



普通高等教育 电气信息类 应用型规划教材

C语言程序设计

(第三版)

陆蓓 主编
易幼庆 楼永坚 汪志勤 副主编
郑宁 主审



科学出版社



免费提供电子教案

普通高等教育电气信息类应用型规划教材

内容简介

C 语言程序设计

(第三版)

陆蓓 主编
易幼庆 楼永坚 汪志勤 副主编
郑 宁 主审

科学出版社
北京

普通高等教育计算机类课程教材

内 容 简 介

本书主要介绍了 C 语言程序设计基础知识、基本数据类型与常用库函数、各种运算符与表达式、控制结构与语句、数组、函数、编译预处理、指针、结构、文件等,并对 C 语言的难点、重点和例子程序都做了详尽的阐述。

本书可作为高等院校相关专业高级语言程序设计课程的教材,也可作为计算机等级考试的教学用书。

图书在版编目(CIP)数据

C 语言程序设计/陆蓓主编. —3 版. —北京:科学出版社,2014

ISBN 978-7-03-012828-7

I. ①C… II. ①陆… 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(2004)第 003754 号

责任编辑:陈晓萍/责任校对:耿 耘

责任印制:吕春珉/封面设计:耕者设计工作室

科学出版社 出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

铭浩彩色印装有限公司 印刷

科学出版社发行 各地新华书店经销

*

2004 年 2 月第 一 版 2017 年 7 月第三十三次印刷

2009 年 9 月第 二 版 开本:787×1092 1/16

2014 年 8 月第 三 版 印张:19 1/2

字数:465 000

定价:39.00 元

(如有印装质量问题,我社负责调换〈铭浩〉)

销售部电话 010-62142126 编辑部电话 010-62138978-2009

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303

前言

C 语言是程序设计中非常活跃的高级语言之一。“C 语言程序设计”作为计算机文化基础、技术基础和应用基础三个层次中第二个层次的一门主要课程,是所有高等院校理工科学生的一门重要的必修课。以 C 为核心的 C++ 是面向对象程序设计的主要语言之一,对初学者而言, C 语言程序设计更是 C++ 的先修课程,是进一步学习 C++ 的基础。

本书自 2004 年 2 月出版以来,受到了各大专院校师生的欢迎。经过多年的教学实践,本书在内容设置、知识点涵盖面、内容叙述方式、章节安排顺序、例题分析形式以及习题选用等方面,都收到了许多读者的意见和建议。在此,表示衷心的感谢。

根据读者意见及教学实践,我们将教材的各章内容重新审视,融合各方面的意见,对教材的部分章节做了调整,并对部分章节的习题做了增删,现将其修订成第二版。第二版的教材分为两部分:第 1 章~第 5 章为程序设计基础篇;第 6 章~第 12 章为程序设计提高篇,各章内容安排简述如下。

第 1 章 简介计算机中的常用进制数、计算机中数值与非数值的表示、计算机的运算基础、C 语言程序的基本结构、开发一个 C 语言程序的过程。

第 2 章 介绍了 C 语言的基本数据类、常量与变量的概念;介绍了如何进行数据的输入/输出以及常用标准库函数的使用方法。

第 3 章 介绍了 C 语言的各种表达式及使用规则。C 语言中运算符丰富、表达式灵活,在本章中,从应用的角度出发,重点介绍各类运算符、表达式的使用,淡化了实际应用中很少出现的复杂计算,力求建立简洁易读的表达式。

第 4 章 介绍了结构化程序设计的三种基本结构,通过例子给出了表示算法的流程图,旨在帮助读者对流程图有一定的了解。重点介绍了顺序结构、分支结构和对应的语句;重点介绍了三种循环结构及对应的语句。较详细讨论了各类程序设计的例子,通过不同类型的例子,提高读者对这部分内容的掌握。第 4 章是程序设计的基础,也是后续学习的前提。而事实上程序设计能力不是通过例子看会的,重要的是实践。为了使读者很好地掌握该章内容,本章给出了丰富的习题帮助读者在实践中提高程序设计能力。

