

高职高专“十五”规划教材系列

# C 语言程序设计

闫利平 马林艺 编著



机械工业出版社

本书系统介绍了 C 语言编程知识,共分 11 章:第 1 章为 C 语言基础知识;第 2、3、4 章分别介绍了结构化程序设计的顺序、选择、循环结构;第 5 章为函数;第 6 章为编译与预处理;第 7 章为数组;第 8 章为指针;第 9 章介绍的是结构体、联合体、枚举类型;第 10 章为文件;第 11 章是综合应用。其中前 6 章是基础部分,后 5 章是实用的编程方法和技巧。

全书结构简洁,文字流畅;配有大量典型程序实例,突出通过程序学习程序的教学思想。每章章末均精选了基本题、练习题,作为学习该章内容的巩固与延伸。

本书适合作高职、高专、成人高校的教材和参考书,亦可作为 C 语言自学用书。

## 图书在版编目(CIP)数据

C 语言程序设计/闫利平,马林艺编著. —北京:机械工业出版社,2003.6  
高职高专“十五”规划教材

ISBN 7-111-12419-7

I. C... II. ①闫...②马... III. C 语言—程序设计—高等学校;技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 048034 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策 划:胡毓坚

责任编辑:时 静

责任印制:路 琳

北京蓝海印刷有限公司印刷·新华书店北京发行所发行

2003 年 7 月第 1 版·第 1 次印刷

787mm×1092mm 1/16·18.5 印张·459 千字

0001—5000 册

定价:26.00 元

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话(010)68993821、88379646

封面无防伪标均为盗版

# 出版说明

为了贯彻国务院发[2002]16号文件《国务院关于大力推进职业教育改革与发展的决定》的精神，进一步落实《中华人民共和国职业教育法》和《中华人民共和国劳动法》，实施科教兴国战略，大力推进高等职业教育改革与发展，我们组织力量，对实现高等职业教育培养目标和保证基本教学规格的文件基础课程、专业技术基础课程和重点建设专业主干课程的教材进行了规划和编写。

本套教材的内容涵盖了普通大专院校计算机及非计算机专业的文化基础课、专业基础课、专业课以及选修课程。主要分为文化基础、编程语言、硬件技术、网络信息、数据库应用及多媒体技术等几大类。为配合高职教育关于“培养 21 世纪与我国现代化建设要求相适应的一线科技实用型人才”的最新理念，我们特为本系列教材配备了实践指导丛书，以利于老师的教学和学生的学习。

本套教材将理论教学和实践教学紧密结合，图文并茂、内容实用、层次分明、讲解清晰，其中融入了作者长期的教学经验和丰富的实践经验。可用作各类大专院校、职业技术学校的教材，也可作为各类培训班的教材。

# 前 言

C 语言是一种应用十分广泛的语言，其程序处理能力强，代码效率高，具有高级语言程序设计的优点，又具有低级语言的特点，因此成为开发系统软件和应用软件的首选语言。例如：UNIX、Linux 操作系统都是使用 C 语言编写的。基于其诸多优势，不但计算机专业人员需要掌握它，广大非计算机专业人员也应该学习并掌握它。

作者结合多年 C 语言教学经验与教学实践，编写了本书，力求突出以下特色：

第一、深入浅出，简洁明快。学习计算机语言是为了编程解决实际问题，而通过程序深入学习计算机语言又是一种最有效的学习方法。为此，在透彻讲解基本概念的前提下，本书精选了约 130 个经过验证的典型程序，并对每个程序的结构、难点、变量设置及运行结果进行了详细的分析说明。对那些概念容易模糊、程序容易出错的地方，通过正、反两方面加以强调、说明。相信读者在认真思考的基础上，定能举一反三，触类旁通。

第二、原理和应用相结合，这是作为一本语言教程在深度和广度两方面的体现。在原理指导下的应用，才能是高效的应用；而结合应用讲解原理，才更深入、更易理解。原理和应用的完美结合，展现了 C 的语言潜能和它的强大生命力。

第三、每章后面附有基本题和程序设计题以及上机实验，总计约 230 道习题，10 个实习作业。这些习题很有特点，它不是简单的复述前面的概念，也不是前面例子的翻版，而是比例子更具创造性思考。读者如能认真完成这些练习与实验，必将大大加深和巩固所学知识，提高自己的编程能力。

