



面向21世纪高等院校计算机基础系列规划教材

C语言程序设计

北京希望电子出版社 总策划

黄纯国 匡松 主 编

吕峻闽 陈 康 杨大友 副主编



 科学出版社
www.sciencep.com



面向21世纪高等院校计算机基础系列规划教材

C语言程序设计

北京希望电子出版社 总策划

黄纯国 匡松 主 编

吕峻闽 陈 康 杨大友 副主编

江苏工业学院图书馆
藏书章

 科学出版社
www.sciencep.com

内 容 简 介

本书结合全国计算机等级考试二级 C 语言考试大纲编写,系统介绍了 C 语言的基本语法、程序结构和程序设计方法。

本书内容包括 C 语言的基本数据类型、运算符与表达式、输入与输出语句、结构控制语句、数组、函数、指针、预处理、构造型数据结构、位运算、文件等。每章都根据学生的实际情况编排了大量的习题和实验,以加深和巩固所学知识,从而帮助读者在初步掌握 C 语言的基础上,提高程序设计的能力。

本书适合作为高等院校学生学习 C 语言程序设计的教材,也可作为自学 C 语言程序设计人员的参考用书。

需要本书或技术支持的读者,请与北京清河 6 号信箱(邮编:100085)发行部联系,电话:010-82702660, 62978181(总机),传真:010-82702698, E-mail: tbd@bhp.com.cn。

图书在版编目(CIP)数据

C 语言程序设计 / 黄纯国, 匡松主编. —北京: 科学出版社, 2006.7

(面向 21 世纪高等院校计算机基础系列规划教材)

ISBN 7-03-016716-3

I. C... II. ①黄... ②匡... III. C 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 157624 号

责任编辑: 方红琴 / 责任校对: 马 君
责任印刷: 媛 明 / 封面设计: 刘孝琼

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>

北京媛明印刷厂印刷

科学出版社发行 各地新华书店经销

*

2006 年 7 月第 一 版 开本: 787×1092 1/16
2006 年 7 月第一次印刷 印张: 17 1/2
印数: 1—3 500 字数: 396 000

定价: 29.00 元

面向21世纪高等院校计算机基础系列规划教材编委会

主任：沈复兴 全国高等师范学校计算机教育研究会副理事长
北京师范大学信息科学学院院长

副主任：吴 跃 教育部计算机科学与技术教学指导委员会委员
电子科技大学计算机学院、软件学院院长

何炎祥 武汉大学计算机学院院长

桂卫华 中南大学信息科学与工程学院院长

匡 松 全国高等院校计算机基础教育研究会理事
西南财经大学经济信息工程学院副院长

黄纯国 西南财经大学电子商务学院副院长

杨宪泽 西南民族大学计算机科学与技术学院院长

陆卫民 中国科学出版集团北京希望电子出版社社长

委员：（按姓氏笔画为序）

王江晴	王学平	甘 玲	邓志华	吕峻闽	李华贵	李节阳
李建平	肖阳春	何登旭	张裔智	杨 波	杨大友	杨清平
邹显春	武兆辉	周鑫淼	周学文	郑明红	洪汝渝	徐建军
唐光海	曹永存	韩延明	谢志龙	覃 俊	雷开彬	蔡乐才

秘 书：徐建军

本书编委会成员

(按姓氏笔画为序)

宁 涛	李长松	匡 松	吕峻闽	吴希敏
张 英	陈 康	陈 斌	杨大友	杨 城
金 星	姚一永	赵 华	袁 勋	黄纯国
梁庆龙	梁浴文	韩延明	谢志龙	

前 言

C 语言在计算机程序设计领域应用非常广泛,它具有功能丰富、语句简洁、使用方便、语法灵活、数据结构多样、能对硬件进行操作、高移植性和通用性等诸多优点。C 语言既有高级语言的特点,又有汇编语言等低级语言的特点,因此被称为“中级语言”,它已经成为编制系统软件和应用软件的首选语言。当今流行的面向对象程序设计语言 C++、Java 就来源于 C 语言。

C 语言是我国各高校普遍开设的一门重要的计算机基础课程,同时也是计算机专业学生学习程序设计语言的必修课程。在编写本书过程中,作者结合自身多年从事 C 语言教学的经验,理论联系实际,力求通俗易懂。在体系结构安排上尽可能将概念、知识点与例题结合起来,每章的开始都指出本章的学习目标,每章结束后对本章内容进行小结。在例题的选择上具有很强的针对性,通过一些典型程序将前后的一些知识点联系起来,使读者在对比中理解各种实现方式的特点和异同,能够融会贯通、举一反三。每章还配有适量的习题,让学生加深和巩固所学知识,提高学生的编程能力,培养良好的编程风格和习惯。