第 5 章 函数是 C 程序的主体。本章用较为简洁的方式介绍了函数定义、函数调用,力求使读者很快地能够掌握函数定义及调用的方法。在本章中,通过一个简单的例子来说明结构化程序设计的思想及 C 程序的结构。在解释函数的递归调用时,给出了函数递归调用执行过程的示意图,能较好地帮助读者理解递归执行的过程。

第 6 章 在数组章中,重点介绍了一维数组、二维数组的定义和引用,通过多个例子解释用数组编程的方法。在一维数组中介绍了基本的排序算法和查找算法。对于初学者来说,冒泡排序和顺序查找这两个基本算法应该掌握。字符串是本章介绍的另一个重点内容。介绍了字符串的概念、字符串的存储方式和字符串的输入输出。通过例子讲解了字符串的应用。在本章的最后,介绍了数组作为函数参数的方法及函数调用对实参数组的影响。

第 7 章 在编译预处理章节中,重点讨论了不带参数和带参数的宏的定义方法。解

释了带参数的宏(宏函数)与函数的区别。在该章中通过例子讨论了文件包含的应用。

第8章 本章主要讨论指针。首先通过存储单元、内存地址来阐明、解释指针的概念。介绍了指针变量定义及指针的基本操作。重点讨论了指针作为函数参数的作用及使用方法。在介绍了指针概念的基础上,讨论了数组与指针的关系,进一步阐明一维数组名作为函数参数的真正特点及意义。同时也讨论了二维数组的行指针作为函数的参数,与第6章数组作函数参数的情况作了比较。本章还介绍了指针数组及主要用途、二级指针的概念。通过例子介绍了带参数的主函数。最后介绍了返回指针的函数定义方法、指向函数的指针的用途。

第9章 介绍了结构体类型的定义及结构体变量定义。并通过例子给出了结构体变量的各种应用、结构体变量作为函数的参数用法和结构体数据作为函数的返回值。进一步讨论了结构体数组的用法、结构体指针的应用。最后介绍了链表的概念,单向链表的定义及链表的基本操作。

第10章 介绍了共用体、枚举和位运算。对初学者来讲,该章不是学习的主要内容,因此对内容的介绍没有展开讨论,相对其他几章来讲内容较简单。

第11章 文件在程序设计应用上是比较重要的内容。在该章中介绍了文件的概念、文件的打开和关闭操作。重点介绍了文本文件的数据读取,并通过例子介绍如何在程序中实现文件存取的操作。对二进制文件的数据块读/写也作了简单讨论。同时给出了文件操作的出错检测函数的用法。

第12章 在该章中根据教材内容,有选择性地设计了10个既有趣又易理解的综合性实验。实验根据难易程度分为3个简单题、5个中等题和2个稍难题,且10个综合性实验按难易度作了设计提示,使实践者具有很好的可操作性。避免了过去实践性题目大而难,对初学者来讲实际是无法上手操作的。

本书选编了较多的例子及习题,供老师在教学中根据需要进行选择,也可供学生自学提高。学生通过多读例子和动手上机编程,可开阔思路,提高程序设计的能力。本书中的个别章节或例题前标注了星号“*”,可作为初学者的选学内容。

相信修订后的第二版教材会得到更多读者的欢迎,同时我们也热忱期望读者继续提出宝贵的意见。

本书共分12章,其中第1、4、5、8章由杭州电子科技大学陆蓓教授编写;第2、3、6、9章由杭州电子科技大学易幼庆副教授编写;第7、10、11章由杭州电子科技大学楼永坚副教授编写;第12章由中国计量学院汪志勤副教授编写;附录由王卓玮(浙江大学)整理。全书由陆蓓教授担任主编并统稿,杭州电子科技大学郑宁教授担任主审。

杭州电子科技大学韩建平、饶万成、张平、张大兴、王小华、修晓杰等参与了本书的讨论并提出了许多宝贵的建议;陈法叶、曾智军、张犀、陆展、蒋玉兰参与了本书的文字录入、编辑及校对工作,在此一并表示衷心感谢。