由于作者水平有限，书中不足、错误之处敬请广大读者批评指正。

编 者

# 目 录

出版说明

前言

<b>第 1 章 C 语言基础知识</b> .....	1
1.1 概述 .....	1
1.2 C 语言程序的开发和运行环境 .....	4
1.2.1 Turbo C 简介 .....	4
1.2.2 Turbo C 2.0 的基本操作 .....	5
1.3 数据类型 .....	7
1.3.1 数据类型概述 .....	7
1.3.2 常量与变量 .....	8
1.4 运算符及表达式 .....	11
1.4.1 算术运算符及其表达式 .....	12
1.4.2 关系运算符及其表达式 .....	14
1.4.3 逻辑运算符及其表达式 .....	14
1.4.4 赋值运算符及其表达式 .....	15
1.4.5 逗号运算符及其表达式 .....	16
1.4.6 条件运算符及其表达式 .....	16
1.4.7 求字节数运算符 .....	17
1.4.8 位逻辑运算符 .....	17
1.4.9 位移运算符 .....	18
1.4.10 位自反运算符 .....	18
1.5 程序综合示例 .....	19
1.6 小结 .....	20
1.7 习题 .....	20
<b>第 2 章 顺序结构程序设计</b> .....	24
2.1 C 语言的基本语句 .....	24
2.1.1 表达式语句 .....	24
2.1.2 函数调用语句 .....	24
2.1.3 空语句 .....	25
2.1.4 复合语句 .....	25
2.1.5 流程控制语句 .....	25
2.1.6 语句书写要求 .....	26
2.2 程序的基本结构与算法 .....	26
2.2.1 程序的基本结构 .....	26
2.2.2 算法与描述 .....	27
2.3 输入与输出函数 .....	31

2.3.1	格式化输入、输出函数 scanf ( ) 和 printf ( )	31
2.3.2	非格式化输入输出函数	39
2.4	不同数据类型数据间的转换	42
2.4.1	自动转换	42
2.4.2	强制类型转换	43
2.5	程序综合示例	44
2.6	小结	45
2.7	习题	46
<b>第 3 章</b>	<b>分支结构程序设计</b>	<b>50</b>
3.1	if 语句	50
3.1.1	if 语句格式	50
3.1.2	if 语句执行过程	51
3.1.3	if 语句示例	51
3.2	if 语句的嵌套	52
3.2.1	if 语句的规则嵌套	52
3.2.2	if 语句的任意嵌套	55
3.3	switch 语句	56
3.3.1	switch 语句的格式	56
3.3.2	switch 语句的执行过程	57
3.3.3	switch 语句示例	58
3.4	程序综合示例	59
3.5	小结	63
3.6	习题	64
<b>第 4 章</b>	<b>循环程序设计</b>	<b>69</b>
4.1	循环程序的基本组成	69
4.2	for 语句	69
4.2.1	for 语句格式	69
4.2.2	for 语句执行过程	70
4.2.3	for 语句程序示例	70
4.3	while 语句	72
4.3.1	while 语句格式	72
4.3.2	while 语句执行过程	73
4.3.3	while 语句程序示例	73
4.4	do...while 语句	74
4.4.1	do...while 语句格式	74
4.4.2	do...while 语句执行过程	74
4.4.3	do...while 语句程序示例	75
4.5	循环控制 break 语句和 continue 语句	76
4.5.1	break 语句	76

4.5.2	continue 语句	77
4.6	循环的嵌套	77
4.7	转向语句和标号	79
4.8	程序综合示例	80
4.9	小结	83
4.10	习题	84
<b>第5章</b>	<b>函数</b>	<b>89</b>
5.1	模块化程序设计概述	89
5.2	用户自定义函数和库函数	90
5.3	函数的定义和调用	90
5.3.1	函数的定义	90
5.3.2	函数调用	92
5.3.3	函数说明	93
5.4	变量的作用域和存储类型	95
5.5	函数间的数据传递	99
5.5.1	利用数据复制方式实现函数间的数据传递	99
5.5.2	利用函数返回实现函数间数据传递	100
5.5.3	利用全局变量实现函数间传递数据	101
5.6	函数嵌套调用和递归调用	102
5.6.1	递归函数的概念	102
5.6.2	递归程序设计	102
5.7	内部函数和外部函数	103
5.7.1	C语言源文件	103
5.7.2	内部函数	104
5.7.3	外部函数	104
5.7.4	多个源文件的连接	105
5.8	程序综合示例	105
5.9	小结	109
5.10	习题	110
<b>第6章</b>	<b>编译预处理</b>	<b>117</b>
6.1	概述	117
6.2	宏定义	117
6.2.1	无参宏定义	117
6.2.2	有参宏定义	118
6.3	文件包含	119
6.4	条件编译	121
6.5	程序综合示例	122
6.6	小结	125
6.7	习题	125