本书共 12 章,包括 C 语言概述、基本数据类型、运算符与表达式、数据的输入与输出、结构控制语句、数组、函数、指针、预处理、结构体与共用体、位运算、文件等内容。从 C 语言的基本数据元素、基本语句和结构控制语句、构造数据类型的定义和使用、函数的定义和调用,到指针和文件的灵活运用等方面进行了由浅入深的讲解。本书的特点是结构合理、层次分明、例题丰富、通俗易懂、实用性强,适合于初学者使用。

黄纯国副教授和匡松教授设计了本书的结构,并做了全书的统稿工作。第 1、2、3、4、11 章由吕峻闽编写,第 6、8、9、10 章由陈康编写,第 5、7、12 章由杨大友编写。此外,本书在编写过程中得到了西南财经大学天府学院其他教师的帮助,并对本书提出了许多宝贵意见,在此向他们表示衷心感谢。

由于编者水平有限,书中难免有疏漏之处,敬请广大读者批评指正。

编 者

目 录

第1章 C语言概述.....1	3.5.3 逗号运算符与表达式..... 41
1.1 C语言概述.....1	3.5.4 强制类型转换运算符与表达式..... 42
1.1.1 C语言的发展.....1	3.6 运算符的优先级..... 43
1.1.2 C语言特点.....1	小结..... 44
1.1.3 Turbo C 2.0 介绍.....2	习题..... 44
1.2 第一个C语言程序.....7	第4章 数据的输入与输出..... 48
1.2.1 C语言程序的编制运行过程.....7	4.1 数据的输出函数..... 48
1.2.2 C语言程序的结构.....8	4.1.1 按格式输出函数 printf()..... 48
1.2.3 C语言程序设计基础.....11	4.1.2 标准字符输出函数 putchar()... 51
小结.....12	4.1.3 字符串输出函数 puts()..... 51
习题.....13	4.2 数据输入函数..... 52
第2章 基本数据类型.....16	4.2.1 按格式输入函数 scanf()..... 52
2.1 C语言的数据类型.....16	4.2.2 标准字符输入函数 getchar()..... 55
2.2 常量与变量.....17	4.2.3 字符串输入函数 gets()..... 56
2.3 整型数据.....20	小结..... 57
2.3.1 整型常量.....20	习题..... 57
2.3.2 整型变量.....21	第5章 结构控制语句..... 62
2.4 实型数据.....24	5.1 程序的基本控制结构..... 62
2.4.1 实型常量.....24	5.1.1 顺序结构..... 62
2.4.2 实型变量.....24	5.1.2 选择结构..... 64
2.5 字符型数据.....25	5.1.3 循环结构..... 65
2.5.1 字符型常量.....25	5.2 条件选择语句..... 65
2.5.2 字符型变量.....26	5.2.1 if语句的形式..... 66
2.5.3 字符串常量.....28	5.2.2 if语句的嵌套..... 67
小结.....29	5.3 多路分支语句..... 69
习题.....29	5.4 循环语句..... 72
第3章 运算符与表达式.....33	5.4.1 while型循环语句..... 73
3.1 概述.....33	5.4.2 do...while型循环语句..... 73
3.1.1 表达式的组成.....33	5.4.3 for型循环语句..... 74
3.1.2 运算符的分类.....33	5.4.4 goto语句..... 76
3.2 算术运算符与表达式.....34	5.4.5 continue语句和 break语句..... 77
3.3 关系运算符与表达式.....36	5.4.6 循环嵌套..... 79
3.4 逻辑运算符与表达式.....37	小结..... 82
3.5 其他运算符与表达式.....39	习题..... 82
3.5.1 赋值运算符与表达式.....39	第6章 数组..... 88
3.5.2 条件运算符与表达式.....41	6.1 一维数组..... 88