由于作者水平、能力有限,加之时间仓促,书中错误或不妥之处,敬请广大读者批评、指正。我们的电子邮箱:lubei@hdu.edu.cn。

陆蓓

2014年8月

目 录

第 1 章 C 语言程序设计基础知识	1
1.1 计算机基础知识	2
1.1.1 计数制概念	2
1.1.2 数值与非数值的表示	4
1.1.3 计算机运算基础	7
1.1.4 程序与程序设计语言	8
1.1.5 程序执行	9
1.2 C 语言程序基本知识	9
1.2.1 简单 C 语言程序示例	9
1.2.2 C 语言程序的结构	11
1.2.3 C 语言的特点与良好的编程风格	12
1.3 使 C 语言程序在计算机上执行	14
1.3.1 实现 C 程序在机器上的执行过程	14
1.3.2 在 Microsoft Visual C++ 环境下开发 C 语言程序	15
本章小结	16
习题	17
第 2 章 基本数据类型与常用库函数	20
2.1 字符集与标识符	21
2.2 基本数据类型	22
2.2.1 整型数据	22
2.2.2 实型数据	23
2.2.3 字符型数据	23
2.3 常量与变量	23
2.3.1 常量	23
2.3.2 变量	26
2.4 输入/输出函数	27
2.4.1 格式化输入/输出函数	27
2.4.2 单个字符的输入/输出函数	34
2.5 常用函数	36
2.5.1 常用数学函数	37
2.5.2 常用字符函数	37
2.5.3 其他常用函数	38
本章小结	39
习题	39

第3章 表达式	42
3.1 算术表达式	43
3.1.1 算术表达式	43
3.1.2 自增、自减运算	44
3.2 赋值表达式	45
3.3 关系表达式	46
3.4 逻辑表达式	47
3.5 条件表达式	50
3.6 逗号表达式	51
3.7 多种类型混合运算	51
3.7.1 运算符优先级	51
3.7.2 运算符结合方向	52
3.7.3 混合运算的类型转换	53
本章小结	54
习题	54
第4章 控制结构与语句	59
4.1 结构化程序设计	60
4.1.1 程序的三种基本结构	60
4.1.2 程序设计过程	61
4.2 顺序结构	63
4.2.1 顺序结构	63
4.2.2 复合语句与空语句	64
4.3 选择结构	65
4.3.1 if 语句	65
4.3.2 switch 语句	71
4.3.3 用选择语句编程序	73
4.4 循环结构	75
4.4.1 while 语句	76
4.4.2 do-while 语句	78
4.4.3 for 语句	79
4.4.4 用循环编程序	82
4.5 转向语句	85
4.5.1 break 语句与 continue 语句	85
4.5.2 goto 语句	89
本章小结	89
习题	89
第5章 函数	96
5.1 函数定义	97
5.1.1 函数概述	97

5.1.2	函数定义	98
5.2	函数调用	99
5.2.1	函数调用	99
5.2.2	传值调用的特点	101
5.2.3	函数调用的方式	103
5.2.4	用函数编程序	103
5.3	函数与 C 程序结构	104
5.3.1	C 程序结构	104
5.3.2	函数的嵌套调用	107
5.3.3	函数递归调用	108
*5.3.4	参数求值顺序	113
5.4	函数与变量	114
5.4.1	全局变量和局部变量	114
5.4.2	变量的生命期与变量的存储类别	117
	本章小结	119
	习题	120
第 6 章	数组	127
6.1	一维数组	128
6.1.1	一维数组定义	128
6.1.2	一维数组引用	129
6.1.3	用一维数组编写程序	130
6.1.4	查找与排序	132
6.2	二维数组	137
6.2.1	二维数组定义	138
6.2.2	二维数组引用	139
6.2.3	用二维数组编写程序	141
6.3	字符串	144
6.3.1	字符串的存储	144
6.3.2	字符串的输入/输出	145
6.3.3	字符串应用	147
6.3.4	多字符串处理	149
6.4	数组与函数	151
6.4.1	数组元素作为函数参数	151
6.4.2	数组作为函数参数	152
6.4.3	字符串作为函数参数	155
	本章小结	156
	习题	157
第 7 章	编译预处理	167
7.1	宏定义	168

