

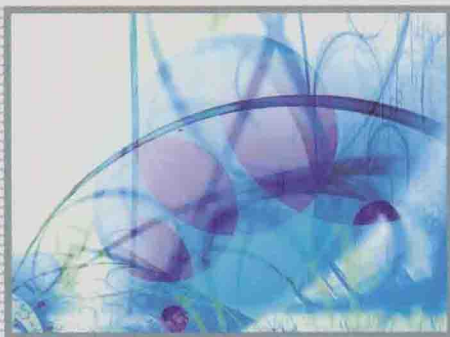
21 世纪高等教育计算机技术规划教材



21 ShiJi GaoDeng JiaoYu JiSuanJi JiShu GuiHua JiaoCai

C 语言程序设计

C Language Program Design



徐新爱 胡佳 主编
吴瑜鹏 卢昕 副主编

 人民邮电出版社
POSTS & TELECOM PRESS

21 世纪高等教育计算机技术规划教材

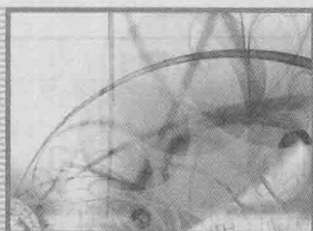


21 ShiJi GaoDeng JiaoYu JiSuanJi JiShu GuiHua JiaoCai

C 语言程序设计

C Language Program Design

徐新爱 胡佳 主编
吴瑜鹏 卢昕 副主编



人民邮电出版社

北京

图书在版编目(CIP)数据

C语言程序设计 / 徐新爱, 胡佳主编. — 北京: 人民邮电出版社, 2014.9
21世纪高等教育计算机技术规划教材
ISBN 978-7-115-36052-6

I. ①C… II. ①徐… ②胡… 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(2014)第156813号

内 容 提 要

全书由浅入深地介绍了C语言的基本理论、基本知识以及编程的基本技能和方法。全书共12章,内容包括:C语言程序设计概述,C语言源程序的基本结构,基本数据类型、运算符和表达式,顺序结构程序设计,选择结构程序设计,循环结构程序设计,数组,函数,指针,用户自定义数据类型,文件和预处理命令。每章都附有丰富的习题,帮助读者巩固所学知识。

本书可以作为高等院校“C语言”课程的教材,也可作为初学者的自学用书。

◆ 主 编 徐新爱 胡 佳

副 主 编 吴瑜鹏 卢 昕

责任编辑 王 平

责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京鑫正大印刷有限公司印刷

◆ 开本: 787×1092 1/16

印张: 19.5

2014年9月第1版

字数: 499千字

2014年9月北京第1次印刷

定价: 44.00元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

前言 PREFACE

C 语言是目前世界上最流行、使用最广泛的高级程序设计语言之一。它不仅具有丰富灵活的控制和数据结构,简洁而高效的表达式语句,清晰的程序结构和良好的可移植性等优点,还具有直接操纵计算机硬件的强大能力。因此,C 语言已成为学习计算机程序设计的基础语言,众多的程序设计课程如 C++、Java、C#、J#、Perl 等很多新型的语言均以 C 语言为基础。

2011 年,南昌师范学院《C 语言程序设计》课程被江西省教育厅评为江西省高等院校高职高专精品课程,众位编者将积累多年的教学经验和应用 C 语言的体会,加之新的教改思想倾注于此本书中。

1. 本书特色

在编写的过程中,注重 C 语言在学科中的基础地位,以程序设计为中心,内容全面、概念清晰、层次分明,讲述力求循序渐进、深入浅出、通俗易懂,尽可能地贴近读者的接受能力,并注重培养良好的程序设计风格和习惯。尤其是注重培养读者分析问题和实际编程能力。

(1) 注重章节学习意义,提出每章的学习目标。让读者在学习的过程中,掌握每章学习的意义及有关知识,避免被动地学习和教条式学习。

(2) 对于每章的编程例题,尤其是 3 种基本结构的例题介绍中,通过算法分析告诉读者怎么做,通过编写源程序的基本步骤告诉读者编写代码的基本方法。

(3) 教材中部分章节的典型应用按照问题描述、算法分析、编写源程序的基本步骤和源代码展示及运行结果的顺序编写,为读者提供解决具体应用问题的基本思路。

(4) 每章的常见错误集锦旨在提醒读者在编写程序的过程中容易出错的地方。

(5) 每章配备了丰富的习题,类型多样,难度各异,具有广泛的代表性和实践性。同时,配套出版了《C 语言程序设计习题解答与实验指导》,提供了全部习题解答和与实验相关的内容。