6.1.1 一维数组的定义.....	88	8.2 指针与数组.....	163
6.1.2 一维数组的初始化.....	89	8.2.1 指向一维数组的指针.....	163
6.1.3 一维数组元素的使用.....	90	8.2.2 指向二维数组的指针.....	167
6.1.4 应用举例.....	92	8.2.3 指向字符串的指针.....	173
6.2 字符数组.....	96	8.3 指向特殊类型数据的指针变量.....	175
6.2.1 字符数组的定义及字符串.....	96	8.3.1 指针数组.....	175
6.2.2 字符数组的初始化.....	97	8.3.2 指向指针的指针.....	178
6.2.3 字符数组的使用.....	97	8.4 指针与函数.....	180
6.2.4 字符串处理函数.....	99	8.4.1 指针变量作为函数参数.....	180
6.2.5 应用举例.....	103	8.4.2 返回指针的函数.....	183
6.3 二维数组.....	105	8.4.3 指向函数的指针.....	186
6.3.1 二维数组的定义.....	105	小结.....	189
6.3.2 二维数组的初始化.....	106	习题.....	189
6.3.3 二维数组的使用.....	107	第9章 预处理.....	197
6.3.4 应用举例.....	108	9.1 宏定义.....	197
小结.....	112	9.1.1 不带参数的宏定义.....	197
习题.....	112	9.1.2 带参数的宏定义.....	198
第7章 函数.....	118	9.2 文件包含.....	199
7.1 用户自定义函数.....	118	9.3 条件编译.....	200
7.1.1 函数的定义.....	119	9.3.1 #ifdef.....	200
7.1.2 函数的调用.....	120	9.3.2 #ifndef.....	201
7.1.3 函数调用中的数据传递.....	122	9.3.3 #if.....	202
7.2 函数的嵌套调用和递归调用.....	130	小结.....	203
7.2.1 函数的嵌套调用.....	130	习题.....	203
7.2.2 函数的递归调用.....	131	第10章 构造型数据结构.....	208
7.3 变量的作用域和生存期.....	135	10.1 结构体.....	208
7.3.1 局部变量和全局变量.....	135	10.1.1 概述.....	208
7.3.2 变量的生存期.....	139	10.1.2 结构体的定义.....	209
7.4 函数的作用范围.....	144	10.1.3 结构体变量的使用.....	211
7.4.1 内部函数.....	144	10.1.4 结构体数组.....	212
7.4.2 外部函数.....	144	10.1.5 指向结构体的指针.....	214
7.5 命令行参数.....	145	10.1.6 结构体变量作为参数 在函数之间进行传递.....	216
7.6 多个文件的连接和运行.....	146	10.2 指针与链表.....	218
小结.....	147	10.2.1 动态数据结构与 动态内存分配.....	218
习题.....	148	10.2.2 链表的定义.....	219
第8章 指针.....	157	10.2.3 链表的操作.....	221
8.1 指针与指针变量.....	157	10.3 共用体.....	227
8.1.1 地址、指针与指针变量.....	157		
8.1.2 指针变量的使用.....	158		

10.3.1 共用体的定义.....	227
10.3.2 共用体变量的使用.....	228
10.4 枚举类型.....	230
10.4.1 枚举类型的定义.....	230
10.4.2 枚举类型变量的使用.....	230
10.5 自定义类型.....	232
小结	233
习题	234
第 11 章 位运算	241
11.1 位运算符.....	241
11.1.1 按位与.....	241
11.1.2 按位或.....	241
11.1.3 按位异或.....	242
11.1.4 按位取反.....	243
11.1.5 按位左移.....	243
11.1.6 按位右移.....	243
11.1.7 位运算复合赋值运算符.....	244
11.2 位段.....	245
小结	246
习题	246
第 12 章 文件	252

12.1 文件概述.....	252
12.1.1 文件的存储方式.....	252
12.1.2 文件类型指针 (FILE 类型指针)	253
12.2 文件的打开和关闭.....	253
12.2.1 打开文件函数 fopen.....	253
12.2.2 关闭文件函数 fclose	254
12.3 文件的读写.....	255
12.3.1 字符的读取和写入.....	255
12.3.2 字符串的读取和写入.....	256
12.3.3 按格式读取和写入.....	258
12.4 随机文件的读写.....	259
12.4.1 随机文件的读取.....	259
12.4.2 随机文件的写入.....	260
12.5 文件定位函数.....	261
12.5.1 rewind 函数.....	261
12.5.2 fseek 函数.....	262
12.5.3 ftell 函数.....	263
小结.....	263
习题.....	263

第 1 章 C 语言概述

C 语言不仅功能强大、表达能力强、使用起来灵活方便、编译效率高、可移植性好,而且具备高级语言和低级语言的优点。自从 C 语言出现至今已经有三十多年的时间,是使用最广泛的程序设计语言之一,它既可以用来编写应用软件,又非常适合编写系统软件。