7.1.1 不带参数的宏定义	168
7.1.2 带参数的宏定义	170
*7.1.3 带参数的宏与函数的区别	170
7.2 文件包含	171
*7.3 条件编译	174
本章小结	177
习题	177
第8章 指针	179
8.1 指针的基本知识	180
8.1.1 内存单元、内存地址及指针	180
8.1.2 指针变量定义	181
8.1.3 指针基本操作	181
8.1.4 指针作函数的参数	183
8.2 指针与数组	186
8.2.1 指针与一维数组	186
8.2.2 一维数组作函数的参数	187
8.2.3 指针在数组上的运算	189
8.2.4 指针与二维数组	190
*8.2.5 行指针作函数参数	191
8.3 指针与字符串	194
8.3.1 字符串指针	194
8.3.2 字符串指针作函数参数	197
8.3.3 常用字符串函数	198
8.4 指针数组与多级指针	201
8.4.1 指针数组与多字符串	201
*8.4.2 指向指针的指针	204
*8.4.3 main 函数的参数	207
8.5 指针与函数	208
8.5.1 指针作为函数的返回值	208
8.5.2 指向函数的指针	210
本章小结	213
习题	213
第9章 结构体	220
9.1 结构体概念	221
9.1.1 结构体类型定义	221
9.1.2 结构体变量定义	222
9.1.3 结构体变量引用	224
9.1.4 结构体变量作函数参数	226
9.1.5 结构体数据作为函数返回值	227

9.2 结构体数组	228
9.2.1 结构体数组定义	229
9.2.2 结构体数组引用	230
9.3 结构体指针	231
9.3.1 结构体指针概念	231
9.3.2 结构体指针应用	232
9.3.3 结构体指针作函数参数	233
*9.4 单向链表	235
9.4.1 链表的概念	235
9.4.2 单向链表的定义	236
9.4.3 动态存储分配库函数	236
9.4.4 单向链表的基本操作	237
本章小结	245
习题	246
*第 10 章 共用体、枚举和位运算	251
*10.1 共用体	252
*10.2 枚举	255
10.3 位运算与位段	257
10.3.1 位运算符	258
10.3.2 位运算符的优先级	263
*10.3.3 位段	264
本章小结	265
习题	265
第 11 章 文件	268
11.1 文件概述	269
11.1.1 文本文件和二进制文件	270
11.1.2 文件类型指针	271
11.2 打开文件与关闭文件	272
11.2.1 打开文件	272
11.2.2 关闭文件	273
11.3 文本文件读写	273
11.3.1 单个字符读/写	274
11.3.2 格式化的数据读/写	278
11.3.3 用文件编程	279
*11.4 二进制文件的块数据读/写	282
*11.5 文件定位函数	283
*11.6 文件操作的出错检测	285
本章小结	286
习题	287

第 12 章 综合性实验	290
实验 1 判断某日是一年中的第几天	291
实验 2 万年历	291
实验 3 猜数字	291
实验 4 计算器	292
实验 5 影碟信息管理系统	292
实验 6 酒店信息管理系统	293
实验 7 图书信息管理系统	293
实验 8 销售信息管理系统	294
实验 9 练习打字程序	294
实验 10 进制转换程序	295
附录	296
附录 1 C 语言中的关键字	296
附录 2 字符与 ASCII 码对照表	296
附录 3 运算符的优先级与结合性	298
附录 4 常用库函数	299
主要参考文献	302

第 1 章 C 语言程序设计基础知识

本章学习目标

- ◆ 掌握计算机运算基础
- ◆ 理解计算机程序基本概念
- ◆ 熟悉 C 语言基础知识
- ◆ 了解 C 语言程序结构
- ◆ 学会在 Visual C++ 环境下开发 C 程序

