

C 语言程序设计

C YUYAN CHENGXU SHEJI

胡燏 ◎ 专著



 知识产权出版社
全国百佳图书出版单位

C语言程序设计

胡燏 专著

 知识产权出版社
全国百佳图书出版单位

责任编辑：于晓菲

图书在版编目（CIP）数据

C 语言程序设计/胡燊专著. —北京：知识产权出版社，2012.9

ISBN 978-7-5130-1503-5

I. ①C… II. ①胡… III. ①C 语言—程序设计 IV.

①TP312

中国版本图书馆 CIP 数据核字（2012）第 207830 号

C 语言程序设计

胡 燊 专著

出版发行：知识产权出版社

社 址：北京市海淀区马甸南村 1 号

网 址：<http://www.ipph.cn>

发行电话：010-82000893 转 8101

责编电话：010-82000860 转 8363

印 刷：知识产权出版社电子制印中心

开 本：787mm×1092mm 1/16

版 次：2012 年 9 月第 1 版

字 数：480 千字

邮 编：100088

邮 箱：bjb@cnipr.com

传 真：010-82005070/82000893

责编邮箱：yuxiaofei@cnipr.com

经 销：新华书店及相关销售网点

印 张：23.5

印 次：2012 年 9 月第 1 次印刷

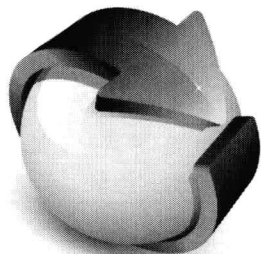
定 价：48.00 元

ISBN 978-7-5130-1503-5/TP·008 (4342)

出版版权专有 侵权必究

如有印装质量问题，本社负责调换。

前言



FOREWORD

本教材以“普通高等教育‘十二五’高职高专规划教材”编写精神为准绳,以本科、高职高专的学生为主要读者对象,以全国计算机等级考试(二级)(C语言方向)的知识点为框架,以“项目规划”为载体,将C语言程序设计的基础知识与实际的程序设计相融合,让读者在掌握基础知识的同时,将所学知识得以应用于程序编写过程中。本书不仅可以作为本科、高职高专院校的计算机网络技术、计算机应用技术、数控技术等专业《C语言程序设计》课程的教材,还可供各类培训、计算机等级考试人员参考使用。

本教材具有如下的特色:

(1)按照程序设计需要的单项能力组织教学项目,不拘泥于传统的C语言教学顺序,可以重新整合内容按项目教学。

(2)每个项目都包含:“项目学习目的”、“项目学习内容”和“项目学习实践”。

(3)每项语法内容如语句格式等,以单色底纹加以突出。按照“格式”、“功能”、“执行过程”,并以适当的“说明”或“注意”等进行表述。

(4)根据内容需要,安排适当的“验证实例”,结合最基本的运用,接受、理解、消化基本知识;安排适当的“应用实例”和“程序设计实践”等实际运用例题或综合运用例题,以加深对知识理解,进一步提高解决实际问题的能力。在个别教学项目中,安排了典型算法实例或一题多解等形式帮助学生拓展学习思路。

(5)设有“综合应用实例与程序调试”项目。

(6)每个实例都有“运行结果”。使用的所有程序,均经实际调试通过。

(7)整本教材从读者的学习习惯考虑,先介绍C语言的基础,如标识符、基本数据、运算符、表达式,这些基础以“C语言”的“单词”形式介绍给读者;然后再引入C语言的“语法”,如三大结构(顺序、选择、循环);接着介绍C语言的核心,如函数、数组、指针、结构体共用体;最后介绍C语言的“其他知识”,如文件、宏定义等。

教学内容参考学时如下表所示:

内容	项目总学时	各节学时
项目1 C语言概述	6	
1.1 程序设计语言简介		1
1.2 计算机中的进制及相互转换		3
1.3 C语言简介		2
习题一		

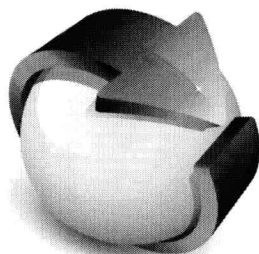
内容	项目总学时	各节学时
项目 2 C 语言的数据类型、运算符及表达式	14	
2.1 C 语言的数据类型结构		1
2.2 标识符、常量及变量		1
2.3 基本数据类型		4
2.4 运算符及表达式		6
2.5 编译预处理		2
习题二		
项目 3 顺序结构程序设计	9	
3.1 C 语句概述		1
3.2 赋值语句		1
3.3 C 语言中数据的输入输出		1
3.4 字符数据的输入输出		2
3.5 格式输入与输出		2
项目学习实践:顺序程序设计		2
习题三		
项目 4 选择结构程序设计	7	
4.1 if 语句		2
4.2 条件运算符		1
4.3 switch 语句		2
项目学习实践:选择程序设计		2
习题四		
项目 5 循环结构程序设计	9	
5.1 while 语句		1
5.2 do-while 语句		1
5.3 for 语句		2
5.4 循环的嵌套		2
5.5 break 语句与 continue 语句		1
项目学习实践:循环程序设计		2
习题五		
项目 6 数组	10	
6.1 一维数组		2
6.2 二维数组		2
6.3 字符数组		4
项目学习实践:数组应用		2
习题六		
项目 7 函数	14	
7.1 结构化程序设计和 C 语言程序组成		1
7.2 库函数		1
7.3 函数的定义和调用		2
7.4 函数参数和函数的值		1

内容	项目总学时	各节学时
7.5 函数调用时参数间的传递		2
7.6 函数的嵌套调用		2
7.7 函数的递归调用		2
7.8 变量的作用域、生存期和存储类别		1
项目学习实践:函数的应用		2
习题七		
项目 8 指针	11	
8.1 指针的基本概念		1
8.2 指针变量		1
8.3 数组与指针		2
8.4 指针与字符串		1
8.5 函数的指针		1
8.6 返回指针值的函数		1
8.7 指针数组		1
8.8 指向指针的指针		1
项目学习实践:指针的应用		2
习题八		
项目 9 结构体和共用体	13	
9.1 结构体		1
9.2 结构体数组		1
9.3 结构体和指针		1
9.4 动态存储分配		1
9.5 指针链表		4
9.6 共用体		2
9.7 枚举类型		2
9.8 类型定义符 typedef		1
习题九		
项目 10 文件概述	6	
10.1 文件		1
10.2 文件的打开和关闭		1
10.3 文件的读写		1
10.4 文件定位		1
10.5 文件检测函数和处理函数		1
项目学习实践:文件		1
习题十		
合计学时	99	99

本书由四川建筑职业技术学院胡燊独立编写完成,书中难免有些疏漏和不足之处,敬请有关专家和广大读者不吝指正。

编者
2012 年 3 月于四川德阳

目 录



CONTENTS

项目 1 C 语言概述	1
1.1 程序设计语言简介	2
1.1.1 机器语言	2
1.1.2 汇编语言	2
1.1.3 高级语言	3
1.2 计算机中的进制及相互转换	4
1.2.1 计算机中的几种进制	4
1.2.2 计算机中机器数的表示方法	5
1.2.3 计算机中进制之间的相互转换	6
1.3 C 语言简介	8
1.3.1 C 语言概述	8
1.3.2 C 语言的程序结构及特点	8
1.3.3 运行 C 程序的环境与步骤	11
习题一	15
项目 2 C 语言的数据类型、运算符及表达式	17
2.1 C 语言的数据类型结构	18
2.2 标识符、常量及变量	18
2.2.1 标识符	18
2.2.2 常量	20
2.2.3 变量	21
2.3 基本数据类型	22
2.3.1 整型数据	22
2.3.2 实型数据	26
2.3.3 字符型数据	28
2.3.4 变量赋初值	32
2.3.5 不同类型数据间的相互转换	33