2. 章节内容

全书由浅入深地介绍了 C 语言的基本理论、基本知识以及编程的基本技能和方法,使读者能全面、系统地理解和掌握用 C 语言进行程序设计的方法。全书以 VC++6.0 为编译平台,全面介绍 ANSI C,不仅涵盖 C 语言的基本知识,而且注重 C 语言程序的详细讲解,更注重初学者如何用计算机程序设计语言来解决实际问题的能力的培养。全书共 12 章,第 1 章全面介绍 C 语言的概貌,包括编程的准备工作、程序设计语言的发展过程、C 语言程序的基本特点以及 C 语言的编译平台,第 2 章~第 6 章介绍 C 语言的基础语法知识和 3 种基本结构,包括关键字与标识符、常量与变量、基本数据类型、运算符与表达式、C 语句、数据的输入输出等知识点,第 7 章、第 8 章、第 10 章、第 11 章和第 12 章介绍 C 语言提供的构造数据类型的使用,包括数组、结构体、共用体和文件等,第 9 章介绍 C 语言的精髓——指针。

本书由徐新爱、胡佳任主编,吴瑜鹏、卢昕任副主编,其中,第 1~6 章、第 12 章及附录 1~4 由徐新爱编写,第 7 章~9 章由胡佳编写,第 10 章由吴瑜鹏编写,第 11 章

由卢昕编写。最后由徐新爱负责全书的修改及统稿工作。在本书的编写过程中，还得到了计算机教研室的全体教师的大力支持，我们深表感谢。本书配套的电子教案及相关资料登录人民邮电出版社的教学服务与资源网（<http://www.ptpedu.com.cn>）下载。

由于编者水平有限，加之时间仓促，书中难免有错漏之外，敬请广大读者批评指正。

编 者

2014年5月

目 录 CONTENTS

第 1 章 C 语言程序设计概述 1

◇学习意义	1	1.3.3 结构化程序设计的基本结构	13
◇学习目标	1	1.3.4 结构化程序设计的基本特点	14
1.1 编程的预备工作	1	1.4 C 语言的发展历史及特点	14
1.1.1 学习编程的心理准备	1	1.4.1 C 语言的发展历史	14
1.1.2 认识编程	2	1.4.2 C 语言的特点	16
1.1.3 数据在计算机中的存储形式	2	1.4.3 C 语言的应用	17
1.2 程序设计语言的基础	5	1.5 开发环境简介	17
1.2.1 程序设计语言的发展	5	1.5.1 Turbo C 开发环境	17
1.2.2 程序设计语言的特点及发展趋势	8	1.5.2 Dev-C++开发环境	18
1.2.3 程序设计的基本过程	9	1.5.3 VC++ 6.0 开发环境	18
1.3 结构化程序设计	12	1.6 编制 C 语言程序的基本步骤	19
1.3.1 什么是结构化程序设计	12	1.7 本章小结	22
1.3.2 结构化程序设计的基本原则	13	习题	22

第 2 章 C 语言源程序的基本结构 24

◇学习意义	24	2.4 带参数的 main 函数	29
◇学习目标	24	2.5 由多个文件构成的源程序	30
2.1 源程序的基本结构	24	2.6 本章小结	33
2.1.1 认识 C 语言源程序	24	2.6.1 知识梳理	33
2.1.2 源程序的基本结构	26	2.6.2 如何编程	33
2.2 源程序的标识符	27	习题	33
2.3 源程序的基本语句	28		

第 3 章 基本数据类型、运算符和表达式 36

◇学习意义	36	3.3.3 字符型数据	44
◇学习目标	36	3.3.4 字符串常量	46
3.1 C 语言的数据类型	36	3.4 常用运算符与表达式	47
3.2 数据的表现形式	37	3.4.1 算术运算符及其表达式	47
3.2.1 常量	37	3.4.2 自增自减运算符、负号运算符	48
3.2.2 变量	39	3.4.3 赋值运算符及其表达式	49
3.3 基本数据类型	39	3.4.4 强制类型转换运算符	50
3.3.1 整型数据	40	3.4.5 逗号运算符及其表达式	51
3.3.2 实型数据	43	3.4.6 sizeof 运算符	51

3.5 常见数学运算表达式在 C 语言中的表示	52	3.6.1 知识梳理	52
3.6 本章小结	52	3.6.2 常见错误集锦	52
		习题	54

第 4 章 顺序结构程序设计 57

◇学习意义	57	4.4.1 数字分离问题	66
◇学习目标	57	4.4.2 图形的面积等计算问题	67
4.1 程序的 3 种基本结构	57	4.4.3 数的交换问题	68
4.2 顺序结构程序设计的思想	58	4.4.4 大小写转换问题	69
4.3 实现顺序结构程序设计的基本语句	59	4.5 本章小结	69
4.3.1 赋值语句	59	4.5.1 知识梳理	69
4.3.2 数据的基本输入与输出	59	4.5.2 常见错误集锦	70
4.4 顺序结构程序设计的典型应用	66	习题	71

