




普通高等教育“十三五”公修课规划教材

C 语言程序设计

C YUYAN CHENGXU SHEJI

陈平 主编



 河南科学技术出版社

普通高等教育“十三五”公修课规划教材

C 语言程序设计

主 编 陈 平

副主编 张 霞 李 政

河南科学技术出版社

·郑州·



内容提要

本书编写以能力培养为核心,以示例为主要特色,以具体示例促进理论知识的学习和运用。全书共分13章,介绍了C语言程序设计的基础知识及C语言程序设计的开发环境,C语言的数据类型及C语言程序的基本结构,数组、结构体和共用体等几种常用的构造类型及C语言的特色数据类型,编译预处理、文件操作和位运算等内容。每章都附有大量的示例来对基础知识进行讲解,并在课后配有多种类型的习题。书末配有C语言常用关键字、基本ASCII码对照表、C语言运算符的优先级与结合性3个附录,以方便读者查阅。

本书既可作为高等职业院校相关专业C语言程序设计课程的教材,还可供参加计算机等级考试及C语言程序设计的初学者自学使用。

图书在版编目(CIP)数据

C语言程序设计/陈平主编. —郑州:河南科学技术出版社,2018.12
普通高等教育“十三五”公修课规划教材
ISBN 978-7-5349-9390-9

I. ①C… II. ①陈… III. ①C语言-程序设计-高等学校-教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2018)第289603号

出版发行:河南科学技术出版社

地址:郑州市经五路66号 邮编:450002

电话:(0371) 65788859 65788607

网址:www.hnstp.cn

策划编辑:徐素军

责任编辑:徐素军

责任校对:马晓灿

封面设计:张伟

责任印制:张艳芳

印刷:河南安泰彩印有限公司

经销:全国新华书店

开本:787 mm×1092 mm 1/16 印张:22.5 字数:550字

版次:2018年12月第1版 2018年12月第1次印刷

定价:48.00元

如发现印、装质量问题,影响阅读,请与出版社联系并调换。

前 言

C 语言是目前广泛流行的计算机高级编程语言，因语法简洁、使用灵活且不失强大的功能而受到编程人员的青睐。它既适合作为系统描述语言，又可用于编写系统软件，还可用来编写应用软件。因此，大部分高等院校都把 C 语言作为学生程序设计入门的一种语言，同时，学习 C 语言也为进一步学习 C++ 及其他语言奠定基础。

本书作为 C 语言程序设计的基础教材，共分 13 章及 3 个附录。主要内容包括：C 语言程序设计入门，数据类型、运算符及表达式，顺序结构程序设计，选择结构程序设计，循环结构程序设计，以及数组、函数、指针、结构体、共用体与枚举、预处理功能、文件、位运算。3 个附录分别是：C 语言的 32 个关键字、基本 ASCII 码对照表、C 语言运算符的优先级与结合性。本书注重基础、突出应用，每章均配有较多的程序案例，以便读者能够综合运用本章所学知识提高实际编程能力。

本书具有以下特色：

1. 易于教学和自学，适合初学者

本书按照循序渐进的原则组织内容，尽可能地通过简明的叙述来阐明基本的概念。书中难点和重点，通过丰富的例题来进行详细的解释。力求做到语言通俗、概念清晰、易学实用，使读者易于接受。

2. 注重基础内容，突出实用性

C 语言内容庞杂，细节特别多。为便于初学者掌握最重要的、最实用的内容，本书对于较少使用的一些内容进行了删减，在保证 C 语言完整性的同时，力求做到突出重点，强化基础，使学生尽快掌握 C 语言的基础知识。

3. 强化编程思想，突出应用性

本书在编写上突出强化编程思想，通过例题和习题去不断地强化。由于程序设计语言是实践性很强的课程，因此建议读者多上机实践，在掌握基本语法知识的同时，尽快掌握 C 语言的编程方法，提高调试程序的能力。