<b>第7章 数组</b> .....	128
7.1 概述 .....	128
7.2 一维数组 .....	128
7.2.1 一维数组的定义 .....	128
7.2.2 一维数组元素的引用 .....	129
7.2.3 一维数组元素的存储 .....	129
7.2.4 一维数组的初始化 .....	129
7.2.5 一维数组程序举例 .....	130
7.3 多维数组 .....	131
7.3.1 二维数组的定义 .....	131
7.3.2 二维数组元素的引用 .....	132
7.3.3 二维数组元素的存储 .....	132
7.3.4 二维数组初始化 .....	133
7.3.5 二维数组程序举例 .....	134
7.4 字符数组与字符串 .....	135
7.4.1 字符数组的定义 .....	135
7.4.2 字符数组的初始化 .....	136
7.4.3 字符串的存储 .....	136
7.4.4 字符数组的引用 .....	137
7.4.5 字符数组程序举例 .....	137
7.4.6 字符串处理函数 .....	139
7.5 数组作为函数参数 .....	141
7.6 程序综合示例 .....	142
7.7 小结 .....	146
7.8 习题 .....	147
<b>第8章 指针</b> .....	153
8.1 指针与地址的概念 .....	153
8.1.1 内存地址——内存中存储单元的编号 .....	153
8.1.2 寻址 .....	153
8.1.3 指针与指针变量 .....	153
8.2 指针与变量 .....	154
8.2.1 指针指向变量 .....	154
8.2.2 指针变量的定义 .....	155
8.2.3 指针变量的赋值 .....	155
8.3 指针运算 .....	157
8.3.1 指针的算术运算 .....	157
8.3.2 指针的关系运算 .....	158
8.3.3 指针的赋值运算 .....	158
8.4 指针与数组 .....	159

8.4.1	数组指针	159
8.4.2	字符指针	164
8.4.3	指针数组	164
8.5	指针与函数	166
8.5.1	函数指针	166
8.5.2	指针函数	167
8.5.3	利用指针进行函数间的数据传递	168
8.5.4	命令行参数	170
8.6	程序综合示例	170
8.7	小结	179
8.8	习题	180
<b>第9章</b>	<b>结构体、联合体与枚举类型</b>	<b>186</b>
9.1	概述	186
9.2	结构体与变量	186
9.2.1	结构体的定义	186
9.2.2	结构体变量的定义	187
9.2.3	结构体变量的引用	188
9.2.4	结构体变量初始化	188
9.3	结构体数组	190
9.3.1	结构体数组定义	190
9.3.2	结构体数组初始化	191
9.3.3	结构体数组成员的引用	191
9.4	结构体指针	192
9.4.1	结构体指针的定义及赋值	192
9.4.2	指向结构体变量的指针	193
9.4.3	指向结构体数组的指针	194
9.5	结构体与函数	194
9.6	利用结构体变量构成链表	196
9.6.1	链表概念	196
9.6.2	动态分配内存	196
9.6.3	建立链表和显示链表	197
9.6.4	链表的插入与删除	198
9.7	联合体	200
9.8	枚举类型	201
9.9	用户定义类型	203
9.10	程序综合示例	204
9.11	小结	206
9.12	习题	207
<b>第10章</b>	<b>文件</b>	<b>211</b>