第 5 章 选择结构程序设计 73

◇学习意义	73	5.3 选择结构程序设计的典型应用	84
◇学习目标	73	5.3.1 数的最值问题	84
5.1 关系运算符、逻辑运算符和条件运算符	73	5.3.2 方程根问题	85
5.1.1 关系运算符及其表达式	73	5.3.3 奖金问题	87
5.1.2 逻辑运算符及其表达式	74	5.3.4 运算器问题	89
5.1.3 条件运算符及其表达式	76	5.4 本章小结	91
5.2 选择结构程序设计	76	5.4.1 知识梳理	91
5.2.1 if 语句	76	5.4.2 常见错误集锦	93
5.2.2 switch 语句	82	习题	96

第 6 章 循环结构程序设计 102

◇学习意义	102	6.2 循环结构语句的选择	117
◇学习目标	102	6.3 循环结构程序设计的典型应用	118
6.1 循环结构程序设计	102	6.3.1 累加或累乘问题	118
6.1.1 for 语句	103	6.3.2 数的判断问题	120
6.1.2 while 语句	105	6.3.3 经典数学问题	121
6.1.3 do-while 语句	107	6.3.4 图形输出问题	123
6.1.4 goto 语句	108	6.4 本章小结	126
6.1.5 for 语句的其他格式	109	6.4.1 知识梳理	126
6.1.6 循环嵌套	111	6.4.2 常见错误集锦	126
6.1.7 break 语句与 continue 语句	114	习题	127

第7章 数 组 132

◇学习意义	132	7.3.3 常用字符串处理函数	144
◇学习目标	132	7.4 数组的典型应用	147
7.1 一维数组	132	7.4.1 最大值和最小值问题	147
7.1.1 一维数组的定义和引用	132	7.4.2 杨辉三角形问题	148
7.1.2 一维数组的初始化	133	7.4.3 矩阵运算问题	150
7.2 二维数组	136	7.4.4 字符串处理问题	152
7.2.1 二维数组的定义和引用	136	7.5 本章小结	153
7.2.2 二维数组的初始化	137	7.5.1 知识梳理	153
7.3 字符串与字符数组	144	7.5.2 常见错误集锦	154
7.3.1 字符数组的定义和引用	144	习题	154
7.3.2 字符数组的赋值	144		

第8章 函 数 159

◇学习意义	159	8.5.2 局部变量的作用域和生存期	175
◇学习目标	159	8.5.3 全局变量的作用域和生存期	176
8.1 函数概述	159	8.6 变量的存储类型	176
8.2 函数的定义与调用	160	8.7 函数的作用域	180
8.2.1 无参数无返回值的函数	161	8.8 函数的典型应用	181
8.2.2 无参数有返回值的函数	164	8.8.1 数的最值问题	181
8.2.3 有参数无返回值的函数	165	8.8.2 最大公约数和最小公倍数问题	182
8.2.4 有参数有返回值的函数	166	8.8.3 阶乘问题	183
8.3 函数参数的传递方式	168	8.8.4 汉诺塔问题	184
8.4 函数的嵌套与递归调用	170	8.9 本章小结	186
8.5 变量的作用域与生存期和 存储类型	174	8.9.1 知识梳理	186
8.5.1 变量的作用域和生存期的概念	174	8.9.2 常见错误集锦	186
		习题	187

第9章 指 针 194

◇学习意义	194	9.1.5 一维数组的指针与指向一 维数组的指针变量	200
◇学习目标	194	9.1.6 字符数组的指针与指向字符 数组的指针变量	201
9.1 指针的基本概念	194	9.1.7 指针作为函数的参数	203
9.1.1 指针与指针变量的概念	194	9.2 指针进阶	205
9.1.2 指针变量的定义和引用	195	9.2.1 指针与动态内存分配	205
9.1.3 指针的基本运算	197	9.2.2 数组指针	206
9.1.4 变量的指针与指向变量的 指针变量	199		

9.2.3 指针数组	208	9.3.3 多个城市地名排名问题	215
9.2.4 指针函数与函数指针	210	9.4 本章小结	216
9.2.5 多级指针	211	9.4.1 知识梳理	216
9.3 指针的典型应用	212	9.4.2 常见错误集锦	217
9.3.1 任意个整数求和问题	212	习题	218
9.3.2 冒泡排序问题	213		