本书介绍的基本内容和程序具有通用性，基本上适应各种 C 语言编译环境，但不同的 C 语言环境由于支持的 C 语言标准不同，所以可能会有一些差别。本书中所有程序及示例均在 VC6.0 和 Visual Studio 2008 下调试通过。

本书体系合理、概念清晰、示例丰富，注重基础、突出应用，以便读者能够综合运用每章所学知识提高实际编程能力。

本书由河南职业技术学院陈平任主编，河南职业技术学院张霞、深圳职业技术学院李政任副主编。具体编写分工为：陈平编写第 1~4 章，张霞编写第 5~9 章，李政编写第 10~13 章。全书由陈平统稿定稿。

由于编者的水平有限，书中可能有不妥之处，恳望读者、同行和专家不吝指正。

编 者
2018 年 10 月

目 录

第 1 章 C 语言程序设计入门.....	(1)
1.1 计算机基础	(1)
1.1.1 计算机系统组成	(1)
1.1.2 计算机系统工作原理	(2)
1.1.3 计算机语言	(2)
1.1.4 数制及其转换	(3)
1.2 C 语言概述	(6)
1.2.1 C 语言的发展	(6)
1.2.2 C 语言的特性	(7)
1.2.3 C 语言程序的基本结构	(7)
1.2.4 C 语言程序的编译和运行	(12)
1.3 Visual studio 使用简介	(13)
1.3.1 创建新项目	(13)
1.3.2 新建 C 语言源文件	(15)
1.3.3 输入代码	(15)
1.3.4 配置编译输出项	(16)
1.3.5 编译链接	(17)
1.3.6 运行程序	(17)
1.3.7 Visual Studio 的编译器和链接器.....	(18)
1.4 小结.....	(18)
习题.....	(19)
第 2 章 数据类型、运算符和表达式	(21)
2.1 字符集、标识符与关键字.....	(21)
2.1.1 字符集	(21)
2.1.2 标识符	(21)
2.1.3 关键字	(22)
2.2 C 语言数据类型概述.....	(22)
2.3 变量与常量.....	(23)
2.3.1 常量	(23)

2.3.2	变量	(24)
2.4	基本数据类型	(26)
2.4.1	整数类型	(26)
2.4.2	浮点数类型	(30)
2.4.3	字符类型	(32)
2.5	运算符和表达式	(34)
2.5.1	运算符和表达式概述	(34)
2.5.2	运算符的优先级与结合性	(35)
2.5.3	算术运算符	(35)
2.5.4	赋值运算符	(37)
2.5.5	逗号运算符	(39)
2.5.6	不同类型数值型数据间的混合运算	(40)
2.6	小结	(42)
	习题	(42)
第3章	顺序结构程序设计	(44)
3.1	结构化程序设计简介	(44)
3.2	算法及其表示	(45)
3.2.1	算法	(45)
3.2.2	算法的表示	(46)
3.3	C 语言语句概述	(47)
3.4	输入/输出、流和缓冲区	(48)
3.4.1	输入/输出	(48)
3.4.2	流	(49)
3.4.3	缓冲区	(49)
3.5	数据的输入与输出	(50)
3.5.1	字符输出 putchar()	(50)
3.5.2	字符输入 getchar()	(51)
3.5.3	格式化输出 printf()	(53)
3.5.4	格式化输入 scanf()	(56)
3.6	综合示例及分析	(60)
3.7	小结	(64)
	习题	(64)
第4章	选择结构程序设计	(67)
4.1	选择结构与条件判断	(67)
4.2	关系运算符与关系表达式	(67)
4.2.1	关系运算符及其优先次序	(67)
4.2.2	关系表达式	(68)
4.3	逻辑运算符与逻辑表达式	(69)
4.3.1	逻辑运算符	(69)