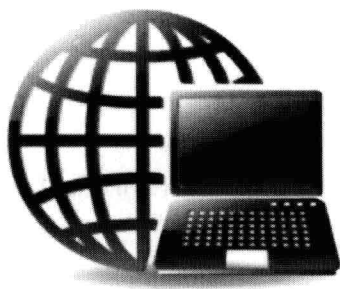
2.4	运算符及表达式	38
2.4.1	C语言运算符简介	38
2.4.2	算术运算符和算术表达式	39
2.4.3	赋值运算符和赋值表达式	39
2.4.4	自增、自减运算符	41
2.4.5	关系运算符和关系表达式	42
2.4.6	逻辑运算符和逻辑表达式	43
2.4.7	位运算符	44
2.4.8	逗号运算符和逗号表达式	48
2.4.9	常用运算符的优先级别汇总	49
2.5	编译预处理	49
2.5.1	宏定义	50
2.5.2	文件包含	56
2.5.3	条件编译	58
	习题二	61
项目3	顺序结构程序设计	65
3.1	C语句概述	66
3.2	赋值语句	67
3.3	C语言中数据的输入输出	68
3.4	字符数据的输入输出	69
3.4.1	putchar 函数	69
3.4.2	getchar 函数	69
3.4.3	gets 函数	70
3.4.4	puts 函数	70
3.5	格式输入与输出	71
3.5.1	printf 函数	71
3.5.2	scanf 函数	80
	习题三	84
项目4	选择结构程序设计	88
4.1	if 语句	89
4.1.1	单边选择结构	89
4.1.2	双边选择结构	91
4.1.3	多分支选择结构	92
4.1.4	嵌套 if 语句	95
4.2	条件运算符	98
4.3	switch 语句	99
	习题四	107

项目 5 循环结构程序设计	112
5.1 while 语句	113
5.2 do - while 语句	115
5.3 for 语句	119
5.4 循环的嵌套	123
5.5 break 语句与 continue 语句	124
5.5.1 break 语句	124
5.5.2 continue 语句	125
习题五	130
项目 6 数组	136
6.1 一维数组	137
6.1.1 一维数组的定义	137
6.1.2 一维数组的引用	138
6.1.3 一维数组的初始化	138
6.1.4 一维数组应用举例	139
6.2 二维数组	142
6.2.1 二维数组的定义	142
6.2.2 二维数组的引用	143
6.2.3 二维数组的初始化	144
6.2.4 二维数组程序举例	145
6.3 字符数组	147
6.3.1 字符数组的定义	148
6.3.2 字符数组的初始化	148
6.3.3 字符数组的引用	149
6.3.4 字符串和字符串结束标志	150
6.3.5 字符数组的输入输出	152
6.3.6 字符串处理函数	153
6.3.7 字符数组应用举例	157
习题六	162
项目 7 函数	168
7.1 结构化程序设计和 C 语言程序组成	169
7.1.1 结构化程序设计	169
7.1.2 C 语言程序的组成	169
7.1.3 函数分类	171
7.2 库函数	172
7.3 函数的定义和调用	174
7.3.1 函数的定义	174

7.3.2 函数的调用	175
7.4 函数参数和函数的值	181
7.4.1 形式参数和实际参数	181
7.4.2 函数的返回值	183
7.5 函数调用时参数间的传递	186
7.5.1 将变量、常量、数组元素作为参数时的传递	186
7.5.2 将数组名作为参数时的传递	188
7.6 函数的嵌套调用	191
7.7 函数的递归调用	196
7.8 变量的作用域、生存期和存储类别	200
7.8.1 局部变量和全局变量	200
7.8.2 变量的存在性和可见性	205
7.8.3 变量的存储类型	205
习题七	219
项目8 指针	225
8.1 指针的基本概念	226
8.2 指针变量	228
8.2.1 指针变量的定义	228
8.2.2 指针变量的赋值	229
8.2.3 指针变量的引用	230
8.2.4 将指针变量作为参数时的传递	232
8.3 数组与指针	236
8.3.1 指向一维数组元素的指针	236
8.3.2 通过指针引用一维数组的元素	237
8.3.3 用一维数组名作函数参数	242
8.3.4 指向多维数组的指针和指针变量	251
8.4 指针与字符串	256
8.4.1 字符串的表示形式	256
8.4.2 字符指针作函数参数	260
8.5 函数的指针	263
8.5.1 用函数指针变量调用函数	263
8.5.2 用指向函数的指针作函数参数	264
8.6 返回指针值的函数	267
8.7 指针数组	269
8.7.1 指针数组	269
8.7.2 main 函数的参数	272
8.8 指向指针的指针	274
8.8.1 指向指针的指针	274
习题八	283

项目 9 结构体和共用体	289
9.1 结构体	290
9.1.1 结构体的定义	290
9.1.2 结构体变量的定义	291
9.1.3 结构体变量的引用	293
9.1.4 结构体变量的赋值	294
9.2 结构体数组	296
9.2.1 结构体数组的定义	296
9.2.2 结构体数组的初始化	297
9.2.3 结构体数组元素的引用	298
9.2.4 结构体数组的应用举例	298
9.3 结构体和指针	300
9.3.1 指向结构体变量的指针	300
9.3.2 指向结构体数组的指针	302
9.3.3 用结构体变量和指向结构体的指针作函数参数	304
9.4 动态存储分配	305
9.5 指针链表	307
9.5.1 结构体构成的链表	307
9.5.2 在链表中插入结点	309
9.5.3 在链表中删除结点	310
9.6 共用体	311
9.6.1 共用体的概念	311
9.6.2 共用体变量的引用	312
9.7 枚举类型	315
9.7.1 枚举类型的定义	315
9.7.2 枚举类型变量的定义	315
9.7.3 枚举类型变量的赋值和使用	315
9.8 类型定义符 typedef	318
习题九	319
项目 10 文件	323
10.1 文件概述	324
10.1.1 文件的概念	324
10.1.2 文件的指针	326
10.1.3 文件的一般操作过程	326
10.2 文件的打开和关闭	327
10.2.1 文件的打开——fopen() 函数	327
10.2.2 文件的关闭——fclose() 函数	329
10.3 文件的读写	330

10.3.1	fgetc 函数和 fputc 函数	330
10.3.2	字符串读写函数 fgets 和 fputs	332
10.3.3	fread 函数和 fwrite 函数	334
10.3.4	fscanf 函数和 fprintf 函数	336
10.4	文件定位	337
10.4.1	rewind 函数	337
10.4.2	fseek 函数	338
10.4.3	ftell 函数	338
10.5	文件检测函数和处理函数	338
10.5.1	feof 函数	339
10.5.2	ferror 函数	339
10.5.3	clearerr 函数	339
10.5.4	exit 函数	339
	习题十	343
附录一	ASCII 码表	345
附录二	考试大纲	346
附录三	C 语言的运算符和结合性	351
附录四	C 语言常用语法	353
附录五	C 库函数	357
附录六	C 库文件	362
	参考文献	364



项目 1

C 语言概述



项目学习目的

学习 C 语言之前必须先掌握一些计算机的基础知识,才能保证在 C 语言的学习过程中,不仅能看懂 C 语言程序的结果,还能明白 C 语言程序编写的理由。本项目将提供学习 C 语言所必备的基础知识,补充和弥补读者在计算机领域中的知识空白。通过本项目的学习,既可以丰富读者的计算机知识,也可以为学习 C 语言程序设计做好充分的准备。

本项目的学习目标:

1. 计算机中程序设计语言的分类及其特点
2. 计算机中的进制及进制之间的相互转换
3. C 语言程序结构的特点
4. C 语言程序的运行环境



项目学习内容简述

21 世纪科技高速发展,人类以崭新的步伐迈入计算机时代。在计算机的世界里,要加深对计算机的认识了解,就必须做到“入乡随俗”。小时候,我们就开始学习汉语,于是我们可以和国人自由地交流;在步入学校的大门后,我们开始了英语的学习,于是我们可以和世界上其他国家讲英语的人快乐地沟通,今天我们要交流的对象是计算机,于是我们就需要学习计算机家族的语言,而 C 语言就是计算机家族中极为重要的一门语言。C 语言不但继承了低级语言的特点,而且还拥有了高级语言的特点,是计算机语言中可以控制计算机硬件的高级语言。接下来,就让我们携手步入计算机的语言世界吧。