第 10 章 用户自定义数据类型 223

◇学习意义	223	10.3.1 共同体类型的定义	237
◇学习目标	223	10.3.2 共用体变量的定义和引用	238
10.1 结构体	223	10.3.3 共用体变量的赋值	239
10.1.1 结构体类型的定义	223	10.4 枚举类型变量的定义和引用	241
10.1.2 结构体变量的定义	224	10.5 类型定义	245
10.1.3 结构体变量的引用和赋值	225	10.6 用户自定义数据类型的典型应用——学生信息的基本操作	247
10.1.4 结构体数组	229	10.7 本章小结	253
10.1.5 结构体和指针	233	10.7.1 知识梳理	253
10.2 线性链表	234	10.7.2 常见错误集锦	254
10.2.1 线性链表及其结构	234	习题	254
10.2.2 线性链表的基本操作	235		
10.3 共用体	237		

第 11 章 文 件 257

◇学习意义	257	11.3 文件读写	262
◇学习目标	257	11.3.1 文件读写函数	262
11.1 文件概述	257	11.3.2 文件读写函数选用原则	271
11.1.1 文件的概念	257	11.4 文件的定位与随机读写	271
11.1.2 文件的分类	258	11.5 文件的出错检测	274
11.1.3 文件的处理方法	259	11.6 文件的典型应用——文件复制	275
11.2 文件的打开与关闭	259	11.7 本章小结	277
11.2.1 文件指针	259	11.7.1 知识梳理	277
11.2.2 文件的打开	260	11.7.2 常见错误集锦	277
11.2.3 文件的关闭	261	习题	278
11.2.4 exit()函数	262		

第 12 章 预处理命令 280

◇学习意义	280	12.4 本章小结	286
◇学习目标	280	12.4.1 知识梳理	286
12.1 文件包含命令	280	12.4.2 常见错误集锦	287
12.2 宏定义	282	习题	287
12.3 条件编译	284		

附录 288

附录 1	ASCII 码对照表	288	附录 3	常见的 C 语言库函数	291
附录 2	C 语言的保留字	290	附录 4	编译常见错误中英文对照表	297

参考文献 302

第1章

C语言程序设计概述

◇学习意义

本章介绍了编程的预备工作，从认识编程入手，逐步了解计算机程序设计语言的基础知识，帮助读者了解程序设计的基本过程。

通过介绍C语言的历史及发展可以初步认识计算机程序设计语言，同时了解C语言常用的几种编译平台，掌握C语言程序编译的基本步骤。

◇学习目标

- (1) 了解编程的预备工作
- (2) 了解程序设计语言的发展
- (3) 掌握程序设计的基本过程
- (4) 了解C语言的发展及特点
- (5) 了解C语言的开发环境
- (6) 掌握VC++6.0编译C语言程序的基本步骤

1.1 编程的预备工作

1.1.1 学习编程的心理准备

随着信息技术及 Internet 的普及，人们对计算机功能的要求也越来越高，如办公自动化、购物网络化、理财网络化、生活智能化等。有预言说，编程技能变得越来越重要，将会变成 21 世纪生存技能中的核心竞争力。同时，随着时间的推移，编程的工作岗位也将有大幅增加。据美国劳工统计局的数据，在 2010 年有 91.3 万个计算机程序员职位，到 2020 年这一岗位预计将增长 30%，其他所有的非农就业岗位平均增幅预计只有 14%。因此，希望有志于编程的读者能够继续努力学习，实现自己的终极理想。

1. 培养兴趣，对事物永远保持一颗好奇心

兴趣指兴致，对事物喜好或关切的情绪。心理学认为兴趣是人们力求认识某种事物和从事某项活动的意识倾向。它表现为人们对某件事物、某项活动的选择性态度和积极的情绪反应。兴趣在人的实践活动中具有重要的意义，可以使人集中注意力，产生愉快紧张的心理状态。如果一直待在兴趣范围内，就会产生学习的欲望。同时，对事物保持一颗好奇心，也会触发学习的动机，积极开展学习，享受学习的过程和学习带来的成就感，形成良性循环。

2. 制订目标，通过实现目标获得成就感

学习过程中，制订计划是达到有效学习的一个重要方法。大学生活平淡无奇，有时觉

得自己在不断地浪费时间，没有学到具体的知识，没有实现自己的目标。怎么办呢？最有效直接的方法就是制订自己的学习计划。计划可以包括短期的计划，如日计划、周计划，也可以是长期的计划，如月计划、学期计划或学年计划。通过制订计划，按照计划去完成一定的目标。在完成一定的学习目标后，可以适当对自己进行奖励，让自己的学习热情更持久，成就感也会更加强烈，就有了挑战下一个目标的动力。同时，还可以积极参加一些计算机专业的学科竞赛，拓展自己的视野，结交更多志同道合的朋友，制订更高的目标。