4.3.2	逻辑表达式	(70)
4.4	条件运算符与条件表达式	(71)
4.5	用 if 语句实现选择结构	(72)
4.5.1	单分支 if 语句	(72)
4.5.2	双分支 if 语句	(74)
4.5.3	多分支 if 语句	(77)
4.6	if 选择结构的嵌套	(81)
4.7	用 switch 语句实现多分支选择结构	(83)
4.8	综合示例及分析	(88)
4.9	小结	(92)
	习题	(92)
第 5 章	循环结构程序设计	(97)
5.1	goto 语句以及用 goto 语句构成的循环	(97)
5.2	while 循环语句	(98)
5.2.1	while 循环语句定义	(98)
5.2.2	while 循环应用示例	(99)
5.3	do...while 循环语句	(100)
5.3.1	do...while 循环语句定义	(100)
5.3.2	do...while 循环应用示例	(101)
5.3.3	while 和 do...while 循环比较	(102)
5.4	for 循环语句	(103)
5.4.1	for 语句定义	(103)
5.4.2	for 语句注意事项	(104)
5.4.3	for 循环应用示例	(106)
5.5	循环嵌套	(110)
5.6	几种循环的比较	(112)
5.7	循环状态改变语句	(112)
5.7.1	break 语句	(112)
5.7.2	continue 语句	(114)
5.8	综合示例及分析	(115)
5.9	小结	(123)
	习题	(123)
第 6 章	数组	(128)
6.1	一维数组	(128)
6.1.1	一维数组的定义	(128)
6.1.2	一维数组元素的引用	(129)
6.1.3	一维数组的初始化	(130)
6.1.4	一维数组应用示例	(131)
6.2	二维数组	(136)

6.2.1	二维数组的定义	(136)
6.2.2	二维数组元素的引用	(137)
6.2.3	二维数组的初始化	(138)
6.2.4	二维数组应用示例	(140)
6.3	字符数组	(144)
6.3.1	字符数组的定义	(144)
6.3.2	字符数组的初始化	(144)
6.3.3	字符数组的引用	(145)
6.3.4	字符串与字符串结束标志	(146)
6.3.5	字符数组的输入和输出	(147)
6.3.6	字符串处理函数	(150)
6.3.7	字符数组及字符串示例	(153)
6.4	综合示例及分析	(154)
6.5	小结	(162)
	习题	(162)
第7章	函数	(165)
7.1	函数概述	(165)
7.2	函数的定义	(167)
7.2.1	函数的定义形式	(167)
7.2.2	函数的参数	(170)
7.2.3	函数的返回值	(172)
7.2.4	函数声明	(173)
7.3	函数的调用	(174)
7.3.1	函数调用的方式	(174)
7.3.2	函数间的数据传递	(175)
7.3.3	函数的嵌套调用	(181)
7.3.4	函数的递归调用	(182)
7.4	变量的存储属性	(184)
7.4.1	局部变量和全局变量	(184)
7.4.2	变量存储方式和生存期	(187)
7.5	其他函数	(192)
7.5.1	main() 函数	(192)
7.5.2	库函数	(195)
7.5.3	内部函数和外部函数	(195)
7.6	综合示例及分析	(196)
7.7	小结	(200)
	习题	(200)
第8章	指针	(204)
8.1	指针与指针变量	(204)