1.1 程序设计语言简介

计算机在其发展过程中,先后经历了机器语言、汇编语言、高级语言的时代,下面就简要地介绍一下计算机语言的发展历程。

1.1.1 机器语言

机器语言是第一代计算机语言,是面向机器的低级语言,是直接能被计算机执行的语言,是0和1两种元素组成的机器指令的集合。

要使计算机按人的意图工作,就需要有一种与计算机之间能相互交流、相互理解的语言。机器语言就是人类和计算机都能读懂的语言。机器语言只能识别0和1两种状态,而这两种状态的表示极为简单和方便,如光电输入中,纸带有孔的地方代表1,纸带无孔的地方代表0;通电的状态代表1,断电的状态代表0;……。

直接用0和1组成的机器指令编写的程序,叫做机器语言的源程序,这是计算机唯一能够直接“听”懂的语言。但对于使用计算机的人类来说,机器语言具有难读、难记、难写、容易出错、通用性差等特点,是一门十分难懂的语言。为了提高人类与计算机之间的沟通和交流,人们势必要发明一种能让用户与计算机交流起来更方便的语言。因此,迫于形势的需求,计算机语言需要不断地发展,这必将诞生出新一代的计算机语言。

1.1.2 汇编语言

汇编语言是第二代计算机语言,也叫符号语言,它是用一些便于人们记忆的符号去表示计算机中二进制数的相关指令。下面是用汇编语言完成两个数加法运算的源代码。

LDA	A	(语句作用:将A放入寄存器中)
ADD	B	(语句作用:将B与寄存器中的A相加)
STA	C	(语句作用:将A与B相加之后的和放在C中)
PRINT	C	(语句作用:输出C的值)
STOP		(语句作用:程序停止)

这种用助记符号代替二进制代码指令的计算机语言,称为汇编语言,如LDA、ADD等都是汇编语言中的助记符或指令符号。用汇编语言编写的程序,称为汇编语言源程序,常常简称为汇编语言程序。

汇编语言,相对于机器语言来说更容易读、更容易记,但是汇编语言不能被计算机直接识别。因此要让计算机能够“读懂”汇编语言,就需要有一个“翻译官”,将汇编语言翻译成计算机能“读懂”的机器语言,这个翻译官就是“汇编程序”,把汇编语言程序翻译成机器能直接识别的机器语言程序的过程叫做“汇编”。计算机中的汇编过程如图1-1所示。

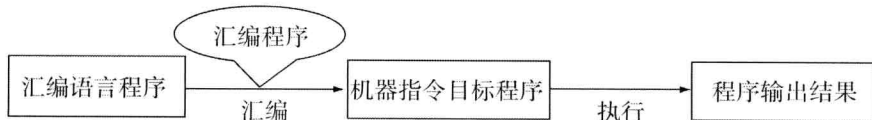


图 1-1 汇编过程

1.1.3 高级语言

高级语言是第三代计算机语言,也叫算法语言,它是一种更接近自然语言和数学表达式的计算机语言,由表示不同意义的“关键字”、“变量”、“表达式”等元素按照一定的计算机语言语法规则组合而成,实现一定的编程思想和编程目的。高级语言编制出来的程序方便了人们之间的交流沟通,易读易记,也便于修改、调试,大大提高了程序开发的效率和程序的可移植性。

高级语言的发展也经历了一系列的演变历程,由早期的 C、QBasic、Foxbase 等单一的纯代码编程的计算机语言,发展为目前的可视化图形界面的 VC、VB、VF、JAVA 等多种类多类别的计算机语言,目前比较流行的计算机语言有 C、VC、VB、VF、JAVA 等几类语言,接下来以 C 语言作为程序设计语言,编程实现两个数的加法运算。

```
#include "stdio. h"           (语句作用:包含头文件,以便使用相关函数)
main( )                      (语句作用:C 语言的主函数)
{ int x,y,sum;               (语句作用:定义 x,y,sum 为整型变量,以便使用)
  scanf( "% d,% d", &x,&y);   (语句作用:从键盘上输入任意求和的两个数 x,y)
  sum = x + y;               (语句作用:将 x 与 y 相加的和,放在 sum 中)
  printf( "sum = % d\n", sum); (语句作用:输出求和结果 sum 的值)
}
```

