

新编考研辅导丛书

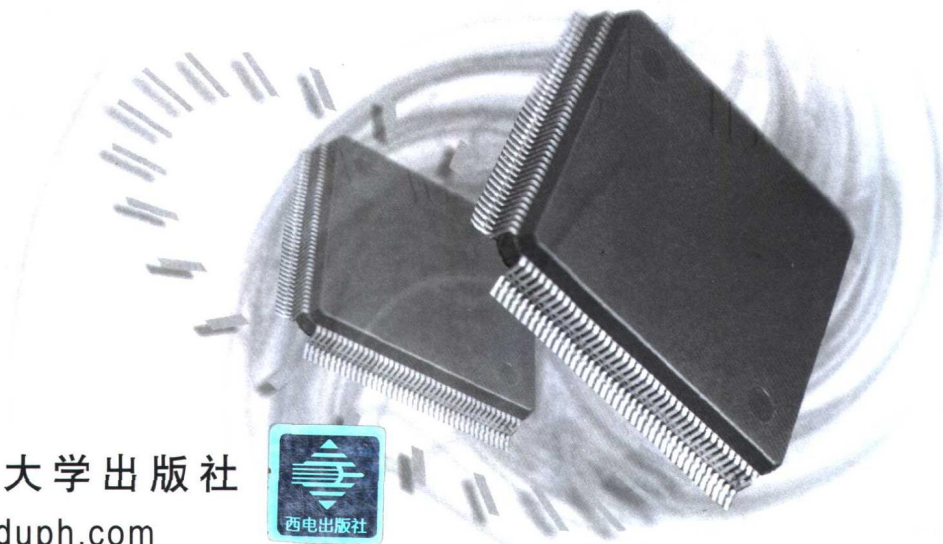
(2003)版

C Programming Language

C语言程序设计辅导

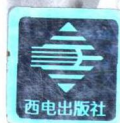
张淑平 霍红卫 编著

核心内容阐述
典型例题分析
常见错误点评
全真考题解答



西安电子科技大学出版社

<http://www.xduph.com>



新编考研辅导丛书（2003 版）

C 语言程序设计辅导

张淑平 霍红卫 编著

西安电子科技大学出版社
北京信息工程学院图书馆

内 容 简 介

C 语言是当前使用最为普遍的几种程序设计语言之一。

学习一种程序设计语言的目的, 在于使用这种语言编制程序, 以便更好地利用计算机解决现实的问题。因此本书主要着眼于对 C 语言的基本内容进行概括式的阐述, 期望能帮助考生掌握 C 语言的核心内容; 力求通过大量实例体现出该语言的基本要领的运用, 并突出知识的综合运用。书中的常用算法一章, 进一步体现了 C 语言在一些典型算法中的应用。精选的习题可供读者自练自测。本书还提供了多套近年来我校及其它院校的研究生入学考试试题及参考答案。全书题源广泛, 题型多样, 综合性强。

本书既可作为报考硕士研究生的考生的复习辅导书, 也可用作相关专业课程学习或复习的指导书, 还可作为有关教师的教学参考书。

新编考研辅导丛书 (2003 版)

C 语言程序设计辅导

张淑平 霍红卫 编著

责任编辑 李惠萍

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)8227828 邮 编 710071

http: //www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印 刷 西安文化彩印厂

版 次 2002 年 10 月第 1 版 2002 年 10 月第 1 次印刷

开 本 787 毫米×960 毫米 1/16 印张 17.875

字 数 358 千字

印 数 1~6 000 册

定 价 22.00 元

ISBN 7-5606-0915-5 / TP · 0845

XDUP 1186A01-1

*** 如有印装问题可调换 ***

本书封面贴有西安电子科技大学出版社的激光防伪标志, 无标志者不得销售。

前 言

C 语言是当前使用最为普遍的几种程序设计语言之一。由于其简洁灵活、目标程序效率高、可移植性好以及兼有高级语言和低级语言的特点,因而 C 语言在学术界和工业界都得到了广泛的使用。面向对象语言 C++便是对 C 语言的一种兼容性扩充,因此学好 C 语言是进一步学习 C++的基础。

学习一种程序设计语言的目的,在于使用这种语言编制程序,以便更好地利用计算机解决实际问题。本书主要着眼于对 C 语言的基本内容进行概括式的阐述,期望能帮助考生掌握 C 语言的核心内容,力求通过大量实例体现出该语言的基本概念的运用,并突出知识的综合运用和对重点、难点、考点的解析。需要注意的是,研究生入学考试中关于程序设计语言的部分,已经逐渐过渡到和数据结构课程中的算法部分相结合来进行考核,因此在编程语言的学习和复习过程中,编写出能够求解问题的完整程序是贯穿始终的目标。

全书共分为 10 章。从第 1 章至第 9 章都由基本内容、典型例题解析和常见错误分析组成。其中,基本内容部分简明扼要地介绍了 C 语言的基本概念,主要包括运算符和数据类型。典型例题解析构成全书的主要部分,强调对基本概念的应用,这一部分选题广泛,题型多样,覆盖面广,信息量大,综合性强,重点突出。常见错误分析给出了使用 C 语言编写程序时的常见错误及分析方法。其中的常用算法一章,进一步体现了 C 语言在一些典型算法中的应用。精选的习题可供读者自练自测。本书多数例题和参考答案都带有注释,这一切旨在起到解难释疑、开阔思路、触类旁通之目的。本书还提供了近年来我校及其它院校的研究生入学考试试题及参考答案。书中的程序均按照 ANSI C 标准编写,并在 Turbo C2.0 和 Visual C++6.0 的编译环境中调试通过。

本书既可作为报考硕士研究生的考生的复习辅导书,也可用作相关专业课程学习或复习的指导书,还可作为有关教师的教学参考书。

林兴亚老师仔细审阅了全稿,并提出了宝贵意见,在此表示感谢。

在本书的编写过程中,得到了西安电子科技大学出版社李惠萍编辑的具体指导,在此表示衷心的感谢。

尽管作者有较长时间从事该课程教学及实践的经验,但由于工作繁忙和水平有限,书中可能还存在疏忽和错误之处,恳请读者和同行批评指正。

编 者
2002 年 8 月

目 录

第 1 章 概述.....	1
1.1 C 语言的由来及特点.....	1
1.2 C 语言的词汇.....	2
1.3 C 语言程序设计的基本要素.....	4
1.4 C 程序的运行步骤.....	5
第 2 章 数据、运算和表达式.....	7
2.1 数据.....	7
2.2 运算和运算符.....	12
2.3 表达式.....	22
2.4 输入和输出.....	23
2.5 常见错误.....	28
2.6 典型例题解析.....	29
2.7 练习题.....	33
第 3 章 C 语句和控制流.....	34
3.1 C 语言的语句.....	34
3.2 常见错误.....	44
3.3 典型例题解析.....	45
3.4 练习题.....	54
第 4 章 数组和字符串.....	56
4.1 数组.....	56
4.2 字符串.....	60
4.3 常见错误.....	62
4.4 典型例题解析.....	62
4.5 练习题.....	77
第 5 章 函数.....	78
5.1 基本内容.....	78

5.2	存储类别和类型限定符 const 和 volatile	83
5.3	常见错误	87
5.4	典型例题解析	89
5.5	练习题	95
第 6 章	指针	97
6.1	基本内容	97
6.2	指针的应用	100
6.3	常见错误	102
6.4	典型例题解析	102
6.5	练习题	118
第 7 章	枚举类型、结构体与共用体	119
7.1	枚举类型和 typedef	119
7.2	结构体和共用体	121
7.3	常见错误	127
7.4	典型例题解析	128
7.5	练习题	150
第 8 章	文件	151
8.1	文件类型	151
8.2	常见错误	157
8.3	典型例题解析	157
8.4	练习题	168
第 9 章	预处理命令	169
9.1	基本内容	169
9.2	常见问题	172
9.3	典型例题解析	174
9.4	练习题	176
第 10 章	常用算法	177
10.1	排序算法	177
10.2	常用的算法设计方法	186
10.3	典型例题解析	203

附录 A 研究生入学考试试题	219
研究生入学考试试题(一)——西安电子科技大学 2002 年考题	219
研究生入学考试试题(二)——西安电子科技大学 2001 年考题	222
研究生入学考试试题(三)——西安电子科技大学 2000 年考题	225
研究生入学考试试题(四)——西安电子科技大学 1999 年考题	229
研究生入学考试试题(五)——西安电子科技大学 1998 年考题	234
研究生入学考试试题(六)——浙江大学 1999 年考题	238
研究生入学考试试题(七)——中科院 1999 年考题	240
研究生入学考试试题(八)——西北工业大学 2002 年考题	243
研究生入学考试试题(九)——西北工业大学 2001 年考题	247
附录 B 研究生入学考试试题参考答案	251
研究生入学考试试题(一) 参考答案	251
研究生入学考试试题(二) 参考答案	255
研究生入学考试试题(三) 参考答案	257
研究生入学考试试题(四) 参考答案	259
研究生入学考试试题(五) 参考答案	260
研究生入学考试试题(六) 参考答案	261
研究生入学考试试题(七) 参考答案	264
研究生入学考试试题(八) 参考答案	272
研究生入学考试试题(九) 参考答案	276
参考文献	278



第 1 章

概 述

人们在使用计算机帮助自己工作(解题)时,应告知计算机要“做什么工作(解决什么问题)”,和“怎样工作”,即告知它工作(解题)的具体方法(思路)和步骤。

处理一件工作或解一个问题的方法和思路,称为“算法”。但要计算机帮助我们工作(解题),还必须将“算法”转化为具体的解题步骤。解题步骤须用计算机语言一句一句地描述出来,最终以计算机“指令”的序列表示出来。这种用计算机高级语言的语句序列或“机器指令”的序列描述出的解题步骤称为“程序”。

根据求解问题的“算法”设计程序的过程,称为“程序设计”。计算机就是通过执行程序来处理信息或数据而最终解决问题的。

C 语言是处理信息、实现人和计算机交流信息用的工具,是人机交流的一种程序设计语言。



1.1 C 语言的由来及特点

C 语言是由 Dennis Ritchie 发明并首先在使用 UNIX 操作系统的 PDP-11 上面实现的,它是从着手开发一个称为 BCPL 的语言的过程中产生出来的。BCPL 主要在欧洲使用。BCPL 由 Martin Richards 研制,它对一个称为 B 的语言有影响,B 语言由 Ken Thompson 发明,它导致了 C 语言的研制。

C 语言既具有高级语言的特点(结构化设计、可读性及可移植性强),又具有低级语言的特性(位操作、对内存地址的操作等),这使得它很适合于进行系统程序设计。C 语言的另一个特点是它只有很少的关键字(或称保留字,只有 32 个),因此很简洁。

虽然 C 语言有五种基本的数据类型,但与 Pascal 或 Ada 相比,它不是一种强类型化的语言。C 语言几乎允许所有类型转换,并且在大多数表达式中可以自由地把字符类型与整型混合在一起使用。

习惯上将 1978 年 C 的作者 D.M.Ritchie 发表的《C 语言修订报告》称为传统 C。标准



C 是指国际标准化组织 ISO 所规定的 C 语言 (ISO/IEC 9899:1990(E) INTERNATIONAL STANDARD Programming Language C)。ANSI 是美国国家标准协会的缩写, 该协会致力于包括编程语言在内的多种标准的制定, 1990 年, ANSI 委员会 X3J11 完成了 C 的标准草案并获得 ISO 的批准。

ANSI C 标准规定了用 C 编写的程序的格式, 并确定了如何解释这些程序。该标准的目的是促进 C 语言程序在各种机器上的可移植性、可靠性、可维护性和有效执行。所有主要的 C 编译器都遵循 ANSI C 标准。



1.2 C 语言的词汇

C 程序是一个字符序列, 序列中允许出现的字符集合构成 C 语言的字母表。由字母表中的字符构成 C 语言的基本词汇, 可将这些基本词汇分为六种类型, 即标识符、保留字(关键字)、常量、串常量、运算符及标点符号。下面我们先简单介绍 C 语言中用到的字符(集)和词汇(保留字、标识符)。

1.2.1 C 语言的字符集

C 语言用到的字符集合就是 ASCII 字符集。我们知道 ASCII 字符集共有 128 个字符, 其中 95 个为图形字符(可打印字符), 33 个为控制(非图形)字符。

95 个图形字符是:

- (1) 大小写英文字母(A~Z, a~z, 共 52 个);
- (2) 数字(0~9, 共 10 个);
- (3) 其他符号(+ - * / % = () [] { } < > ' ` " ! & | ^ ~ , . : ; \ _ @ # \$? , 共 32 个);
- (4) 空白字符(空格)。

对于控制字符, 在需要使用时就得用特殊的图形字符表示出来。这些特殊的字符用一种特殊的方式进行描述, 即借助于转义字符“\”和图形(可打印)字符, 参看表 1.1。

“转义”就是将一个字符原来的含义另作解释, 如小写字母 a 前加上反斜杠, 形成的符号“\a”就成了响铃报警这一控制功能的图形符号, 然后就可在程序中表示出来。单引号(')和双引号(")在 C 语言中是分别作为单个字符和字符串的分界符使用的, 反斜杠字符(\)则是作为转义字符的前缀即定义符来使用的, 因此当需要打印(输出)这三个字符(图形)时, 也只得借助于转义符来表示, 如表 1.1 所示。至于转义字符\后跟 0~7 的数字串, 在 C 语言中就规定其对应八进制数, 最多用三位八进制数字来表示一个字符的 ASCII 码值。转义



字符`\x(X)`后跟 0~9 或 A~F 的字符串, 在 C 语言中就规定其对应十六进制数, 最多用两位十六进制数字表示一个字符的 ASCII 码值。

表 1.1 转义字符表

字符(ASCII 值)	功 能	字符(ASCII 值)	功 能
<code>\n(10)</code>	换行(LF, NL)	<code>\a (7)</code>	响铃报警(BEL)
<code>\t(9)</code>	水平制表(跳格 HT)	<code>\' (39)</code>	单引号(')字符
<code>\v(11)</code>	垂直制表(VT)	<code>\" (34)</code>	双引号(")字符
<code>\b(8)</code>	退格(BS)	<code>\\ (92)</code>	反斜杠(\)字符
<code>\r(13)</code>	回车(CR)	<code>\ddd</code>	八进制数所表示的字符
<code>\0(0)</code>	空字符(串结束标志)	<code>\xhh</code>	十六进制数所表示的字符

1.2.2 C 语言使用的单词

1. 标识符

标识符是指用户自定义的字符序列, 通常用于表示程序中需要指明的对象名称, 比如表示符号常量、变量、数组、函数等对象的名字。

C 语言规定, 标识符是以字母或下划线开头的由字母、数字、下划线组成的字符串, 标识符的有效长度通常由编译系统确定, 一般不要长于 31 个字符, 而有些 C 编译程序只识别前 8 个字符。

标识符中对字母的大小写是敏感的, 如 `indx`、`Indx` 和 `INDEX` 是不同的标识符。

由于可能和系统定义的名字相冲突, 因而程序员定义的标识符尽量不要以 “_” 开头。

标识符主要用于给程序涉及的对象(事、物或信息)命名, 所以应尽可能地选取有意义的词汇, 以便能方便地从标识符理解其所代表的对象的特征, 比如 `Price`、`Tax_rate` 等。

表示常量的标识符在 C 语言中一般由大写字母组成。

2. 保留字

C 语言中用到一些英语单词, 它们符合标识符的定义, 但是已为其规定了特殊的含义, 不允许再作其它用途。这些特定的标识符称为“保留字(或关键字)”。

C 语言规定的保留字如表 1.2 所示。



表 1.2 C 语言规定的保留字

保留字	含 义	保留字	含 义	保留字	含 义
auto	自动	extern	外部	sizeof	计算字节数
break	中止	float	浮点	static	静态
case	情况	for	对于	struct	结构
char	字符	goto	转向	switch	开关
continue	继续	int	整型	typedef	类型定义
const	常量	if	如果	union	共用
default	缺省	long	长整型	unsigned	无符号
do	做	register	寄存器	void	空
double	双	return	返回	volatile	可变的
else	否则	short	短整型	while	当
enum	枚举	signed	带符号		

在不同的 C 版本中，保留字是有所不同的。



1.3 C 语言程序设计的基本要素

程序模块或函数是有机组成的语句序列。程序设计包括很多的要素，但是最基本的(不是全部)要素有六个，即输入、数据、运算、输出、程序流控制和函数(子程序)。

(1) 输入(Input): 从键盘、外存或其它输入/输出端口读入信息。

(2) 数据(Data): 指各种类型的信息，最基本的数据类型有数值(整数和实数)、文字(字符和字符串)和地址(变量的内存地址)等。

(3) 运算(Operation): 包括赋值运算(将一个表达式的值赋给变量 a)、算术运算(加、减、乘、除、整除取余)、关系运算(大于、大于等于、小于、小于等于、相等、不等)和逻辑运算(与、或、非、异或)等。

(4) 输出(Output): 将信息写出到显示器(屏幕)、打印机、外存或其它输入/输出端口。

(5) 程序流程控制结构:

- 顺序执行：语句的执行按照书写顺序依次进行。
- 条件执行(Conditional Execution)：一组语句只有当满足一个特定条件(真)时才执行(如果不满足条件(假)，则跳过这组语句)。
- 循环(Loops)：一组语句重复执行一定的次数，直到不再满足能够重复的条件时为止。

(6) 函数(Function)：是完整而独立的有“名字”的一组语句。它的名字可以在别的程序模块内出现(称为调用)，程序模块运行时，遇到函数名，就执行该函数。

学习程序设计的过程，就是熟悉并掌握这些基本要素的过程。



1.4 C 程序的运行步骤

一个 C 语言程序由一个或若干个编译单位构成，每个编译单位可以有若干个函数。C 程序中必须有且只有一个名字为 `main` 的函数(主函数)，可以任意安排函数的排列次序。C 程序的总体结构如图 1-1 所示。

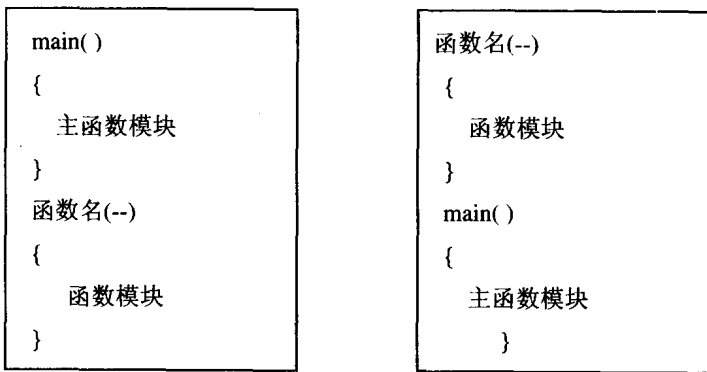


图 1-1 C 程序的结构

一个 C 程序的运行包括编辑、编译、连接和运行等阶段，大多数的 C 编译系统都提供一个集成的环境，使得程序员可以在该环境中完成这几个阶段的工作。一个 C 程序执行时总是从主函数开始的。

我们用一个流程图描述一般情况下编写 C 程序并调试运行的步骤，如图 1-2 所示。

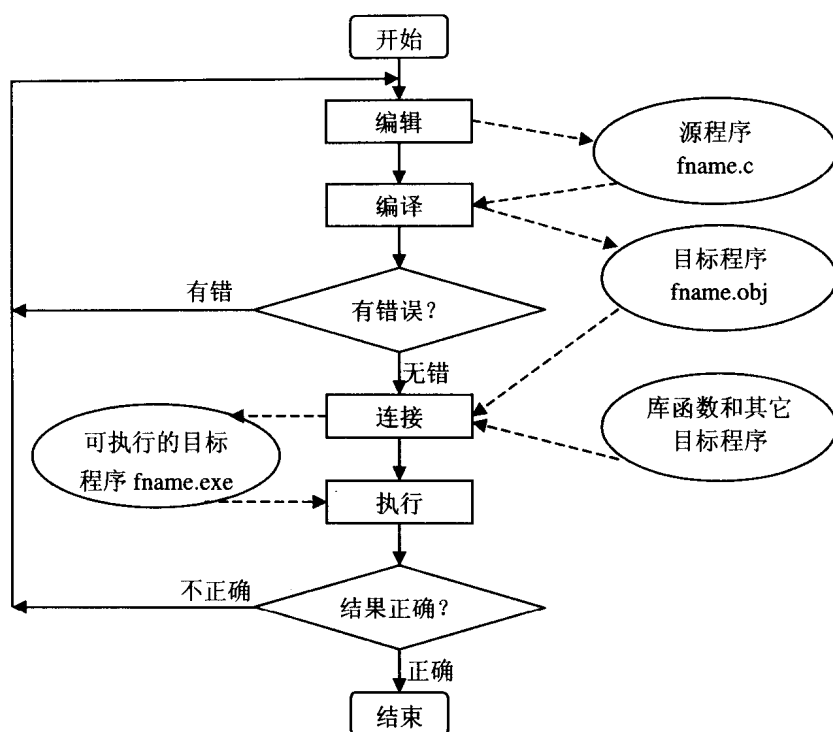


图 1-2 运行 C 程序的一般步骤



第2章

数据、运算和表达式

计算机程序加工处理的对象是“数据(各种信息)”，这里的“数据”一词是广义的，它可以是能进行运算的数，也可以是能进行增、删、拼接处理的字符(文字、符号)等。

在 C 语言中，“运算”一词也是一个广义的概念，它包含算术运算、关系运算、逻辑运算以及一些 C 语言中定义的运算(如赋值运算、逗号运算)等。

本章内容是学习 C 程序设计的重要基础。



2.1 数 据

各种信息在 C 语言中统称为数据。数据有以下两种属性：

- (1) 数据类型：指数据内在性质的表现形式；
- (2) 存储类型：指数据在存储器中的存储方法。

C 语言中的基本数据类型可以分为数值型和字符型，其中数值型又可分为整数类型和浮点类型。基本数据类型还可用于以下修饰符修饰：signed(带符号的)，unsigned(无符号的)，short(短型的)，long(长型的)。

当字符类型(char)用 signed 和 unsigned 修饰后，作为小整数类型使用。

C 语言的数据类型可归纳为图 2-1 所示。

计算机中只能表示有限范围的数据，基本数据类型的取值范围如表 2.1 所示。

表 2.1 数据的值域(取值范围)

数据类型	占用内存空间(字节数)	值域(数据的取值范围)
char	8b(1B)	0~255 -128~127
(signed) int	16b(2B)	$-2^{15} \sim 2^{15}-1$ (32 767)
(signed) short	16b(2B)	$-2^{15} \sim 2^{15}-1$ (32 767)
(signed) long	32b(4B)	$-2^{31} \sim 2^{31}-1$ (2 147 483 647)



续表

数据类型	占用内存空间 (字节数)	值域(数据的取值范围)
unsigned int	16b(2B)	0~65 535 ($2^{16}-1$)
unsigned short	16b(2B)	0~65535 ($2^{16}-1$)
unsigned long	32b(4B)	0~ $2^{32}-1$ (4 294 967 295)
float	32b(4B)	3.4E-38~3.4E+38
double	64b(8B)	1.7E-308~1.7E+308
long double	>=64b(8B)	1.7E-308~1.7E+308

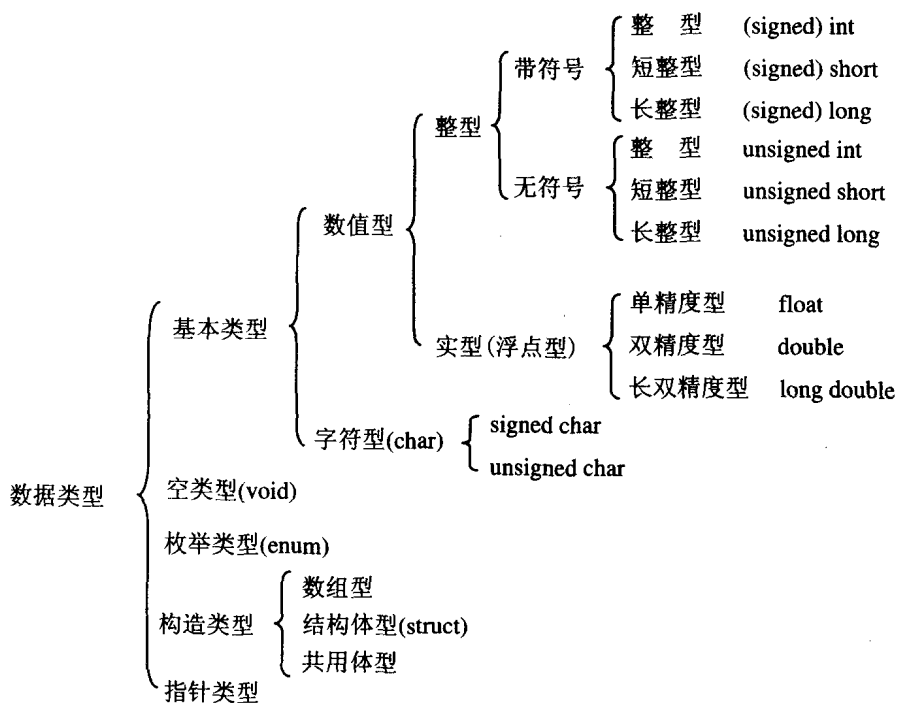


图 2-1 C 语言的数据类型

程序中处理的数据还有常量和变量的区别。

2.1.1 常量

常量(常数)是在程序运行过程中值保持不变的数据。

1. 整型常量

整型常量可以用十进制整数、以 0 起始的八进制整数和 0x(0X)起始的十六进制整数等形式书写。若在数的末尾加 L(或 l), 则表示是长整型数; 正数前的符号“+”可省略。

2. 实型常量

实型常量一般由符号(“+”号可省)、整数、小数点、小数等部分组成, 其中, 整数或小数之一可省略。指数型实数由尾数(前带符号, “+”号可省)、字母 e 或 E(代表底数 10)、指数等部分组成, 其中指数是整型数。

3. 字符常量

字符常量就是‘字符’(该字符的 ASCII 常量值)形式的常量, 这里单引号(')成了专用符号, 因此单引号本身只得借助转义字符来表示, 即表示成('\'), 其 ASCII 常量值就可记为'\'(39); 而反斜杠字符(\)就需表示成('\\'), 其 ASCII 常量值为'\\'(92)。因此, '\n'(10)、'\101'(65)、'\x41'(65)等都是转义字符常量, 括号内为对应 ASCII 字符的十进制码值。

4. 字符串常量

字符串常量是程序中用双引号括起来的符号串。这里双引号(")成了专用符号, 因此字符串中的双引号(")本身用转义字符来表示, 即表示成('\"). 例如, 与字符串(\\ABCD\\)对应的字符串常量就是\\ABCD\\, 与字符串(ABCD)对应的字符串常量表示为"ABCD", 也可以用"\101\102\x43\x44"表示。

字符串常量中字符(包括空格)的个数称为字符串的长度。一个字符常量在内存中占用一个字节单元, 而一个字符串常量在内存中占用的空间是字符串长度值加 1 个字节单元, 这多出的一个字节单元存放的是空字符, 用转义字符 '\0' 表示, 用作字符串的结束标记。因此, 字符常量'A'和字符串常量"A"是不同的, 前者只占一个字节的内存空间, 后者却占 2 个字节的内存空间。

5. 符号常量

符号常量就是用标识符表示的常量。一般在程序开始处给标识符定义一个值, 在程序中用此标识符代表该常量值, 这样就可以明确地表明该常数的特性, 如圆周率 3.1416, 用 PI 表示就比较简洁明确。另外, 当需要在程序中多处书写一个常量值时(如 3.1416), 用符号常量就非常方便且不易出错。使用符号常量也便于修改常量的值, 例如将 3.1416 改为 3.1415926, 只需修改程序开始处的定义就可以了。

符号常量的定义(说明)格式如下:



#define 符号常量 常量值 (符号常量的数据类型由常量值决定)

符号常量可以按标识符的规则构成, 一般使用大写英文字母以突出其特性。例如:

```
#define PI 3.1415926
```

2.1.2 变量

变量是在程序运行过程中其值可以发生变化的量。变量的取名规则与标识符相同, 其中的英文字母常用小写, 用户程序中的变量名一般不以下横线(_)起始。

1. 变量的定义及初始化

程序中对数据的处理过程实际上是通过变量的操作实现的, 因此任一变量在使用前必须定义(或说明)其数据类型。变量定义(说明, 并可给变量赋初值)的语句格式如下:

数据类型符 变量名 1(=初值 1), 变量名 2(=初值 2), ...;

```
main()
{ int i1, i2;          /*定义变量 i1,i2*/
  char c1, c2='a';     /*定义变量 c1、c2 并赋初值(初始化)*/
  float f;
  ...
}
```

2. 变量的生存期和作用域

一个变量可以在函数(或语句块)内部定义, 称为局部变量(也称为内部变量), 也可以在各函数外部(两个函数之间)定义, 称为全局变量(也称为外部变量)。

编译程序将为变量分配物理存储单元(内存单元或 CPU 寄存器)。显然, 变量占有物理存储单元的时间总是有限的, 这段时间就称为变量的生存期。如果变量的生存期与某个函数(或某个复合语句)的执行期相当, 则称此变量为“局部变量”; 如果变量的生存期与某个程序(编译单位)的执行期相当, 则称此变量为“全局变量”。与变量生存期相对应的函数(或语句块)或程序文件(编译单位), 称为变量的作用域。简而言之, 作用域是可以存取变量的代码范围。生存期是可以存取变量的时间范围。

全局变量在整个程序运行期间都存在, 但局部变量却随着控制流程进入函数(复合语句)而产生、退出函数(复合语句)而销毁。

【例 1】在下面的例子中, 有一个全局变量 x 和两个局部变量 x 和 y 。在函数 $f()$ 中的 x 为一个局部变量, 与全局变量 x 无关。在函数 $\text{main}()$ 中引用的 x 是全局变量 x 。

```
#include <stdio.h>

int x;                      /*全局变量 x*/
```