计算机程序设计语言是程序员与计算机之间交流的语言,它遵循一定的规则和形式,构成程序的实现工具。C 语言于 1972 年至 1973 年间诞生于美国贝尔实验室,由 Denis M. Ritchie 创建,并用 C 语言重写了著名的 UNIX 操作系统。之后,C 语言发展成为通用的程序设计语言。1983 年美国国家标准化协会 ANSI 为了使 C 语言得到更好更快的发展,成立了一个委员会来制定 C 语言标准,称为 ANSI C。随着计算机技术的迅速发展和广泛应用,C 语言在计算机软件开发中起着越来越重要的作用,它既适合编写各种计算机系统软件,又适合编写各种应用程序,已成为世界上广为流行的计算机程序设计语言之一。

1.1 计算机基础知识

1.1.1 计数制概念

在计算机领域,最常用到的是二进制,这是因为计算机是由千千万万个电子元件(如电容、电感、三极管等)组成,这些电子元件一般都是只有两种稳定的工作状态(如三极管的截止和导通),用高、低两个电位表示“1”和“0”在物理上是最容易实现。因此计算机内部采用二进制计数系统,二进制是计算机运算的基础,但二进制的书写一般比较长,而且容易出错。因此除了二进制外,为了便于书写,计算机中还常常用到八进制和十六进制。一般用户与计算机打交道并不直接使用二进制数,而是使用十进制数(或八进制、十六进制数),然后由计算机自动转换为二进制数。了解二进制及各种进制间的转换以及信息编码的概念,是学习程序设计的前提。

1. 数制

R 进位计数制的主要特点就是逢 R 进一。逢十进一的是十进制,逢八进一的是八进制,逢二进一的是二进制,逢十六进一的是十六进制。表 1.1 是四种进制对照表。

进位计数制由三个基本要素组成:数码、基数和位权。数码:进位计数制中用来计数的符号。如,十进制的数码有 0, 1, 2, ..., 9; 二进制数码有 0, 1; 十六进制数码有 0, 1, 2, ..., 9, A, B, ..., F。基数:进位计数制中采用数码的个数。如,十进制的基数是 10,二进制的基数是 2。位权:在进位计数制中,数码在数中不同的位置上,其位权值不同。某个数码的位权的计算方法是:以基数为底,以数码所在数中的位置为指数的整数次幂,即为该数码的位权。例如,十进制 356.24 从左到右各个数码的位权分别是 10^2 、 10^1 、 10^0 、 10^{-1} 、 10^{-2} 。

因此,用任何一种进位计数制都可以表示为它的各个数码与位权乘积之和。假设任意数值 N 用 R 进制数来表示, $N = D_{m-1}D_{m-2} \cdots D_0.D_{-1}D_{-2} \cdots D_{-k}$ (其中, D_i 为数码,有 m 位整数和 k 位小数),则数值 N 的实际值可展开表示为

$$N = D_{m-1} \times R^{m-1} + D_{m-2} \times R^{m-2} + \cdots + D_0 \times R^0 + D_{-1} \times R^{-1} + D_{-2} \times R^{-2} + \cdots + D_{-k} \times R^{-k} \quad (1.1)$$

我们最熟悉的是十进制,它的基数为 10,它的数码是 0~9。十进制数 1635.458 就可以写为

$$1635.458 = 1 \times 10^3 + 6 \times 10^2 + 3 \times 10^1 + 5 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2} + 8 \times 10^{-3}$$

表 1.1 四种进制对照表

十进制	二进制	八进制	十六进制
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

2. 数制转换

(1) 十进制转换为二、八、十六进制

整数部分和小数部分须分别遵守不同的转换规则。假设将十进制数转换为R进制数：

整数部分：除以R取余法，即整数部分不断除以R取余数，直到商为0为止，最先得到的余数为R进制数最低位，最后得到的余数为最高位。

小数部分：乘以R取整法，即小数部分不断乘以R取整数，直到积为0或达到有效精度为止，最先得到的整数为R进制数最高位（最靠近小数点），最后得到的整数为最低位。

例如，将 $(114.35)_{10}$ 转化为二进制数，误差不超过10进制数的0.1。

按照除以2取余法，可得 $(114)_{10} = (1110010)_2$ ，即

		0.35	取整数
$2 \overline{) 114}$	余数	$\times 2$	
$2 \overline{) 57}$	0 (最低位)	0.70	0 (最高位)
$2 \overline{) 28}$	1	$\times 2$	
$2 \overline{) 14}$	0	1.40	1
$2 \overline{) 7}$	0	$\times 2$	
$2 \overline{) 3}$	1	0.80	0
$2 \overline{) 1}$	1	$\times 2$	
$2 \overline{) 0}$	1 (最高位)	1.60	1 (最低位)