8.1.1	指针的概念	(204)
8.1.2	指针变量的概念	(205)
8.1.3	指针的引用和运算	(205)
8.1.4	指针变量的定义和赋值	(206)
8.1.5	指针变量的引用	(208)
8.2	指针与函数	(208)
8.2.1	指针作函数的参数	(209)
8.2.2	函数返回指针	(211)
8.2.3	指向函数的指针	(211)
8.3	指针与数组	(213)
8.3.1	通过指针引用一维数组中的元素	(213)
8.3.2	指针基本运算	(215)
8.3.3	通过指针引用二维数组中的元素	(218)
8.4	指针与字符串	(220)
8.4.1	字符数组与字符指针	(220)
8.4.2	常见字符串操作	(220)
8.5	指针数组	(224)
8.5.1	指针数组与数组指针	(224)
8.5.2	main () 函数的参数	(227)
8.6	多级指针	(229)
8.7	综合示例及分析	(230)
8.8	小结	(235)
	习题	(236)
第9章	结构体	(241)
9.1	结构体类型定义	(241)
9.2	结构体类型变量定义	(242)
9.3	结构体成员的引用	(244)
9.4	结构体变量的赋值和初始化	(244)
9.5	结构体数组	(246)
9.6	指向结构体变量的指针变量	(249)
9.6.1	指向结构变量的指针	(249)
9.6.2	指向结构数组元素的指针	(251)
9.6.3	结构指针变量作函数参数	(252)
9.7	用结构体指针处理链表	(254)
9.7.1	动态存储分配	(254)
9.7.2	链表	(255)
9.8	综合示例及分析	(263)
9.9	小结	(273)
	习题	(273)

第 10 章 共用体与枚举	(277)
10.1 共用体	(277)
10.1.1 共用体类型	(277)
10.1.2 共用体变量的定义	(278)
10.1.3 共用体变量的引用	(279)
10.1.4 共用体变量的赋值和使用	(279)
10.2 枚举类型	(281)
10.2.1 枚举类型的定义	(281)
10.2.2 枚举变量的定义	(282)
10.2.3 枚举变量的赋值和使用	(282)
10.3 typedef 定义新的类型标识符	(285)
10.4 小结	(287)
习题	(288)
第 11 章 预处理功能	(291)
11.1 宏定义	(291)
11.1.1 不带参数的宏	(291)
11.1.2 带参数的宏	(294)
11.2 文件包含	(298)
11.3 条件编译	(298)
11.4 小结	(301)
习题	(301)
第 12 章 文件	(304)
12.1 文件概述	(304)
12.1.1 文件的基本概念	(304)
12.1.2 文件的分类	(304)
12.1.3 文件缓冲区	(305)
12.1.4 文件类型指针	(305)
12.2 文件打开与关闭	(306)
12.2.1 文件打开函数	(306)
12.2.2 文件关闭函数	(308)
12.3 文件的读写	(309)
12.3.1 文件写函数	(309)
12.3.2 文件读函数	(317)
12.4 文件的随机读写	(322)
12.4.1 rewind() 函数	(322)
12.4.2 fseek() 函数	(323)
12.5 文件检测函数	(325)
12.6 综合示例及分析	(325)
12.7 小结	(329)

习题	(329)
第 13 章 位运算	(332)
13.1 位运算符和位运算	(332)
13.1.1 位运算符	(332)
13.1.2 按位与运算	(333)
13.1.3 按位或运算	(333)
13.1.4 按位异或运算	(333)
13.1.5 按位求反运算	(334)
13.1.6 左移运算	(334)
13.1.7 右移运算	(335)
13.2 位域结构	(336)
13.2.1 位域的概念	(336)
13.2.2 位域结构和位域变量的说明	(336)
13.2.3 位域的使用	(337)
13.3 综合示例及分析	(338)
13.4 小结	(340)
习题	(340)
附录	(343)
附录 1 C 语言的常用关键字	(343)
附录 2 基本 ASCII 码对照表	(344)
附录 3 C 语言运算符的优先级与结合性	(346)
参考文献	(348)

第 1 章 C 语言程序设计入门

1.1 计算机基础

1.1.1 计算机系统组成

完整的计算机系统由硬件系统和软件系统两部分组成。如图 1-1 所示。

硬件系统是指由电子器件和机电装置组成的计算机实体，是看得见、摸得着的设备，故称硬件。硬件是计算机进行信息处理的物质基础。计算机硬件包括运算器、控制器、存储器、输入/输出 (I/O) 设备。

