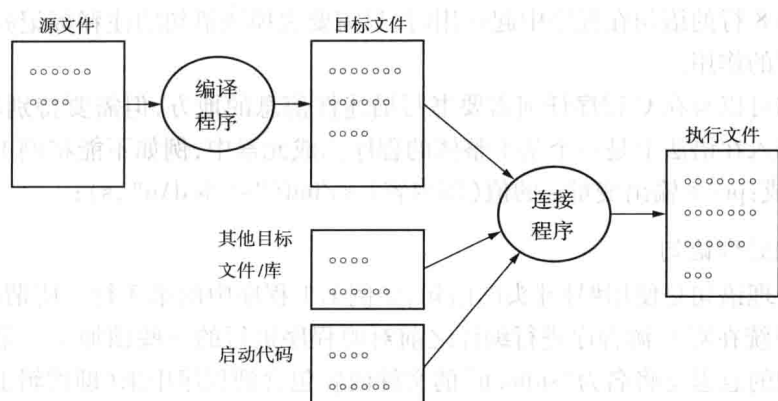


图 1.3 C 程序的处理过程

段的主要任务如下所示：

1) 程序设计和源程序代码的录入编辑

编写程序时,首先通过对问题的理解确定程序中所用数据的名字、类型和个数,设计出解决问题的方法(即算法);然后在开发环境提供的源程序代码编辑窗口中将数据和设计好的算法用C语言规定的格式组成为完整的C程序结构,并将编辑完成的C程序保存为源程序文件。

2) 对源程序代码进行编译

程序源代码录入编辑完成后,编译程序对设计好的源程序文件进行处理,经过对源程序代码的词法分析、语法分析、语意分析、代码生成和代码优化等阶段将所处理的源程序转换为用二进制代码表示的目标程序。

3) 对目标代码进行连接

由于在程序中必然会用到C语言的标准库函数、系统功能调用等,这些执行代码并不会自动添加到二进制目标文件中,所以二进制目标代码虽然能够被计算机系统所理解,但并不能直接执行。使用连接程序对二进制目标文件进行处理,就是将程序中所用的标准函数调用、系统功能调用等嵌入目标程序中,构成一个可以连续执行的二进制执行文件。

4) 执行程序

计算机系统执行这个二进制执行文件就可以实现程序员在源程序文件中所描述和指定的相应功能。

上述C程序设计的四个处理步骤根据需要可以反复进行,直至程序运行的结果能够达到预期的目标为止。

1.2 C语言的基本数据类型

程序包含两方面内容:数据描述和算法描述。数据是程序的必要组成部分,是程序的处理对象。数据在程序中是以变量或者常量的形式来描述的,每个数据都具有确定的数据类型,如整型类数据、实型类数据、字符类数据等。不同类型的数据在内存中分配的存储单