3. 动手动脑，提高学习效果

很多学生在学习编程的过程中，经常会有这样的问题，“书都看了，也都看懂了，但还是写不出代码”。像这种情况的产生一般都是因为没有积极主动去思考和练习所致。在阅读程序的过程中，还要不断思考为什么要这样写，甚至可以质疑书上的内容，做到举一反三。在解决问题的过程中，如果有哪个环节存在疑问，一定要积极地去解决，绝不要把问题带到下一个学习环节。

1.1.2 认识编程

人与人之间交流的语言称为自然语言（Natural Language），而计算机是一种机器，它能够识别的语言称为机器语言（Machine Language）。可见，人与计算机使用的是不同的语言，人不可能去学习计算机的机器语言，计算机也不可能学习人类的自然语言。因此，科学家们就设计了人与机器之间交流的工具——编程语言（Programming Language），又称为程序设计语言（Program Design Language, PDL）。其中，唯一想到利用程序设计语言来解决问题的人是德国工程师康拉德·楚泽，他创造了编程语言。

编程就是让计算机为解决某个问题而使用某种编程语言编写程序代码，并最终得到结果的过程。为了使计算机能够理解人的意图，人类就必须要将解决问题的思路、方法和手段通过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一步一步去工作，完成某种特定的任务。这种人和计算机之间交流的过程就是编程。

通过编程语言，人类将问题转化成语言中的合法符号表达出来，然后通过转换软件转换成机器语言，最终交由计算机执行。在这个过程中，能使用编程语言写程序，并以此为职业的人，称为程序员（Programmer），或者程序设计师。程序员写出来的原始程序称为源代码（Source Code）、代码（Code）或源码。它们之间的关系如图 1-1 所示。



图 1-1 编程的基本过程

1.1.3 数据在计算机中的存储形式

计算机处理的信息可以是数字、文字、图像、声音等，这些信息数据分为数值型数据和非数值型数据。但不管是哪种类型的信息在计算机中最终都是以二进制数据信息来表示和处理。因此，需要计算机加工处理时，首先必须按一定的规则转换成二进制数。

1. 字节

计算机处理数据时，都是把数据存放在计算机硬件系统的内存中。计算机系统的内存储器是分为若干个存储单元，每个存储单元的大小为一个字节（byte）。字节是计算机存储容量的基本单位，每个字节由 8 个二进制位（bit）组成。每个字节中的每个二进制位的取值是 0 或 1，最右端的二进制位称为最低位或第 0 位，最左端的二进制位称为最高位或第 7 位。一个字符占用一个字节，一个汉字占用两个字节，一个 int 型数据占两个字节。CPU 一次能处理的二进制

数的位数称为字长。

2. 进制

进制也就是进位制，是人们规定的一种进位方法。计算机中用来表示数有多种的进制方法，有十进制、八进制、十六进制和二进制。十进制是人们日常生活中使用的进制。因此，给定的一个数没有特别的说明都是以十进制表示。数据在计算机中是以二进制形式存放的，这是因为二进制形式具有容易表示、运算规则简单和节省设备的优点。为了方便，又引入了八进制和十六进制。以此类推，若只用 r 个基本符号 ($0 \sim r-1$) 表示数值，则称为 r 进制， r 称为该数制的基本数 (Radix)。

对于任何一种进制—— X 进制，表示某一位置上的数运算时是逢 X 进一位。例如，十进制是逢十进一，十六进制是逢十六进一，二进制就是逢二进一。不同进制的共同特点如下所述。

(1) 每一种数制都有固定的数字。例如，十进制数制的基本数字有 0, 1, 2, ..., 9。二进制数制的基本数字有两个，即 0 和 1。

(2) 每一种数制都使用位置表示法。即处于不同位置的数符所代表的值不同，与它所在位的权值有关。例如，二进制整数 N 和小数 N 分别表示为：

$$\begin{aligned} N &= (a_n a_{n-1} \cdots a_1 a_0)_2 \\ &= a_n \times 2^n + a_{n-1} \times 2^{n-1} + \cdots + a_1 \times 2^1 + a_0 \times 2^0 \\ &= 2 \times (a_n \times 2^{n-1} + a_{n-1} \times 2^{n-2} + \cdots + a_1 \times 2^0) + a_0 \\ N &= (a_{-1} a_{-2} \cdots a_{-m})_2 \\ &= a_{-1} \times 2^{-1} + a_{-2} \times 2^{-2} + \cdots + a_{-m} \times 2^{-m} \\ &= 2^{-1} \times (a_{-1} + a_{-2} \times 2^{-1} + \cdots + a_{-m} \times 2^{-m+1}) \end{aligned}$$