本章首先介绍 C 语言的基本发展历史、版本情况以及主要特点,然后介绍 C 语言程序的基本编制环境和步骤,使大家在简单程序的编制过程中初步掌握 C 语言程序的构成、特点和编制过程。

1.1 C 语言概述

1.1.1 C 语言的发展

C 语言是 1972 年由美国的 Dennis Ritchie 设计发明的,并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。它由早期的编程语言 BCPL(Basic Combined Programming Language)发展演变而来。

在 1970 年,AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出较先进的并取名为 B 的语言,并用 B 语言写了第一个 UNIX 操作系统。1972 年,贝尔实验室的 Ritchie 在 B 语言的基础上设计出了 C 语言,取的是 BCPL 语言的第二个字母。到了 1973 年,两人又用 C 语言一起改写了 UNIX 操作系统 90%以上的代码,也就是 UNIX 第 5 版本。1978 年,Ritchie 和 Brain Kernighan 编写了《The C Programming Language》,并于 1988 年作了修订,该书就是 C 语言版本的基础,被称为标准 C。

随着计算机的日益普及,出现了许多 C 语言版本,正是 C 语言的高可移植性的特点,使得 UNIX 系统在 AT&T、VAX 等计算机上能够迅速实现。但是由于没有统一的标准,使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况,美国国家标准化协会(ANSI)为 C 语言制定了一套 ANSI 标准,成为现行的 C 语言标准,包括了 1983 年和 1987 年两个版本。到了 1990 年,国际标准化组织(International Standard Organization,简称 ISO)接受了 87 ANSI C 作为 ISO C 的标准,即 ISO 9899—1990。目前比较流行的 C 语言的编译系统都是以此作为基础,如 Turbo C、Borland C、Microsoft C 等。

1.1.2 C 语言特点

C 语言发展如此迅速,而且成为最受欢迎的语言之一,主要因为它具有强大的功能。C 语言的特点如下。

(1) C 语言提供的语句简洁,使用方便,格式紧凑,语法灵活。C 语言一共有 32 个关键字,9 种控制语句,语句简练,书写自由。以下关键字是由系统定义的,不能用作其他定义。

与数据类型相关的关键字:

char int short long signed unsigned float

double enum struct union typedef void

与存储类型相关的关键字:

auto register static extern

与控制语句相关的关键字:

if else switch case default do
while for break continue goto return

其他的关键字:

const sizeof volatile

下面是 C 语言中的 9 种控制语句:

if(...) ... else ... switch(...) ... case for(...) ... while(...) ...
do ... while(...) continue break goto return

(2) C 语言的运算符十分丰富,一共有 34 种运算符,包括算术运算符、关系运算符、逻辑运算符、位运算符、赋值运算符、指针、条件运算符、逗号运算符、下标运算符、类型转换运算符等多种类型。

(3) 数据结构多样,有整型、实型、字符型、枚举类型等基本类型,有数组、结构体、共用体等构造类型以及指针类型,还为用户提供了自定义数据类型,特别是引入了指针概念,能够实现复杂的数据结构。

(4) C 语言的控制语句形式多样、使用方便。有两路分支、多路分支和循环结构几种控制语句,便于结构化模块的实现和控制,便于程序的编制和维护。

(5) C 语言是一种模块化的程序设计语言,采用自顶向下、逐步求精的结构化程序设计方法,各模块功能独立,以函数形式编制,通过函数之间的相互调用和数据传递,实现系统整体的功能要求。这样,把大型系统的实现化整为零,便于分工合作以及共享。

C 语言程序中,所有的关键字都是小写字母,在 C 语言程序里大小写是敏感的;一般习惯使用小写字母作为变量名称和字符串常量。C 语言书写格式自由,不使用行号,可以使用空格和空行,但习惯上采用的锯齿形书写格式。良好的书写习惯,是优秀的程序员必备的素质之一。例如,使用“Tab”键缩进,花括号对齐,在适当的地方添加空行,并添加必要的注释。

(6) C 语言可以直接访问地址、进行位运算,从而能对硬件进行操作,因此 C 语言既具有高级语言编写简单方便、便于理解的优点,又具有低级语言与硬件结合紧密的优点。因此 C 语言被称为介于高级语言和低级语言之间的中级语言。

(7) C 语言具有很强的移植性,由 C 语言编写的程序基本不用太多的修改就可以用于不同型号的计算机上,程序和硬件的匹配由 C 语言的编译程序完成,同时也可以在各种操作系统下使用。