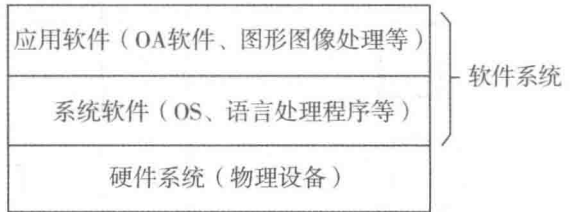


图 1-1 计算机系统组成示意图

软件系统是为计算机运行工作服务的全部程序、数据和技术资料的总称。软件

是计算机的灵魂。软件系统包括系统软件和应用软件两类。系统软件控制和维护计算机正常运行，管理计算机各种资源；应用软件是在系统软件支持下为完成某一特定任务而工作的软件。计算机的系统组成如图 1-2 所示。



图 1-2 计算机系统组成

1.1.2 计算机系统工作原理

计算机的基本工作原理是存储程序和程序控制。要预先把指挥计算机进行操作的指令序列（称为程序）和原始数据通过输入设备输送到计算机内存存储器中。每一条指令中明确规定了计算机从哪个地址取数，进行什么操作，然后送到什么地址去等步骤。

程序与数据一样存储，按程序编排的顺序，一步一步地取出指令，自动地完成指令规定的操作是计算机最基本的工作原理。这一原理最初是由美籍匈牙利数学家冯·诺依曼于1945年提出来的，故称为冯·诺依曼原理。计算机系统工作原理如图1-3所示。

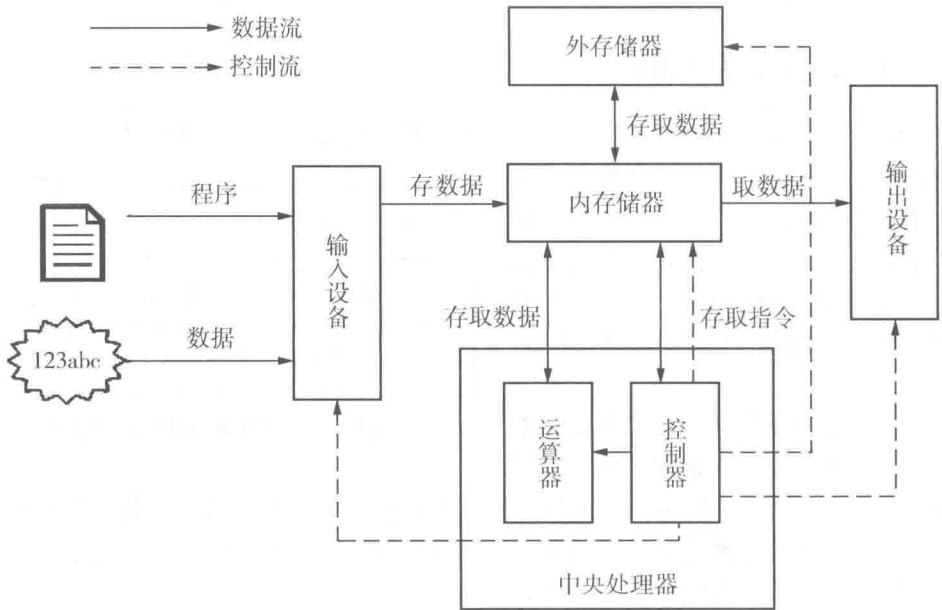


图 1-3 计算机系统工作原理

1.1.3 计算机语言

计算机语言 (Computer Language) 是指用于人与计算机之间通信的语言，是由计算机能识别的字符集和语法规则组成的计算机各种指令（或各种语句）。

计算机语言的种类非常多，总的来说可以分成机器语言、汇编语言、高级语言三大类。

(1) 机器语言：机器语言是指数据和指令均用二进制表示，计算机能够识别的语言。机器指令在设计硬件时，其格式及含义已经由设计者定义好了。用机器语言编写的程序，只有0和1两种符号，0和1的不同组合，代表不同的信息。机器语言可被计算机直接识别，运行速度快。但是依赖于特定的硬件，不便于记忆、阅读和书写。机器语言源程序如图1-4所示。

```

机器语言源程序
0010 0100 1011 0000
1100 1011 1110 1011
1000 1010 0000 0101
.....
    
```