10.1	概述	211
10.2	文件的打开与关闭	211
10.2.1	文件指针	211
10.2.2	文件的打开	212
10.2.3	文件的关闭	212
10.3	文件的顺序读写	213
10.3.1	单个字符读写函数	213
10.3.2	字符串读写函数	214
10.3.3	格式化读写函数	215
10.3.4	数据块读写函数	216
10.4	文件的随机读写	218
10.5	文件的错误检测	220
10.6	程序综合示例	220
10.7	小结	223
10.8	习题	223
<b>第 11 章</b>	<b>综合应用</b>	<b>228</b>
11.1	确定软件功能	228
11.2	定义核心数据结构	228
11.3	程序的模块划分	230
11.3.1	初始化模块	230
11.3.2	游戏主控模块	230
11.3.3	游戏结束	234
11.4	源程序	234
11.5	调试	245
11.5.1	编译时常见错误	245
11.5.2	连接时的常见错误	250
11.5.3	运行时的常见错误	250
<b>附录</b>		<b>251</b>
附录 A	集成环境 (IDE) 下各选项的作用	251
附录 B	Turbo C2.0 常用库函数	256
附录 C	ASCII 代码表	265
附录 D	错误信息中英文对照	266
附录 E	C 语言中的关键字	270
附录 F	C 语言程序设计考试要求	271
附录 G	习题参考答案	273
附录 H	上机实验	278
<b>参考文献</b>		<b>284</b>

# 第 1 章 C 语言基础知识

## 1.1 概述

计算机的程序设计语言经历了从机器语言、汇编语言到高级语言这样一个发展过程。C 语言属于高级语言，但是由于 C 语言同时具有低级语言的特性，所以许多人又把 C 语言称为中级语言。

### 1.1.1 C 语言的发展

计算机诞生初期，人们指挥计算机工作直接用计算机能识别的二进制指令来书写。这种程序设计语言就是机器语言。机器语言是直接对计算机硬件产生作用的，所以不同型号的计算机的机器语言不一样，这使得机器语言很难被人们掌握和推广，因此一般只有少数计算机专家或专业技术人员才使用。为减轻人们在编程中的劳动强度，人们开始采用“助记符号”来代替 0, 1 码编程的二进制指令。比如，在机器语言中，二进制数 10110110 代表加法运算，可以用英文单词 ADD 代表，它比一长串二进制数要简洁直观并且易于记忆。这种用助记符号描述的指令系统，称为汇编语言。

由于汇编语言和机器语言都是面向机器的，随机而异，程序可移植性差，因此人们发明了与人类自然语言非常接近的高级程序设计语言。高级语言不仅易学、易用，而且写出的程序更加简练，便于在不同型号的机器上运行，即程序可移植性好。但是用高级语言编写的程序不能被计算机直接理解和运行，而必须先由这种语言的编译或解释程序翻译成机器指令，然后再让计算机执行。人们设想能否找到一种既具有高级语言特性，又具有低级语言特性的语言，集它们的优点于一身。于是，C 语言就在这样的情况下诞生了。

C 语言的前身是英国剑桥大学 1963 年推出的复合程序设计语言 CPL (Combined Programming Language)，它是仿造 ALGOL 60 而设计出来的语言，由于 CPL 庞大而复杂，不便于使用，推广受到了很大限制。于是 Martin Richards 对 CPL 语言作了简化，推出了 BCPL (BASIC CPL) 语言。70 年代初，美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础，作了进一步改进，提出了一种 B 语言(取 BCPL 的第一个字母)。之后贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出了 C 语言(取 BCPL 的第二个字母)。最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种语言而设计的。1973 年，Ken Thompson 和 D.M.Ritchie 两人合作，将由他们在 1969 年用汇编语言研制的 UNIX 操作系统的 90% 以上代码用 C 改写。从此，UNIX 操作系统和 C 语言像一对孪生姐妹，以自己崭新的面貌引起人们广泛的关注。后经不断地改进与完善，迅速成长、成熟，并显示了强大的生命力。

### 1.1.2 C 语言的特点

C 语言不仅具有高级语言的易学、易用、可移植性强等特点，而且具有低级语言的执行

效率高、可对硬件直接操作等优点。这一重要特征可以使 C 语言大大提高运算效率，其在硬件、软件中的地位如图 1-1 所示。目前它已成为计算机程序设计语言的主流语种，许多系统软件包括 Windows 95 操作系统大部分都是用 C 语言编写的。