(8) C 语言具有很好的通用性,既可以用于编写应用软件,也适合编写系统软件。例如 UNIX 操作系统的源代码就是用 C 语言编制的。

1.1.3 Turbo C 2.0 介绍

1. Turbo C 的发展

Turbo C 是由美国的 Borland 公司开发的产品,从 1987 年推出 Turbo C 1.0 以来,该公

司相继推出了一系列 Turbo C 版本。Turbo C 1.0 拥有一套集成开发环境，以菜单的方式把程序编制过程中要使用的命令分门别类地集中起来，提供给用户方便清晰的编程环境。1988 年，Borland 公司又推出 Turbo C 1.5 版，增加了图形库处理和文本窗口函数库等功能；1989 年推出 2.0，增加了纠错功能，可以在微型模式下直接生成可执行文件；此后随着图形化操作系统的普及推广，程序编制的模式发生了较大的变化，出现了面向对象的程序设计方法；Borland 公司在 Turbo C 2.0 的基础上推出了面向对象的程序设计语言 Turbo C++，把 Turbo C 的集成开发环境与面向对象的程序设计结合起来，后来又发展为 Borland C++。

2. Turbo C 的组成

Turbo C 2.0 的系统文件比较精炼，主要包含如表 1-1 所示的一些文件。

表 1-1 Turbo C 的文件

文件名	说明	文件名	说明
INSTALL.EXE	安装程序文件	TLINK.EXE	连接器
Turbo C.EXE	集成编译文件	TLIB.EXE	库管理工具
Turbo CINST.EXE	系统配置文件	C0?.OBJ	不同模式的启动代码
Turbo CHELP.Turbo CH	帮助文件	C?.LIB	不同模式的运行库
THELP.COM	读取帮助的驻留程序	GRAPHICS.LIB	图形库
README	信息文件	EMU.LIB	8087 仿真库
Turbo CCONFIG.EXE	配置文件转换程序	FP87.LIB	8087 库
MAKE.EXE	项目管理工具	*.H	Turbo C 头文件
Turbo CC.EXE	命令行编译程序	*.BGI	图形驱动程序
*.C	Turbo C 源文件	*.OBJ	目标程序

表 1-1 中的“?”可以有的取值及其含义如下。

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)。

3. Turbo C 的安装和启动

Turbo C 2.0 的安装非常简单，运行安装盘上的 Install.exe 安装程序，屏幕上显示 3 种选择。

(1) 在硬盘上创建一个新目录来安装整个 Turbo C 2.0 系统。

(2) 对 Turbo C 1.5 更新版本。这样的安装将保留原来对选择项、颜色和编辑功能键的设置。

(3) 为只有两个软盘而无硬盘的系统安装 Turbo C 2.0。

如果按第一种选择进行安装,只要在安装过程中按提示操作,就可以顺利地进行安装。安装完毕将在 C 盘根目录下建立一个 TC 子目录, Turbo C 下还建立了两个子目录 LIB 和 INCLUDE, LIB 子目录中存放库文件, INCLUDE 子目录中存放所有头文件。

如果不是上述的安装盘,则直接将文件拷贝到相应的目录下即可。

Turbo C 2.0 可以在 DOS 和 Windows 环境下运行。运行 Turbo C 2.0 时,只要在 Turbo C 子目录下运行主程序 tc.exe 即可进入 Turbo C 2.0 集成开发环境。

在 DOS 环境下,如果当前目录是 Turbo C 子目录,键入“tc”后回车即可启动,例如:

```
C:\Turbo C>tc
```

如果当前目录是不是 Turbo C 子目录,则先改变当前目录为 Turbo C 子目录,或在“tc”之前加上路径,例如:

```
C:\>CD Turbo C
```

```
C:\Turbo C>tc
```

或

```
C:\>c:\Turbo C\tc
```

在 Windows 环境下运行,只要通过“我的电脑”或“资源管理器”,找到 Turbo C 2.0 系统文件夹的位置,找到 tc.exe 文件后双击,则可以启动进入 Turbo C 2.0 集成环境。