可以看出，各种进位计数制中权的值正好是基数的某次幂。因此，对任何一种进位计数制表示的数都可以写成按权展开的多项式。

计算机表示数的不同进制的数码、基数、权和运算规则如表 1-1 所示。

表 1-1 不同进制的数码、基数、权和运算规则

数制	十进制数	二进制数	八进制数	十六进制数
数码	0~9	0, 1	0~7	0~9, A~F (a~f)
基数	10	2	8	16
权	从低位到高位依次为： $10^0, 10^1, 10^2, \cdots, 10^n$ 从低分位到高分位依次为： $10^{-1}, 10^{-2}, 10^{-3}, \cdots, 10^{-m}$	从低位到高位依次为： $2^0, 2^1, 2^2, \cdots, 2^n$ 从低分位到高分位依次为： $2^{-1}, 2^{-2}, 2^{-3}, \cdots, 2^{-m}$	从低位到高位依次为： $8^0, 8^1, 8^2, \cdots, 8^n$ 从低分位到高分位依次为： $8^{-1}, 8^{-2}, 8^{-3}, \cdots, 8^{-m}$	从低位到高位依次为： $16^0, 16^1, 16^2, \cdots, 16^n$ 从低分位到高分位依次为： $16^{-1}, 16^{-2}, 16^{-3}, \cdots, 16^{-m}$
运算规则	逢十进一	逢二进一	逢八进一	逢十六进一

3. 进制间的相互转换

(1) 八进制、十六进制、二进制转换为十进制。八进制、十六进制、二进制转换为十进制

的规则是将八进制、十六进制、二进制的数按权展开，然后计算该多项式的各项和，此“和”就是十进制表示的数。

【例 1-1】：二进制数 1011.1 转换为十进制的数。

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} = 11.5$$

【例 1-2】：八进制数 13.1 转换为十进制的数。

$$1 \times 8^1 + 3 \times 8^0 + 1 \times 8^{-1} = 11.125$$

【例 1-3】：十六进制数 13.1 转换为十进制的数。

$$1 \times 16^1 + 3 \times 16^0 + 1 \times 16^{-1} = 19.0625$$

(2) 十进制转换为八进制、十六进制、二进制。十进制转换为八进制、十六进制、二进制的规则是采用短除法，其中整数部分除以基数 (r) 取余，小数部分乘以基数 (r) 取整。

【例 1-4】：十进制数 10.25 转换为二进制的数。

具体方法为：整数部分除以 2，每一步取余；小数部分乘以 2，每步取结果的整数部分，具体步骤如图 1-2 所示。

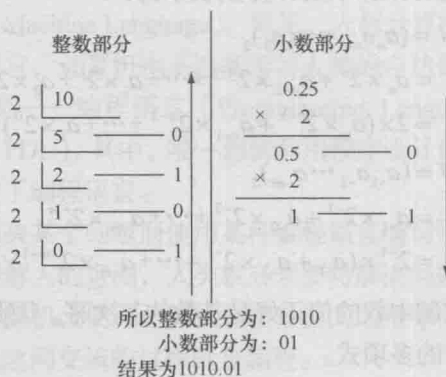


图 1-2 转换过程

十进制转换为八进制、十进制转换为十六进制都与上面的步骤一样，只是将 2 改成 8 或者 16。

(3) 二进制与八进制或十六进制相互转换。二进制转换成八进制的规则是：以小数点为分界点，把二进制数的整数部分从右往左每 3 位分成一组，不够 3 位的前面补 0；把二进制数的小数部分从左往右每 3 位分成一组，不够 3 位的后面补 0，然后把每一组对应的十进制数算出，顺序排列起来就得到了八进制数。

【例 1-5】：二进制数 10101111.10111 转换为八进制的数。

具体步骤为：

第 1 步（划分组）： (010 101 111.101 110)

第 2 步（求对应的十进制数）： 2 5 7. 5 6

第 3 步（八进制）： 257.56

二进制转换成十六进制的规则是：以小数点为分界点，把二进制数的整数部分从右往左每 4 位分成一组，不够 4 位的前面补 0；把二进制数的小数部分从左往右每 4 位分成一组，不够 4 位的后面补 0，然后把每一组对应的十六进制数算出，顺序排列起来就得到了十六进制数。

【例 1-6】：二进制数 10101111.10111 转换为十六进制的数。

具体步骤为:

第1步(划分组): (1010 1111.1011 1000)

第2步(求对应的十六进制数): A F. B 8

第3步(十六进制): AF.B8

反过来,将一个八进制数转换为二进制数的规则是将该八进制数的每一位,按照十进制转换二进制的方法变成用3位二进制表示的序列,然后按照顺序排列,就转换为二进制数了。

【例1-7】:八进制数257.56转换为二进制的数。

$$\begin{aligned} & \quad \quad \quad 2 \quad 5 \quad 7.5 \quad 6 \\ & = (010 \ 101 \ 111.101 \ 110) \\ & = 10101111.101 \end{aligned}$$

同样,将一个十六进制数转换为二进制的规则是将该十六进制数的每一位,按照十进制转换二进制的方法变成用4位二进制表示的序列,然后按照顺序排列,就转换为二进制数了。

【例1-8】:十六进制数AF.B8转换为二进制的数。

$$\begin{aligned} & \quad \quad \quad A \quad F. \quad B \quad 8 \\ & = (1010 \ 1111.1011 \ 1000) \\ & = 10101111.10111 \end{aligned}$$

4. 数据在内存中的存储形式

所有的数据在计算机中都是以二进制形式存放的,其机内表示形式称为机器数。机器数的表示方法通常有原码、反码和补码3种形式。根据它们的运算规则,补码运算最简单,可连同符号位一起参与运算。因此,计算机系统规定:数据都是以补码的形式存放在内存中。

(1) 原码。数值的原码是指将最高位用作符号位,其他各位表示数据值本身绝对值的二进制形式。其中0表示正号,1表示负号。

【例1-9】:若采用8个二进制位存储数据(此位数称为机器字长),则

$$\begin{aligned} [+1]_{\text{原}} &= 0000 \ 0001 & [-1]_{\text{原}} &= 1000 \ 0001 \\ [+45]_{\text{原}} &= 0010 \ 1101 & [-45]_{\text{原}} &= 1010 \ 1101 \end{aligned}$$

(2) 反码。对于一个带符号的数来说,正数的反码与其原码相同,负数的反码为其原码除符号位以外的各位按位取反。

【例1-10】:若机器字长 n 等于8,则

$$\begin{aligned} [+1]_{\text{反}} &= 0000 \ 0001 & [-1]_{\text{反}} &= 1111 \ 1110 \\ [+45]_{\text{反}} &= 0010 \ 1101 & [-45]_{\text{反}} &= 1101 \ 0010 \end{aligned}$$

(3) 补码。数值的补码的最高位是符号位,其他位取决于该数是正数还是负数。其中,正数的补码与其原码相同,负数的补码为其反码在最低位加1。

【例1-11】:若机器字长 n 等于8,则

$$\begin{aligned} [+1]_{\text{补}} &= 0000 \ 0001 & [-1]_{\text{补}} &= 1111 \ 1111 \\ [+45]_{\text{补}} &= 0010 \ 1101 & [-45]_{\text{补}} &= 1101 \ 0011 \end{aligned}$$

1.2 程序设计语言的基础

1.2.1 程序设计语言的发展

编程语言又称为程序设计语言,是进行程序设计的必要工具,是一组用来定义计算机程序

的语法规则。它是一种被标准化的交流技巧,用来向计算机发出指令。自 20 世纪 60 年代以来,世界上公布的程序设计语言已有上千种之多,但是只有很小一部分得到了广泛的应用。程序设计语言的发展历程如图 1-3 所示。

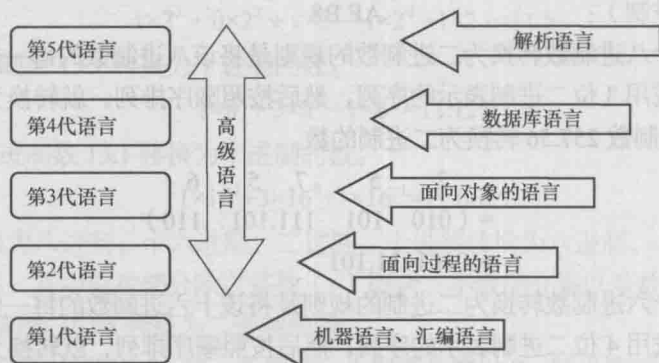


图 1-3 程序设计语言的发展历程

从以上程序设计语言的发展历程图来看,程序设计语言可以分为 5 代,其中除第 1 代之外,其余 4 代都属于高级语言,其中第 2 代语言和第 3 代语言往往习惯于作为高级语言的典型代表,而第 4 代语言和第 5 代语言更具有特定的应用,因此,下面将计算机程序设计语言的发展分为四个阶段进行介绍。

1. 第 1 阶段

第 1 代语言属于属于程序设计语言发展的第 1 阶段,简称为 1GL,又称为是低级语言。这代语言包括机器语言和汇编语言,在计算机诞生和发展初期使用的语言。

(1) 机器语言。机器语言是计算机能够唯一识别的语言,也称为面向机器的语言。机器语言程序是由二进制数字“0”、“1”组成的串,依赖具体的计算机系统,因而程序的通用性、移植性都很差。使用机器语言编写的程序,由于每条指令都对应计算机一个特定的基本动作,所以程序占用内存少、执行效率高。但是,使用机器语言编程对程序员来说是一件非常痛苦和困难的事情,一不小心,“0”写成了“1”,就可能会造成灾难性的后果。

(2) 汇编语言。为了解决使用机器语言编写应用程序所带来的困难,人们想到了使用助记符号来代替不容易记忆的机器指令,如用“ADD”代表加法。像这种使用助记符号来表示计算机指令的语言称为符号语言,也称汇编语言。在汇编语言中,每一条用符号来表示的汇编指令与计算机机器指令一一对应;记忆难度减少了,而且易于检查和修改程序错误,指令、数据的存放位置也可以由计算机自动分配。

用汇编语言编写的程序称为源程序,但是,计算机不认识这些符号。这就需要一个专门的媒介程序,专门负责将这些符号翻译成机器语言,像这种翻译程序被称为汇编程序。它们之间的关系如图 1-4 所示。

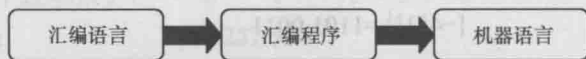


图 1-4 汇编语言的汇编过程

使用汇编语言编写计算机程序,需要程序员十分熟悉计算机系统的硬件结构,因此,从程序设计本身上来看仍然是低效率的、烦琐的。但是,与计算机硬件系统关系密切的特定场合,如对时空效率要求很高的系统核心程序以及实时控制程序等,汇编语言仍然是十分强有力的程

序设计工具。

2. 第2阶段

第2代语言和第3代语言属于程序设计语言发展的2阶段,简称为2GL,起始于20世纪50年代中期。汇编语言虽然在一定程度上克服了机器语言的不足,但是对程序员又提出了更高的要求。因此,人们意识到,应该设计一些接近于数学语言或人的自然语言的语言,同时又不依赖于计算机硬件,编写出来的程序能在所有机器上通用,这种语言被称为高级语言,是按照一定的语法规则,由表达各种意义的运算对象和运算方法构成。1954年,第一个完全脱离机器硬件的高级语言——FORTRAN问世了,继FORTRAN之后先后出现过几百种高级语言。当前使用较普遍的计算机编程语言主要有:C语言、Java语言、C#语言、Visual Basic、Delphi等。

(1) 从应用角度分类。从应用角度来看,高级语言可以分为基础语言、结构化语言和专用语言。

① 基础语言。基础语言也称通用语言。它历史悠久,流传很广,有大量的已开发的软件库,拥有众多的用户,为人们所熟悉和接受。像FORTRAN、COBOL、BASIC、ALGOL等都是基础语言。FORTRAN语言是目前国际上广为流行、也是使用得最早的一种高级语言,从20世纪90年代到现在,在工程与科学计算中占有重要地位,备受科技人员的欢迎。BASIC语言是在20世纪60年代初为适应分时系统而研制的一种交互式语言,可用于一般的数值计算与事务处理。BASIC语言结构简单,易学易用,并且具有交互能力,成为许多初学者学习程序设计的入门语言。

② 结构化语言。20世纪70年代以来,结构化程序设计和软件工程的思想日益为人们所接受和欣赏。在它们的影响下,先后出现了一些很有影响的结构化语言,这些结构化语言直接支持结构化的控制结构,具有很强的过程结构和数据结构能力。PASCAL、C、ADA语言就是结构化语言的突出代表。

③ 专用语言。专用语言是为某种特殊应用而专门设计的语言,通常具有特殊的语法形式。一般来说,这种语言的应用范围狭窄,移植性和可维护性不如结构化程序设计语言。目前使用的专业语言已有数百种,应用比较广泛的有APL语言、Forth语言、LISP语言。

(2) 从客观系统的描述分类。从描述客观系统来看,程序设计语言可以分为面向过程语言和面向对象语言。

① 面向过程语言。以“数据结构+算法”程序设计范式构成的程序设计语言,称为面向过程语言。面向过程的语言具有以下特点。

采用模块分解与功能抽象的方法,自顶向下,逐步求精。

按功能划分为若干个基本的功能模块,开成一个树状结构。各模块间的关系尽可能简单,功能上相对独立。第一个功能模块内部都是由顺序、选择和循环3种基本结构组成。

② 面向对象语言。以“对象+消息”程序设计范式构成的程序设计语言,称为面向对象语言。面向对象的语言的目标是实现软件的集成化,把相互联系的数据以及对数据的操作封装成通用的功能模块,各功能模块可以相互组合,完成具体的应用,还可以重复使用,而用户不必关心其功能是如何实现的。目前比较流行的面向对象语言有Delphi、Visual Basic、Java、C++等。

(3) 从需要的转换方式分类。用高级语言编写的程序称为源程序,计算机系统不能直接理解和执行,必须通过一个语言转换系统将其转换为计算机能够识别、理解的目标程序。按照转