图 1-4 机器语言源程序

(2) 汇编语言：汇编语言是指用一些简洁的英文符号来替代计算机能识别的二进制指令，其本质仍然是对硬件直接操作，源程序较长、复杂、通用性差、容易出错。汇编语言书写的程序并不能直接运行，需要使用汇编程序将源程序代码翻译成二进制的机器语言。汇编语言源程序如图 1-5 所示。

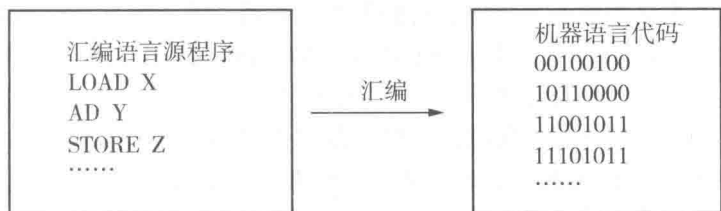


图 1-5 汇编语言源程序

(3) 高级语言：高级语言是指采用类似人类自然语言的编程语言。其通用性好，不依赖于机器。计算机不能直接运行高级语言编写的程序，而需要借助工具将高级语言翻译成机器语言。通常将高级语言翻译成机器语言需要分两个步骤，如图 1-6 所示。

第 1 步：将高级语言源程序编译成二进制目标文件代码。

第 2 步：将目标文件代码与库文件链接生成可直接运行的二进制机器语言代码。

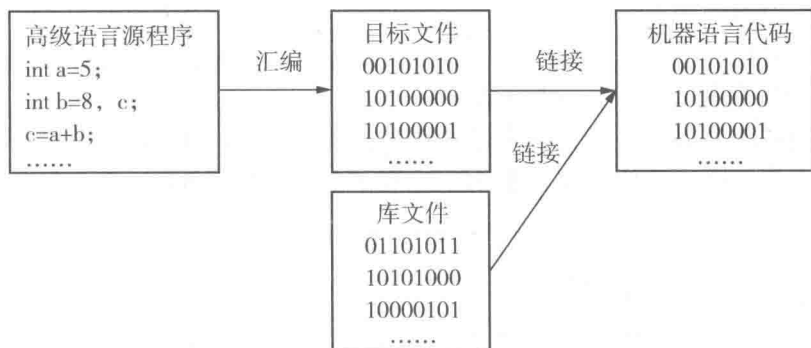


图 1-6 高级语言源程序翻译成机器语言代码

1.1.4 数制及其转换

1. 数的表示及进制间转换

数的进制也就是进位制。

我们日常使用的计数方式，只有 0~9 共 10 个数字，基数为 10，在加减法运算中，逢 10 进 1，借 1 当 10，所以叫作十进制。

二进制只有 0 和 1 两个数字，基数为 2，在加减法运算中，逢 2 进 1，借 1 当 2。

除了二进制，编程中也经常使用八进制和十六进制。

八进制有 0~7 共 8 个数字，基数为 8，逢 8 进 1，借 1 当 8。

十六进制中，用 A 来表示 10，B 表示 11，C 表示 12，D 表示 13，E 表示 14，F 表示 15，因此有 0~9 和 A~F 共 16 个数字，基数为 16，逢 16 进 1，借 1 当 16。

2. 内存中数据的存储

计算机要处理的信息是多种多样的，如十进制数、文字、符号、图形、音频、视频等，这些信息在人们的眼里是不同的。但对于计算机来说，它们在内存中都是一样的，都是以二进制的形式来表示。

内存条包含了上亿个电子元器件，它们很小，达到了纳米级别。这些元器件的电压会变化，要么是 0，要么是 5V，只有这两种电压。5V 是通电，用 1 来表示，0 是断电，用 0 来表示。所以，一个元器件有 2 种状态，0 或者 1。

通过控制这些元器件的通、断电，会得到很多 0、1 的组合。例如，8 个元器件有 $2^8 = 256$ 种不同的组合，16 个元器件有 $2^{16} = 65536$ 种不同的组合。虽然一个元器件只能表示 2 个数值，但是多个元器件结合起来就可以表示很多数值了。

一般情况下我们将 8 个元器件看作一个单位，1 个元器件称为 1 比特 (bit) 或 1 位，8 个元器件称为 1 字节 (Byte)，那么 16 个元器件就是 2Byte，32 个就是 4Byte，依此类推。

单位换算：

8 bit = 1Byte

1024Byte = 1KB

1024KB = 1MB

1024MB = 1GB

1024GB = 1TB

3. 十进制转换为二进制——辗除法

辗除法即“除模取余”法，将十进制转换为二进制时，将十进制的数除以二（二进制的模）取余数。图 1-7 所示为十进制的 19 转换为二进制的过程。

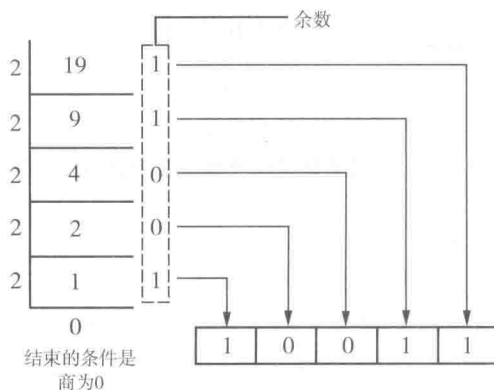


图 1-7 辗除法示意图

如图 1-7 所示，以 2 为除数，一直相除下去，直到商为 0，余数则为求得的二进制数。注意：余数要倒序排列，也就是说，最先求得的余数排在二进制的最后面，最后求得的余数排在二进制的最前面。上面的例子中，最后求得的二进制数为 10011。

4. 二进制、八进制、十六进制转换为十进制

在 C 语言中，八进制通常以“0”开头（注意是数字 0，而不是字母 o），例如 0307 表示八进制数 307；十六进制通常以“0x”或“0X”开头（不区分大小写），例如 0xE27、

0X89F 表示十六进制数 E27、89F。

十进制 $4321 = 4 \times 10^3 + 3 \times 10^2 + 2 \times 10^1 + 1 \times 10^0$

二进制 $1101 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1 = 13$

二进制 $110.11 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 4 + 2 + 0 + 0.5 + 0.25 = 6.75$

八进制 $0723 = 7 \times 8^2 + 2 \times 8^1 + 3 \times 8^0 = 7 \times 64 + 2 \times 8 + 3 \times 1 = 467$

十六进制 $0X89F = 8 \times 16^2 + 9 \times 16^1 + 15 \times 16^0 = 8 \times 256 + 9 \times 16 + 15 \times 1 = 2207$

5. 二进制和八进制的转换

二进制向八进制的转换是每三位二进制数转换为一位八进制数，运算的顺序是从低位向高位依次进行，高位不足三位用零补充。以二进制“1011101”为例，如图 1-8 所示。