4. Turbo C 2.0 的集成开发环境

进入 Turbo C 2.0 集成开发环境后,主窗口如图 1-1 所示。

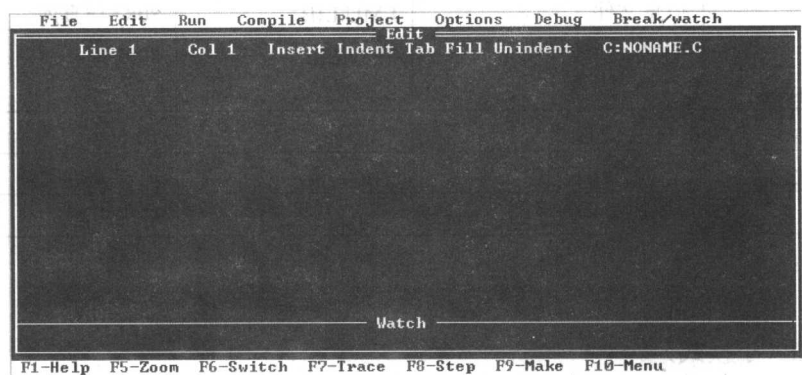


图 1-1 Turbo C 2.0 主窗口

主窗口最上面一行是 Turbo C 2.0 的主菜单,中间窗口为编辑区,接下来是信息窗口,最下面一行为参考行。C 语言程序的编辑、编译、调试以及运行都将在主窗口中进行。

选择菜单项时,可以用“Alt”键加上菜单项的第一个字母(即大写字母),或者按“F10”键,再按向左或向右的光标键。选择菜单项后,可以按向上或向下的光标键选择菜单命令,然后按回车键执行相应的功能。若要退到主菜单或从它的下一级菜单退回均可用“Esc”键。

Turbo C 2.0 的主菜单包括 8 个菜单项,下面简单介绍各菜单项的基本功能。

(1) File (文件) 菜单: 主要进行一些文件相关的操作。该菜单包括以下命令。

Load (加载): 装入一个文件,可用类似 DOS 的通配符(如*.C)来进行列表选择。也可装入其他扩展名的文件,只要给出文件名(或只给路径)即可。该命令的快捷键为“F3”。

Pick (选择): 将最近装入编辑窗口的 8 个文件列成一个表让用户选择, 选择后将该文件装入编辑区, 并将光标置于上次修改过的地方。其快捷键为 “Alt+F3”。

New (新文件): 新建一个文件, 缺省文件名为 NONAME.C, 存盘时可改名。

Save (存盘): 将编辑区中的文件存盘, 若文件名是 NONAME.C 时, 将询问是否更改文件名, 其快捷键为 “F2”。

Write to (存盘): 可由用户给出文件名将编辑区中的文件存盘, 若该文件已存在, 则询问要不要覆盖。

Directory (目录): 显示目录及目录中的文件, 并可由用户选择。

Change dir (改变目录): 显示当前目录, 用户可以改变显示的目录。

Os shell (暂时退出): 暂时退出 Turbo C 到 DOS 提示符下, 此时可以运行 DOS 命令, 若想回到 Turbo C 中, 只要在 DOS 状态下键入 EXIT 即可。

Quit (退出): 退出 Turbo C, 返回到 DOS 操作系统中, 其快捷键为 “Alt+X”。

(2) **Edit (编辑) 菜单:** 选择编辑菜单后再按回车, 光标出现在编辑窗口, 此时用户可以进行文本编辑。用 “F1” 键可以获得有关编辑方法的帮助信息。

① Turbo C 的双界符包括以下几种符号: 花括号 {和}, 尖括号 <和>, 圆括号 (和), 方括号 [和], 注释符 /*和*/, 双引号", 单引号'。

② Turbo C 在编辑文件时还有一种功能, 就是能够自动缩进, 即光标定位和上一个非空字符对齐。在编辑窗口中, “Ctrl+OL” 为自动缩进开关的控制键。

(3) **Run (运行) 菜单:** 主要用于程序的运行和跟踪调试。该菜单有以下命令。

Run (运行程序): 运行由 Project 菜单中 Project name 命令指定的文件名或当前编辑区的文件。如果上次编译后的源代码未做过修改, 则直接运行到下一个断点 (没有断点则运行到结束)。否则先进行编译、连接后才运行, 其快捷键为 “Ctrl+F9”。

Program reset (程序重启): 中止当前的调试, 释放分给程序的空间, 其快捷键为 “Ctrl+F2”。

Go to cursor (运行到光标处): 调试程序时使用, 选择该项可使程序运行到光标所在行。光标所在行必须为一条可执行语句, 否则提示错误。其快捷键为 “F4”。

Trace into (跟踪进入): 在执行一条调用其他用户定义的子函数时, 若用 Trace into 命令, 则执行长条将跟踪到该子函数内部去执行, 其快捷键为 “F7”。