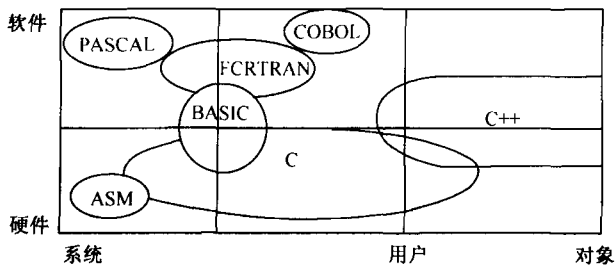


图 1-1 C 语言在硬件、软件中的地位

作为高层开发工具的 C 语言，其明显的特点主要体现在以下几方面：

(1) C 语言是一种结构化程序设计语言，适合大型项目的编程和管理。

(2) 语言简捷易读。C 语言只有 32 个关键字（见附录 E），所谓关键字就是已被 C 语言本身使用，不能作其他用途使用的字。例如关键字不能用作变量名、函数名等。它们构成了 C 语言的全部指令。

(3) C 语言具有丰富的数据类型，可实现各种复杂的数据结构。

(4) C 语言具有丰富的运算能力，运算符有 34 种之多。

(5) 可直接对硬件进行操作，具有低级语言的许多功能。

(6) 移植性好。C 语言程序本身不依存于机器硬件，可从一种环境移植到另一种环境中运行。

下面通过一些简单的 C 程序，来分析 C 语言程序的结构特点。

**【例 1-1】** 一个简单程序示例。

```
main ( )          /*主函数*/
{
    printf ("I Learn Turbo C\n"); /*输出一串字符,\n 表示换行*/
}
```

程序运行结果：

I Learn Turbo C

**【例 1-2】** 一个简单的多行输出程序。

```
main ( )
{
    printf ("    咏柳\n");
    printf ("碧玉妆成一树高, \n");
    printf ("万条垂下绿丝绦. \n");
    printf ("不知细叶谁裁出? \n");
    printf ("二月春风似剪刀. \n");
}
```

运行结果：

咏柳

碧玉妆成一树高，  
万条垂下绿丝绦。  
不知细叶谁裁出？  
二月春风似剪刀。

【例 1-3】 两个数据输入、输出示例。

```
main ( )                /*主函数*/
{
int x, y; /*定义变量*/
printf ("输入两个数： ");
scanf ("%d, %d", &x, &y); /*输入变量 x, y 的值*/
printf ("x=%d y=%d", x, y); /*输出 x、y 的值*/
}
```

运行结果：

输入两个数： 3, 5 ✓

x=3 y=5

说明：本书用矩形框描述程序的运行结果，其中以下划线“  ”及“✓”表示用户输入。例如，【例 1-3】程序运行结果应为，屏幕显示：输入两个数：，用户输入：3, 5 回车，屏幕换行显示：x=3 y=5。

C 语言程序具有以下主要结构特点：

- (1) 所有的 C 语言程序都是由一个或多个函数组成，即函数是组成 C 语言程序的基本单位。
- (2) C 语言程序总是从主函数 main ( ) 开始执行，不论 main ( ) 函数放在程序的什么位置。
- (3) C 语言程序中的语句用分号“;”终止。因此，程序书写格式灵活，允许一行中多个语句，也可以将一条语句分成几行书写。
- (4) 用一对大括号{ }表示程序的结构层次，一个完整的程序模块用一对大括号表示其范围。
- (5) C 语言严格区分大小写英文字母。
- (6) C 语言程序可以使用注释。注释部分格式：

```
/*注释内容*/
```

C 语言可对程序的任何部分作注释。注释可分为若干行，但不允许嵌套。编译系统不对注释部分进行编译，这一点在调试程序时很有用。

### 1.1.3 C 语言标识符

标识符是一种特定的字符序列，用来标记：变量名、常量名、数组名、函数名、结构名、标号和文件名等。

C 语言标识符命名规则：

(1) 只能由 1 至 32 个英文字母、数字和下划线组成。必须以英文字母或下划线开头。系统内部使用了一些用下划线开头的标识符，为防止与用户自己定义的标识符冲突，建议用户尽量避免使用下划线开头来定义标识符。