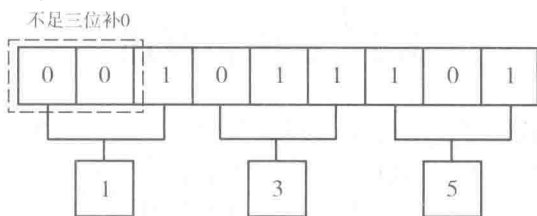


图 1-8 二进制转八进制

二进制 1011101 转换为八进制为 0135。

八进制向二进制转换的思路是八进制的一位转换为二进制的三位，运算的顺序是从低位向高位依次进行。同样以八进制 0135 为例，如图 1-9 所示。

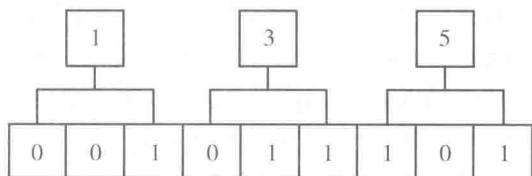


图 1-9 八进制转二进制示意图

八进制 0135 转换为二进制为 1011101。

6. 二进制和十六进制的转换

二进制向十六进制转换时，四位二进制数转换成十六进制的一位，运算的顺序是从低位向高位依次进行，高位不足四位用零补。以二进制 1001011101 转换成十六进制为例，如图 1-10 所示。

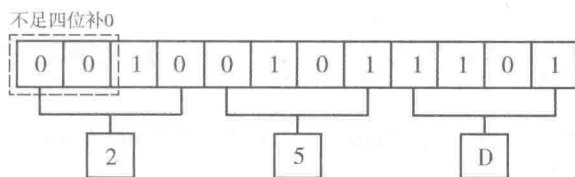


图 1-10 二进制转十六进制

二进制 1001011101 转换为十六进制为 0X25D。

十六进制向二进制转换，就是把十六进制的一位转换成二进制的四位，注意运算的顺序是从低位向高位依次进行。同样以十六进制“0X25D”为例，如图 1-11 所示。

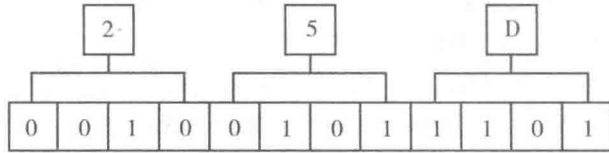


图 1-11 十六进制转二进制

十六进制“0X25D”转换为二进制为 1001011101。

1.2 C 语言概述

C 语言是一种计算机程序设计语言，它既具有高级语言的特点，又具有汇编语言的特点。它由美国贝尔研究所的 D. M. Ritchie 于 1972 年推出，它可以作为系统设计语言，编写系统应用程序。它的应用范围广泛，不仅仅用在软件开发上，各类科研都需要用到 C 语言，适于编写系统软件，二维、三维图形和动画，以及嵌入式系统开发。

1.2.1 C 语言的发展

C 语言之所以命名为 C，是因为 C 语言源自美国计算机科学学者 Ken Thompson 发明的 B 语言，而 B 语言则源自 BCPL (Basic Combined Programming Language)。

1967 年，剑桥大学的 Martin Richards 对 CPL 进行了简化，于是产生了 BCPL。

1970 年，美国贝尔实验室的 Ken Thompson，以 BCPL 为基础，设计出很简单且很接近硬件的 B 语言（取 BCPL 的首字母）。并且他用 B 语言写了第一个 UNIX 操作系统。

1972 年，美国贝尔实验室的 D. M. Ritchie 在 B 语言的基础上最终设计出了一种新的语言，他取了 BCPL 的第二个字母作为这种语言的名字，这就是 C 语言。

1977 年，D. M. Ritchie 发表了不依赖具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。

1978 年，由美国电话电报公司 (AT&T) 贝尔实验室正式发表 C 语言。Brian Kernighan 和 Dennis Ritchie 出版了一本书，名叫 *The C Programming Language*。这本书被 C 语言开发者们称为“K&R”，很多年来被当作 C 语言的非正式的标准说明。人们称这个版本的 C 语言为“K&R C”。

到 20 世纪 80 年代，C 语言被广泛应用，从大型机到小型机，也衍生了 C 语言的很多不同版本。

1983 年，美国国家标准局 (American National Standards Institute, ANSI) 成立了一个委员会，专门制定 C 语言标准。

1989 年 C 语言标准被批准，被称为 ANSI X3.159—1989 *Programming Language C*。这个版本的 C 语言标准通常被称为 ANSI C。

1990 年，国际标准化组织 ISO (International Organization for Standards) 接受了 89