这样的程序代码就是利用高级语言实现编程算法的相应语句和代码。不难发现,高级语言离人们的理解越近,离计算机的理解也就越远了。因此和汇编语言一样,要将高级语言转变为计算机能读懂的机器语言,就需要像“汇编程序”一样的“翻译官”,将高级语言翻译成机器语言,一般有两种做法:编译方式和解释方式。

编译方式,即事先编好一个称为编译程序的机器指令程序,并放在计算机中,当有高级语言编写的源程序输入计算机时,编译程序便把源程序整个翻译成用机器指令表示的目标程序,然后执行该目标程序以便得到计算结果,如图 1-2 所示。

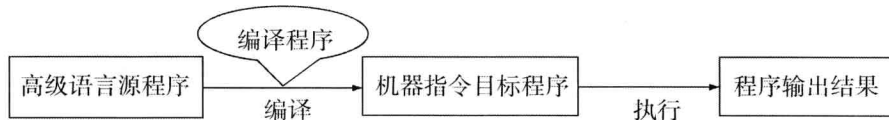


图 1-2 编译过程

解释方式,即事先编好一个称为解释程序的机器指令程序,并放在计算机中,当有高级语言编写的源程序输入计算机时,它会逐句地翻译,译出一句执行一句,即边解释边执行,而不像编译方式那样把源程序整个翻译成机器指令然后执行,这种方法较编译方式更花费时间,其过程如图 1-3 所示。



图 1-3 解释过程

通过编译或者解释过程,能将高级语言源程序翻译为机器指令程序,大大拉近了人与计算机的距离,从而降低了编程人员的工作量。因此可以说,有了高级语言的出现,人们能很

快地学会使用计算机,可以完全不顾机器指令,就能方便地使用计算机进行各种科学计算或事务处理等工作,高级语言的诞生是人类掌握计算机的一个重要里程碑。

在高级语言中,又常分为面向过程的程序设计语言和面向对象的程序设计语言。面向过程的程序设计语言是以过程为基本程序结构单位的程序设计语言,如 C、QBasic、Foxbase 等。面向对象的程序设计语言,也叫非过程化的语言,是一类以对象为基本程序结构单位的程序设计语言。面向对象的语言用于以对象为核心的程序设计中,对象是程序运行时的基本成分。面向对象的语言与之前的面向过程的语言相比,多提供了类、继承等成分,使程序的编写更加灵活和方便。

20 世纪 60 年代后,崛起的结构化程序设计,希望以规范的软件结构消除软件自身的复杂性和混乱性,解决“软件危机”的诸多问题。结构化程序设计在起初的使用阶段还顺风顺水,但随着硬件的发展,软件规模的逐步扩大,结构化程序设计在解决“软件危机”的问题上越来越显得“力不从心”。为了解决程序设计中的尴尬局面,在 20 世纪 90 年代初,面向对象的程序设计(Object Oriented Programming, OOP)应运而生,它的到来给软件产业带来了新的春天。面向对象的程序设计的主要特征有:数据隐藏,并发程序设计,重载与类型的多态性,重用的继承机制,数据抽象和抽象数据模型等。

面向对象的程序设计语言有近百种,其中具有代表性的有 C++、Object-C、Object Pascal、Actor、Smalltalk、Eiffel、Java 等。在这些语言中,Smalltalk、Eiffel、Actor 属于纯面向对象的语言,其余的称为混合型面向对象语言(即具有面向对象语言的特点,又有面向过程语言的特点,即属于非过程化语言,又属于过程化的高级语言)。纯面向对象语言比较重视实用性,混合型面向对象语言是在传统的过程式语言中增加面向对象的语言成分,因而在实用性上更具有优势。例如 C++ 语言,是在 C 语言基础上扩充的,对 C 语言兼容,在保留了 C 语言的所有语言机制,简洁高效的编程优点的基础上,增加了类和对象、继承、重载、虚函数、友元、内联等支持面向对象程序设计的机制,因此 C++ 顺理成章地称为目前应用最广泛的面向对象的程序设计语言。

1.2 计算机中的进制及相互转换

1.2.1 计算机中的几种进制

1. 计算机中进制的基本概念

所谓进位计数制是指按照进位的方法进行计数的数制,简称进位制。在计算机中主要采用的数制是二进制,同时在计算机中还存在八进制、十进制、十六进制的数据表示法。下面先来看看进制中的一些基本概念。