Step over (单步执行): 执行当前函数的下一条语句, 即使用户函数调用, 执行长条也不会跟踪进函数内部, 其快捷键为 “F8”。

User screen (用户屏幕): 显示程序运行时在屏幕上显示的结果, 其快捷键为 “Alt+F5”。

(4) **Compile (编译) 菜单:** 主要用于程序的编译。该菜单有以下几个命令。

Compile to OBJ (编译生成目标码): 将一个 C 源文件编译生成 OBJ 目标文件, 同时显示生成的文件名, 其快捷键为 “Alt+F9”。

Make EXE file (生成执行文件): 此命令生成一个 .EXE 的文件, 并显示生成的 .EXE 文件名。

Link EXE file (连接生成执行文件): 把当前 .OBJ 文件及库文件连接在一起生成 .EXE 文件, 其快捷键为 “F9”。

Build all（建立所有文件）：重新编译项目里的所有文件，并进行装配生成.EXE 文件。该命令不作过时检查（上面的几条命令要作过时检查，即如果目前项目里源文件的日期和时间与目标文件相同或更早，则拒绝对源文件进行编译）。

Primary C file（主 C 文件）：当在该命令中指定了主文件后，在以后的编译中，如没有项目文件名则编译此命令规定的主 C 文件，如果编译中有错误，则将此文件调入编辑窗口，不管目前窗口中是不是主 C 文件。

Get info：获得有关当前路径、源文件名、源文件字节大小、编译中的错误数目、可用空间等信息。

(5) Project（项目）菜单：主要用于工程管理和多个文件的连编。该菜单包括以下命令。

Project name（项目名）：项目名具有.prj 的扩展名，其中包括将要编译、连接的文件名。例如有一个程序由 file1.c、file2.c、file3.c 组成，要将这 3 个文件编译装配成一个 file.exe 的执行文件，可以先建立一个名为 file.prj 的项目文件，将要连编的文件名写入该文件，每一个文件名各占一行，内容如下：

```
file1.c  
file2.c  
file3.c
```

然后将项目文件名 file.prj 写入 **Project name** 项中，以后进行编译时将自动对项目文件中规定的 3 个源文件分别进行编译，然后连接成 file.exe 文件。

如果其中有些文件已经编译成.obj 文件，而又没有修改过，可直接写上.obj 扩展名。此时将不再编译而只进行连接。例如项目文件的内容为：

```
file1.obj  
file2.c  
file3.c
```

将不对 file1.c 进行编译，而直接连接。当项目文件中的每个文件无扩展名时，默认为.c 源文件。另外，其中的文件也可以是库文件，但必须写上扩展名.lib。

Break make on（中止编译）：由用户选择是否有 **Warning**（警告）、**Errors**（错误）、**Fatal Errors**（致命错误）时或 **Link**（连接）之前退出 **Make** 编译。

Auto dependencies（自动依赖）：当开关置为 on，编译时将检查源文件与对应的.OBJ 文件日期和时间，否则不进行检查。

Clear project（清除项目文件）：清除 **Project** 菜单中 **Project name** 命令所设置的项目文件名。

Remove messages（删除信息）：把错误信息从信息窗口中清除掉。

(6) Options（选项）菜单：主要进行一些系统参数设置，该菜单对初学者来说要谨慎使用。包括 **Compiler**、**Linker**、**Environment**、**Directories**、**Arguments**、**Save options**、**Retrieve options** 等。