对于小数由于要求误差不超过十进制数的0.1，而 $1/2^4 = 1/16 < 0.1$ ，因此取二

进制小数位数 4 位即可, 按照乘以 2 取整法, 可得 $(0.35)_{10} = (0.0101)_2$; 所以, $(114.35)_{10} = (1110010.0101)_2$ 。

(2) 二、八、十六进制转换为十进制

根据式 (1.1), 按位权展开求和, 即将每位数码乘以各自的权值并累加。

例如,

$$\begin{aligned}(1001.1)_2 &= 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} \\ &= 8 + 1 + 0.5\end{aligned}$$

$$= (9.5)_{10}$$

$$\begin{aligned}(345.73)_8 &= 3 \times 8^2 + 4 \times 8^1 + 5 \times 8^0 + 7 \times 8^{-1} + 3 \times 8^{-2} \\ &= 192 + 32 + 5 + 0.875 + 0.046875\end{aligned}$$

$$= (229.921875)_{10}$$

$$\begin{aligned}(A3B.E5)_{16} &= 10 \times 16^2 + 3 \times 16^1 + 11 \times 16^0 + 14 \times 16^{-1} + 5 \times 16^{-2} \\ &= 2560 + 48 + 11 + 0.875 + 0.01953125\end{aligned}$$

$$= (2619.89453125)_{10}$$

(3) 二进制转换为八、十六进制

因为 $2^3=8$, $2^4=16$, 所以 3 位二进制数对应 1 位八进制数, 4 位二进制数对应 1 位十六进制数。二进制数转换为八、十六进制数比转换为十进制数容易得多, 因此常用八、十六进制数来表示二进制数。

将二进制数以小数点为中心分别向两边分组, 转换成八 (或十六) 进制数, 每 3 (或 4) 位为一组, 不够位数在两边加 0 补足, 然后将每组二进制数化成八 (或十六) 进制数即可。

例如, 将二进制数 1001101101.11001 分别转换为八、十六进制数:

$$(\underline{001} \underline{001} \underline{101} \underline{101} . \underline{110} \underline{010})_2 = (1155.62)_8 \quad (\text{注意: 在两边补零})$$

$$1 \quad 1 \quad 5 \quad 5 \quad 6 \quad 2$$

$$(\underline{0010} \underline{0110} \underline{1101} . \underline{1100} \underline{1000})_2 = (26D.C8)_{16}$$

$$2 \quad 6 \quad D \quad C \quad 8$$

(4) 八、十六进制转换为二进制

将每位八 (或十六) 进制数展开为 3 (或 4) 位二进制数, 不够位数在左边加 0 补足。

例如,

$$\begin{aligned}(631.02)_8 &= (\underline{110} \underline{011} \underline{001} . \underline{000} \underline{010})_2 \\ &\quad 6 \quad 3 \quad 1 \quad 0 \quad 2\end{aligned}$$

$$\begin{aligned}(23B.E5)_{16} &= (\underline{0010} \underline{0011} \underline{1011} . \underline{1110} \underline{0101})_2 \\ &\quad 2 \quad 3 \quad B \quad E \quad 5\end{aligned}$$

1.1.2 数值与非数值的表示

1. 原码、反码、补码的表示

在计算机中参与运算的数据有两种: 无符号数和有符号数。对于无符号数, 所有二进制数据位数均用来表示数值本身, 没有正负之分; 而对于有符号数, 则其二进制数据

位,除了必须表明其数值大小外,还必须保留正负符号的位置,表明数值的正负,通常用二进制数据位的最高位表示数的符号,分别用0和1表示数的正和负。因此,在计算机中,机器字长相同时,无符号数和有符号数其数据的表示范围是不同的。例如,当机器字长为16位时,无符号数的数据表示范围是0~65535,而有符号数补码的数据表示范围是-32768~+32767。