(1) 基数

计算机中的数制是以表示数值所用的符号个数来命名的,其中用来表明计数制允许选用的基本数码的个数称为基数,用 R 表示。例如:二进制数,每个数位上允许选用 0 和 1,共 2 个不同数码,因此它的基数 $R=2$;八进制数,每个数位上允许选用 0,1,2,3,4,5,6,7,共 8 个不同数码,因此它的基数 $R=8$;十六进制数,每个数位上允许选用 0,1,2,3,4,5,6,7,8,9,

A,B,C,D, E,F,共 16 个不同数码,它的基数 $R=16$ 。

(2) 权

在进位计数制中,一个数码处在数列中的不同位置,所代表的数值也是不一样的。每一个数位赋予的对应的数值称为位权,简称权。

权的大小是以基数 R 为底,数位的序号 i 为指数的整数次幂,用 i 表示数位的序号,用 R^i 表示数位的权。例如,二进制数 10010.01,其各个数位的权由左至右,分别为 2^4 、 2^3 、 2^2 、 2^1 、 2^0 、 2^{-1} 、 2^{-2} ;八进制数 1276.5,其各个数位的权由左至右,分别为 8^3 、 8^2 、 8^1 、 8^0 、 8^{-1} ;十进制数 543.21,其各个数位的权由左至右,分别为 10^2 、 10^1 、 10^0 、 10^{-1} 和 10^{-2} ;十六进制数 1EDB.CA,其各个数位的权由左到右,分别为 16^3 、 16^2 、 16^1 、 16^0 、 16^{-1} 、 16^{-2} 。

(3) 进位计数制的按权展开式

在进位计数制中,每个数位的数值等于该位数码与该位的权之乘积,用 K_i 表示第 i 位的系数,则该位的数值为 $K_i R^i$ 。任意进制的数都可以写成按权展开的多项式和的形式。

例如:二进制数 10010.01,按权展开的多项式形式为

$$10010.01 = 2^4 \times 1 + 2^3 \times 0 + 2^2 \times 0 + 2^1 \times 1 + 2^0 \times 0 + 2^{-1} \times 0 + 2^{-2} \times 1;$$

八进制数 1276.5,按权展开的多项式形式为

$$8^3 \times 1 + 8^2 \times 2 + 8^1 \times 7 + 8^0 \times 6 + 8^{-1} \times 5;$$

十进制数 543.21,按权展开的多项式形式为

$$10^2 \times 5 + 10^1 \times 4 + 10^0 \times 3 + 10^{-1} \times 2 + 10^{-2} \times 1;$$

十六进制数 1EDB.CA,按权展开的多项式形式为

$$16^3 \times 1 + 16^2 \times 14 + 16^1 \times 13 + 16^0 \times 11 + 16^{-1} \times 12 + 16^{-2} \times 10。$$

2. 计算机中常用的几种进制

在计算机中常用的进制有二进制、八进制、十进制和十六进制。二进制数的区分符常用字母 B 表示,八进制数的区分符常用字母 O 表示,十进制数的区分符常用字母 D 表示或不用字母表示,十六进制数的区分符常用字母 H 表示。

(1) 二进制(Binary System)

在二进制数中,数值按“逢二进一”的原则进行计数。其使用的数码为 0,1,二进制数的基数为“2”,权是以 2 为底的幂。

(2) 八进制(Octal Numerical System)

在八进制数中,数值按“逢八进一”的原则进行计数。其使用的数码为 0,1,2,3,4,5,6,7,八进制数的基数为“8”,权是以 8 为底的幂。

(3) 十进制(Decimal System)

在十进制数中,数值按“逢十进一”的原则进行计数。其使用的数码为 0,1,2,3,4,5,6,7,8,9,十进制数的基数为“10”,权是以 10 为底的幂。

(4) 十六进制(Hexadecimal System)

在十六进制数中,数值按“逢十六进一”的原则进行计数。其使用的数码为 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,十六进制数的基数为“16”,权是以 16 为底的幂。

1.2.2 计算机中机器数的表示方法

在计算机中机器数常用的表示方法有以下四种:原码、反码、补码和移码。