其中的 **Directories**（路径）菜单项，用于规定编译、连接所需文件的路径，有下列各项。

Include directories：包含文件的路径，多个子目录用“;”分开。

Library directories: 库文件路径, 多个子目录用“;”分开。

Output directory: 输出文件(.OBJ, .EXE, .MAP 文件)的目录。

Turbo C directory: Turbo C 所在的目录。

Pick file name: 定义加载的 pick 文件名, 如不定义则从 current pick file 中取。

(7) Debug (调试) 菜单: 该菜单主要用于查错。

(8) Break/watch (断点及监视表达式) 菜单: 该菜单有以下内容。

Add watch: 向监视窗口插入一个监视表达式, 快捷键为“Ctrl+F7”。

Delete watch: 从监视窗口中删除当前的监视表达式。

Edit watch: 在监视窗口中编辑一个监视表达式。

Remove all watches: 从监视窗口中删除所有的监视表达式。

Toggle breakpoint: 对光标所在的行设置或清除断点, 快捷键为“Ctrl+F8”。

Clear all breakpoints: 清除所有断点。

View next breakpoint: 将光标移动到下一个断点处。

1.2 第一个 C 语言程序

1.2.1 C 语言程序的编制运行过程

C 语言程序的编制运行过程如图 1-2 所示, 可分为以下几步。

1. 编辑

执行 File 菜单中的 New 命令, 创建一个新的源程序文件。在编辑区域中, 按照程序功能进行 C 语言源程序的编辑。保存扩展名为.C 的源文件。

2. 编译

编辑好源程序后, 执行 Compile 菜单中的 Compile to OBJ 命令, 或者按快捷键“Alt+F9”, 将第一步编辑的源程序编译成扩展名为.OBJ 的目标文件。如果出错, 则回到第一步编辑修改源程序。

3. 连接

编译通过后, 执行 Compile 菜单中的 Link EXE file 命令, 或者按快捷键“F9”, 将第二步编译的目标文件连接成扩展名为.EXE 的可执行程序。如果出错, 则回到第一步编辑修改源程序。

4. 运行

连接成功后, 执行 Run 菜单中的 Run 命令, 或者按快捷键“Ctrl+F9”, 将运行第三步连接的可执行程序。在 Turbo C 中, 当程序运行完后会自动返回 Turbo C 的开发环境, 用户需要执行 Run 菜单中的“User Screen”命令, 或者按快捷键“Alt+F5”到用户屏幕上查看结果。如果结果不正确, 则回到第一步编辑修改源程序。

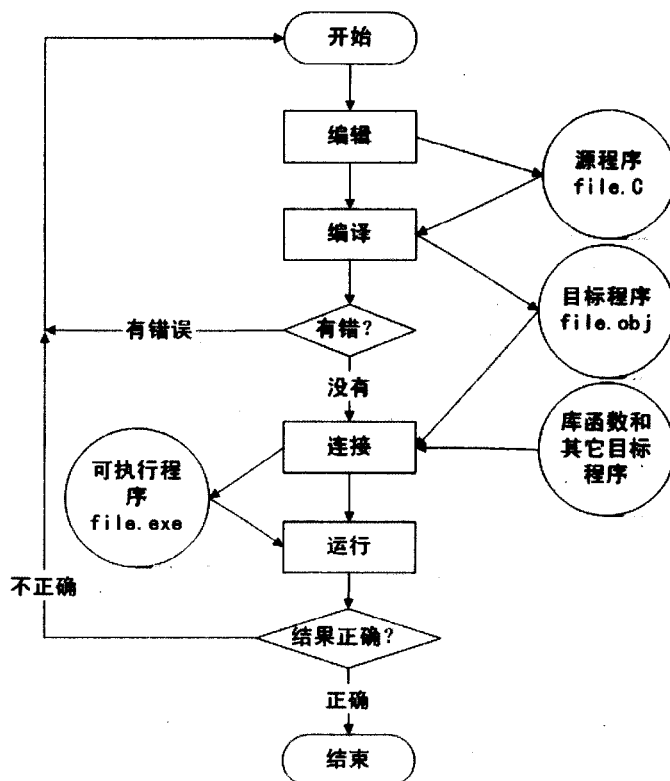


图 1-2 C 语言程序的编制运行过程

1.2.2 C 语言程序的结构

1. C 语言程序的构成

C 语言的源程序存放在扩展名为.C 的文件中, 源程序是由函数构成的, 至少包含一个 main 函数, 或者 main 函数和其他函数。函数是 C 语言程序的基本单位, 可以由用户根据自己的需要进行定义, 也可以是系统提供的库函数。

由用户编写的 main 函数和若干其他函数从结构和功能上都相对独立, 只是由于相互之间的调用关系而联系在一起。C 语言程序的执行从 main 函数开始, 再通过 main 函数对其他函数的调用以及其他函数相互之间的调用实现各函数功能, 然后层层返回各级调用点, 最后返回到调用的起点 main 函数, 从而结束一次程序的运行。

【例 1-1】第一个 C 语言程序。

```

/*这是我的第一个 C 语言程序*/           /*注释*/
#include <stdio.h>                         /*编译预处理*/
void main()                               /*主函数*/
{                                           /*函数开始*/
    printf("This is my first C program!"); /*语句*/
}                                           /*函数结束*/
  
```

其中, 第一行用/*...*/括起来的表示注释, 第二行是编译预处理, 第三行是主函数名, void 表示主函数没有返回值, 第四、六行的一对花括号界定函数体, 表示函数的开始和结束, 第五行是一条 C 语句。