(2) 严格区分大小写字母，例如：NAME 和 name 是两个不同的标识符。

(3) 关键字不能作为标识符。

**【例 1-4】** 找出下列符号中合法的标识符。

3a6b、a.com、Abc、red\_yellow、\_3x4y、\$abc、FLOAT、float

分析：合法的标识符有 Abc、red\_yellow、\_3x4y、FLOAT。

在定义标识符时，建议遵循下列原则：

(1) 标识符要“见名知义”，以增加可读性。如：name、age、salary，可以用来定义姓名、年龄、工资。

(2) 习惯变量名、函数名用小写，符号常量用大写。

(3) 尽量避免在容易出现混淆的地方出现容易认错的字符，如：0（数字）与 o（字母）、1 与 l 等。

## 1.2 C 语言程序的开发和运行环境

C 语言有多种不同的编译器，Turbo C 是其中比较有代表性的一种。本书以 Turbo C 为“集成开发”环境，介绍 C 语言程序的开发和运行。

### 1.2.1 Turbo C 简介

Turbo C 是美国 Borland 公司的产品，该公司曾开发了一系列 Turbo 产品，如 Turbo BASIC、Turbo PASCAL 等等。1987 年公司首推 Turbo C 1.0，将文本编辑、连接、运行等集于一体，简称 IDE (Integrated Development Environment) 集成环境。1988 年该公司又推出了 Turbo C 1.5 版本，增加了图形库和文本窗口函数库等。1989 年推出了 Turbo C 2.0 增加了查错及“协处理器”的仿真等功能。

Turbo C 2.0 的软硬件配置如下：

(1) Turbo C 2.0 可运行于 IBM-PC 系列微机，包括 XT，AT 及 IBM 兼容机。此时要求 DOS 2.0 或更高版本支持，并至少需要 448KB 的 RAM，可在任何彩、单色 80 列监视器上运行。支持数学协处理器芯片，也可进行浮点仿真，这将加快程序的执行。

(2) Turbo C 2.0 有六张低密软盘（或两张高密软盘）。下面对 Turbo C 2.0 的主要文件作一简单介绍：

INSTALL.EXE 安装程序文件；

TC.EXE 集成编译；

TCINST.EXE 集成开发环境的配置设置程序；

TCHELP.TCH 帮助文件；

THELP.COM 读取 TCHHELP.TCH 的驻留程序；

README 关于 Turbo C 的信息文件；

TCCONFIG.EXE 配置文件转换程序；

MAKE.EXE 项目管理工具；  
TCC.EXE 命令行编译；  
TLINK.EXE Turbo C 系列连接器；  
TLIB.EXE Turbo C 系列库管理工具；  
C?.OBJ 不同模式启动代码；  
C?.LIB 不同模式运行库；  
GRAPHICS.LIB 图形库；  
EMU.LIB 8087 仿真库；  
FP87.LIB 8087 库；  
\*.H Turbo C 头文件；  
\*.BG 不同显示器图形驱动程序；  
\*.C Turbo C 例程序（源文件）。

其中：上面的?分别为：

T Tiny（微型模式）；  
S Small（小模式）；  
C Compact（紧凑模式）；  
M Medium（中型模式）；  
L Large（大模式）；  
H Huge（巨大模式）。

## 1.2.2 Turbo C 2.0 的基本操作

### 1. 启动 Turbo C

Turbo C 一般安装在 C:\TC 子目录下，在 C:\ 下装有启动 UC DOS 中文系统的批命令文件 UP.BAT，则启用 Turbo C 的操作步骤如下：

（1）调入 UC DOS 中文系统（若程序不涉及汉字的输入输出，可以跳过本步骤），键入命令：

```
C:\>UP 或 C:\UCDOS\up
```

（2）进入 TC 子目录，键入命令：

```
C:\>CD TC
```

（3）运行 TC 集成环境，键入命令：

```
C:\TC>TC
```

此时出现如图 1-2 所示的屏幕格式。

### 2. 编辑源程序

在出现图 1-2 后，按 F10 键，在主菜单项中选择 File 菜单的新文件 New 子功能如图 1-3 所示，并按回车键，屏幕在该编辑器上出现“-”光标，此时可以输入并编辑 C 语言源程序了，其文件扩展名为“C”。

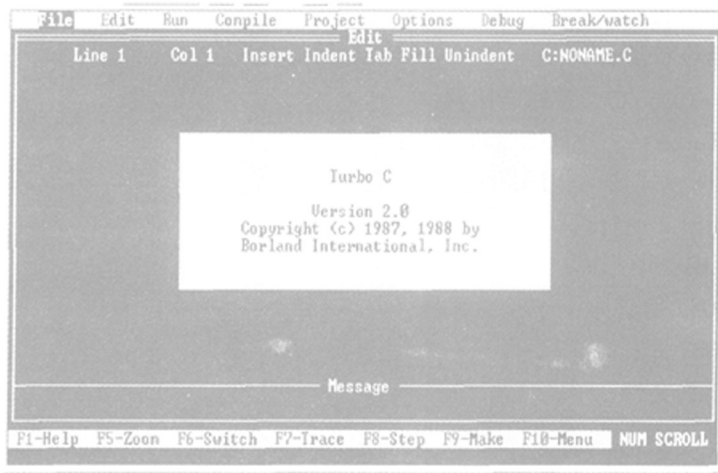


图 1-2 TURBO C 菜单示意图

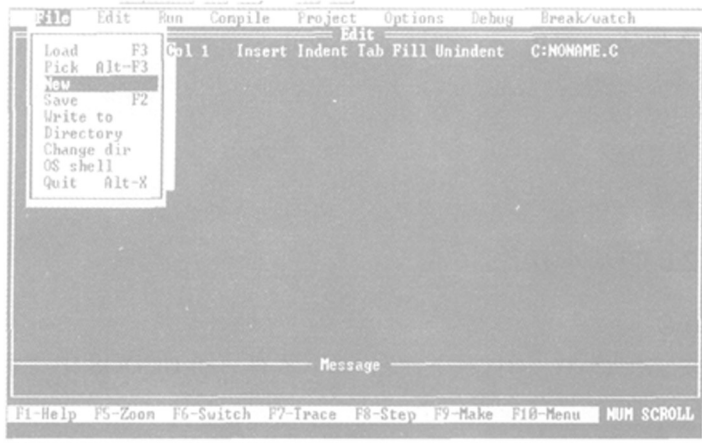


图 1-3 File 菜单示意图

### 3. 编译源程序生成目标程序

对源程序进行语法检查和翻译，一次性的将源程序翻译成机器代码，可以在“Compile”菜单中选择“Compile to obj”子功能，生成与源程序同名的“.obj”目标文件。在编译过程中，若出现错误，则需要察看错误信息后按 F6 键重新编辑，再编译直到正确为止。

### 4. 链接目标程序生成可执行程序

在“Compile”菜单中选择“Make EXE File”子功能，链接各目标程序文件，生成同名的“.exe”可执行文件。

### 5. 执行程序

在“Run”菜单中选择“Run”子功能，即可在 Turbo C 环境中执行程序，察看结果用 Alt-F5 组合键。也可以脱离 C 语言编译系统，在操作系统下键入文件名直接运行。

Turbo C2.0 提供了一个编译、连接、运行的批处理命令：Ctrl-F9 组合键。

## 1.3 数据类型

### 1.3.1 数据类型概述

数据是描述客观事物的数、字符以及所有能输入到计算机中并被计算机程序加工处理的符号的集合。C 语言的数据结构是通过数据的存储类型、数据类型的定义实现的。存储类型说明数据在内存中的存储位置，将在第五章函数部分作详细介绍。数据类型说明数据占内存单元的字节数，即占内存的大小，本节详细讨论数据类型。

C 语言提供了丰富的数据类型，如图 1-4 所示。

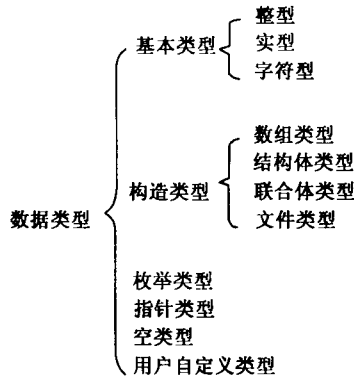


图 1-4 C 语言数据类型

其中基本类型是基础，由它可以构造其他复杂的数据类型，例如：数组类型就是由若干个具有相同基本数据类型的数据构成的。有关构造类型的内容将在以后章节中结合应用陆续介绍，本章重点介绍基本类型。

基本类型有整型、实型、字符型。

整型有短整型、整型、长整型、无符号短整型、无符号整型、无符号长整型；实型有单精度实型和双精度实型。对每一种类型由系统提供相应的关键字来标识，用户不能随意修改。基本类型关键字及所占字节数如表 1-1 所列。

表 1-1 各种基本类型关键字及所占字节数

类型关键字	所占字节数	说 明
int	2	基本整型（简称整型）
short	2	短整型
long	4	长整型
unsigned	2	无符号整型
unsigned short	2	无符号短整型
unsigned long	4	无符号长整型
float	4	单精度实型
double	8	双精度实型
char	1	字符型