计算机中为了便于运算,有符号数有三种表示方法:原码、反码、补码。

(1) 原码

最高位表示数的符号,其他位表示数值的二进制数。符号位规定用0表示正,用1表示负。

例如, $[+8]_{\text{原}} = 00001000\text{B}$ $[-8]_{\text{原}} = 10001000\text{B}$

(2) 反码

正数的反码和原码完全一样。负数的反码是由其原码的数值部分求反(即符号位不变,其他数值位由0变为1,1变为0而得到的)。

例如, $[+8]_{\text{反}} = 00001000\text{B}$ $[-8]_{\text{反}} = 11110111\text{B}$

注意:0的反码有两种形式, $[+0]_{\text{反}} = 00000000$, $[-0]_{\text{反}} = 11111111$ 。

(3) 补码

补码是计算机中带符号数的实用表示方法,计算机中通常用补码进行数值运算。正数的补码与原码和反码是一样的。负数的补码可由其反码的末位加1得到,即负数的补码是对其原码除符号位外各数值位求反,并在末位加1而得到的。

$[+8]_{\text{补}} = 00001000\text{B}$ $[-8]_{\text{补}} = 11111000\text{B}$

注意:0的补码只有一种形式, $[+0]_{\text{补}} = [-0]_{\text{补}} = 00000000$ 。

上面对原码、补码、反码的举例都假设以机器字长8位。

2. 定点数和浮点数的表示

计算机中没有专门的部件用来表示小数点,因此,在机器数中,小数点及其位置是隐含规定的。小数点的隐含规定方法有两种:定点和浮点。

定点数,其小数点的位置是固定不变的,可以分为定点小数和定点整数两种。定点整数用于表示纯整数,其小数点位置隐含固定在最低位之后。定点小数是指小数点准确固定在数据某一个位置上的小数,一般都把小数点固定在最高数据位的左边,称为定点纯小数(实际上小数位并不占用空间,默认在该位置)。如图1.1所示的定点数格式。



图1.1 定点数格式

浮点数,可以方便地表示实数,其小数点的位置由其阶码值规定,因此是浮动的。浮点数N由三部分来决定其数值大小:阶码E、尾数M和阶码的底R(基数)。如图1.2所示的浮点数典型格式。

任何一个二进制数N,可以表示为: $N = M \times R^E$ 。

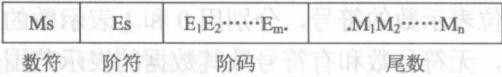


图 1.2 浮点数的典型格式

浮点数中，尾数 M 是一个规格化的定点纯小数，它决定了浮点数有效数值的精度，尾数的符号表示浮点数的正负，称为数符。尾数一般采用原码或补码表示。阶码 E 是一个定点纯整数，阶码的数值大小决定了该浮点数实际小数点的位置，阶码的符号称为阶符。

例如，假设 16 位虚拟机中，阶码占 5 位，尾数占 9 位，数符、阶码各占 1 位，对于实数 28.625 的浮点数表示为 $N=(11100.101)_2=(0.11100101)\times 2^5$ 则该数在 16 位虚拟机中的浮点数表示：



在浮点数的机器表示中，数符、阶码各占一位表示浮点数和阶码的正负，阶码占的位数决定了数的范围，尾数占的位数决定了数的精度。

3. 非数值的表示

在计算机中，处理非数值的文字和其他符号时，要对文字和符号进行编码，用数字编码来表示文字和符号。字符编码就是规定用二进制数表示文字和符号的方法。

ASCII 码是一种字符编码，为美国标准信息交换码，已被国际标准化组织批准为国际标准，它适用于所有西文字母，已在全世界通用。表 1.2 是 ASCII 字符编码表。

表 1.2 ASCII 字符编码表

b6 b5 b4 b3 b2 b1 b0	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	,	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL