

学习和使用 TURBO C 语言

主审 陈世福
编写 潘沈陈 金默兆 贵君乾

南京大学出版社

-P312
PJ²/1

学习和使用 TURBO C 语言

潘金贵 沈默君 王锡江
陈兆乾 谢俊元 陆庆文 编写

陈世福 主审

南京大学出版社
1993·南京

(苏)新登字第 011 号

内 容 简 介

C 语言和 C++ 语言将逐渐成为计算机程序设计的主导语言。

本书以 Turbo C 的最新版本(2.x 版)为背景,详细介绍了美国 Borland 国际公司在 IBM PC 机上实现的一个速度快、功能强的 C 语言。Turbo C 语言不仅完全支持 Kernigha 和 Ritchie 的 C 语言定义,而且与 ANSI 最新的 C 语言标准兼容并有所扩充。它提供了 450 多个库函数以及与 Turbo Prolog、Turbo Pascal 以及汇编等多种语言的接口。

本书题材丰富、内容深入浅出,详细介绍了 Turbo C 语言的功能,使用方法和多种程序开发技术,对与 Turbo Pascal、Turbo Prolog 和汇编子程序(含直接插入汇编代码)的接口、以及正文窗口、图形处理和低级支撑等高级程序设计技术都作了专门介绍。

本书在《Turbo C 程序设计技术》的基础上作了全面修订,不仅编排上有创意,而且内容上面貌一新。与本书配套的《学习和使用 Turbo C 语言(续编)——Turbo C 实用工具与库函数大全》以及《学习和使用 Turbo C 语言(续二)——Turbo C 工具箱》,是一套学习和使用 Turbo C 语言的较为实用的教材和教学参考书,适合正在使用或将要使用 C 语言进行教学和程序开发的广大教师、研究生、大学生以及科技人员使用和参考。

学习和使用 TURBO C 语言

潘金贵、沈默君 等编写

陈世福 主审

清华大学出版社

(南京大学校内,邮编 210008)

江苏省新华书店发行 江苏射阳印刷厂印刷

开本:787×1029 1/16 印张:20.5 字数 511 千

1993 年 6 月第 1 版

1993 年 6 月第 1 次印刷

印数 1—5000

ISBN 7-305-01847-3/TP·56

定价 11.00 元

责任编辑 顾其兵

前 言

C 语言是一种结构化、模块化、可编译的通用程序设计语言,被广泛地用于系统程序和应用程序的开发。例如,著名的 UNIX 操作系统就是用 C 语言书写的。C 语言有着良好的可移植性,用 C 语言书写的程序在不同的计算机系统之间很容易实现转换。因此,几乎所有的程序设计任务均可使用 C 语言来完成。

Turbo C 是美国 BORLAND 国际公司在 IBM PC 机上实现的一个高效、优化的 C 编译程序。编译速度快,例如 Turbo C 1.5 版的编译程序每分钟就可编译 10000 行源程序,Turbo C 2.0 的编译程序较 1.5 版快 20%—30%。它在编译时利用 RAM 存放中间数据结构,一趟扫描产生内部代码,只需要一次性对盘上的源文件读入和写出目标码(其他 C 语言编译程序则要 4—5 次读盘,每次执行一种功能)。编译产生的目标模块的格式与 PC-DOS 连接程序一致,可与汇编程序连接。支持极小、小、中、紧凑、大和特大六种存储模式。可以使用近、远指针的混合模式。

Turbo C 支持 IEEE 浮点标准,Turbo C 包含有处理 80x87 协处理芯片的浮点例程,80x87 协处理器能大大加快程序的运行速度,但不是必要的,当没有 80x87 协处理器时,可使用系统提供的仿真 80x87 协处理器的实用程序进行高速的浮点运算。

Turbo C 提供了一个完整的交互式集成开发环境,包括:

- 友善的用户接口。通过下拉菜单和多窗口,能够从集成开发环境中产生或运行一个可执行文件。
- 高效能的全屏幕编译程序。当编译过程中出现错误时,编辑程序自动地用彩色光标指出源程序中适当的出错位置。
- 强有力的“MAKE”功能。使得 Turbo C 程序开发的管理十分容易。
- 快速的内部 Turbo 连接程序和内含式上下文敏感的帮助功能等。

Turbo C 2.x 是 Turbo C 风格语言的最新一代。其中,Turbo C 2.0 最为显著的特点是包含了可在源程序级进行符号调试的集成调试程序,可以完成单步执行(STEP OVER 和 TRACE INTO)、设置断点、监视和计算表达式等功能。该调试功能还支持 BORLAND 国际公司新增的独立的调试程序 Turbo DEBUGGER。Turbo C 2.0 还提供了丰富的图形函数库,较之前版本新增了许多函数,包括可安装的驱动程序和字体。所以用 Turbo C 2.0 来编制软件,可以使用户界面设计得相当完美。

除了上述的调试和图形功能以外,Turbo C 2.0 还具有如下一些新的特点:

- 更快的内存分配和串函数
- 更快的浮点计算例行程序
- 更快的内部连接程序
- 可连接生成小模式的.COM 文件
- 新增加了 signal 和 raise 函数
- EMS 扩充存储器可用作编辑缓冲区

- 引入了允许用户在编译时向程序插入机器代码的 `__emit__` 机制
- 支持命令行上的通配符
- 支持 `longdouble` 常数和变量
- 能够自动地进行快缩进/回退及优化填充
- MAKE 实用程序可执行自动依赖关系检查
- 与 Turbo C 1.5, Turbo C 1.0 版本向下兼容

Turbo C 提供了与 Turbo Prolog, Turbo Pascal 和汇编语言等多种语言的接口,能把用不同风格程序设计语言建立的目标模块连成单一的程序。Turbo C 实现了美国国家标准局(ANSI)建议的 C 语言标准,完全支持 Kernighan 和 Ritchie 的 C 定义。提供了 450 个库函数和近十个可独立运行的实用程序(将在《学习和使用 Turbo C 语言(续编)——Turbo C 实用工具与库函数大全》一书中详细介绍),支持 CGA、EGA、VGA、HERCULES、3270PC、AT&T400 线和 IBM 8514 等的高质量图形库。此外还包括了一些混合模式程序设计任选的扩充,进一步挖掘了 PC 机的能力。

Turbo C 提供了由用户选择的常规命令行版本和集成开发环境版本。运行 Turbo C 需要有 448KB 以上内存,最好具有双软盘驱动器,或者一个硬盘驱动器和一个软盘驱动器的 IBM PC 机或其他兼容机以及 DOS 2.0 版本或更高版本的支持。

南京大学计算机科学系已对 Turbo C 2.0 版本进行了成功的汉化,为开发汉化应用软件提供了有力的工具。

本书根据国外最近出版的几种介绍 Turbo C 的书籍和最新配套软件手册编译和改写而成,共分三册。本册共 13 章 2 个附录。各章安排和主要内容如下:

第一章: Turbo C 的安装和启动,描述组成 Turbo C 2.0 系统的六张软盘的文件组织,介绍如何在具体的计算机系统上安装 Turbo C 2.0 的文件和库。

第二章: Turbo C 新的集成开发环境,完整地描述 Turbo C 2.0 的集成开发环境及其菜单系统和菜单命令的使用。

第三章: 集成环境下的 C 程序开发,介绍在集成开发环境下如何建立和修改源文件以及编译和运行 Turbo C 程序的步骤和方法。

第四章: Turbo C 程序设计初步,介绍 Turbo C 语言的七种基本元素以及它们在程序设计中的应用。

第五章: Turbo C 进一步的程序设计技术,介绍了包括数组,指针,指令和语句等另外一些 C 程序设计元素,以及其他较为复杂的程序设计技术。

第六章: Turbo Pascal 与 Turbo C 的异同、转换和连接,使用了一些例子,对 Turbo Pascal 和 Turbo C 进行了比较,描述这两种语言的异同、转换和连接。给出了避免程序设计易犯错误的某些提示,可使已经掌握其中一种语言的读者,很快熟悉另一种语言。

第七章: Turbo C 与 Turbo Prolog 的接口技术,介绍 Turbo C 的模块如何与 Turbo Prolog 程序接口,并提供了一些例子来说明这一技术。

第八章: Turbo C 高级程序设计技术,描述了有关启动不同存储模式代码的内存组织,指针计算,汇编语言接口和浮点用法以及窗口和图形管理设施及其用法。

第九章: 集成环境下的 C 程序调试技术,以典型的例子详细说明如何使用 Turbo C 2.0 新增的集成调试程序。

第十章:改进的 Turbo C 交互式编辑程序,详细介绍了编辑程序的命令和用法。

第十一章:Turbo C 命令行,介绍了 Turbo C 命令行编译程序版本提供的每一个命令行选择项及其用法。

第十二章:Turbo C 的用户定做,说明定做程序(TCINST)的使用,以及如何用定做程序来改变集成开发环境中各缺省参数的设置,如屏幕大小,缺省目录,编辑程序命令和改变屏幕的颜色等等。

第十三章:Turbo C 语言参考,描述了与 Kernighan 和 Ritchie 提出的 C 语言定义不一致的特征和所有方面的列表,详解了 Turbo C 对目前 ANSI C 标准中没有给出的一些扩充的细节。

附录 A:Turbo C 语法的 BNF 描述,用改进的 BNF 形式描述了 Turbo C 2.0 的语法定义。

附录 B:编译出错信息,列出并解释每个出错信息,并且分析了可能引起出错的各种原因。

本书适合广泛的读者对象,例如:C 语言的初学者,富有经验的 C 程序员,使用 Turbo Pascal 和 Turbo Prolog 的程序员都可以从中各取所需地加以利用。我们建议:

C 语言的初学者,可先阅读第三章至第五章,也许能提供一个入门的向导。并可熟悉建立和编译 C 语言程序的过程。如果对集成开发环境还不太清楚,可以再回过头来阅读第二章,当准备运行程序时,第三章指出了具体的步骤。

富有经验的 C 程序员,可先读第八章“Turbo C 高级程序设计技术”。如果企图使用 Turbo C 移植和建立 C 程序,那就要读第三章和第十三章,弄清 Turbo C 与 Kernighan 和 Ritchie 以及 ANSI C 标准的差别。

已经熟悉 Turbo Pascal 的程序员,为了提高学习 Turbo C 语言的速度,最好先阅读第六章,那里提供了一些和 Turbo Pascal 程序等价的 Turbo C 程序的例子,并且详细地描述了这两种语言之间的异同和转换。此外,还需要阅读第二、四、五章。若使用过 Borland 的其他菜单驱动软件,则第二章只需快速浏览一下便可。

使用 Turbo Prolog 的程序员,如急于知道 Turbo C 模块怎样与 Turbo Prolog 接口,那么请先阅读第七章。

可配合本书使用的软件有最新版本的 Turbo C(含纯西文和中西文版本)系统盘以及 Turbo C 工具库盘片共十多枚,可与本书的作者联系获得有关信息。

与本册配套的读物还有《学习和使用 Turbo C 语言(续编)——Turbo C 实用工具与库函数大全》以及《学习和使用 Turbo C 语言(续二)——Turbo C 工具库》,各地新华书店均可购得。

本书在《Turbo C 程序设计技术》一书的基础上,根据 Turbo C 的最新发展重新修订而成。此次重新改编时,由潘金贵副教授和陈世福教授主持编写,王锡江和陈兆乾副教授分别完成了第五章和第十三章的修订,其余主要由沈默君、潘金贵、谢俊元、陆庆文等同志完成。潘金贵负责对全书进行了仔细的统编和修改。限于水平和时间仓促,书中若有错误或不妥之处,恳请读者批评指正。

目 录

前 言

第一章 Turbo C 的安装和启动	(1)
§ 1.1 Turbo C 语言简介	(1)
1.1.1 C 语言的起源	(1)
1.1.2 Turbo C 2.0 的新特点	(1)
1.1.3 中西文 Turbo C	(2)
§ 1.2 Turbo C 程序结构	(3)
§ 1.3 Turbo C 系统的组成及文件配置	(4)
§ 1.4 安装和启动 Turbo C 2.0 系统	(5)
第二章 Turbo C 新的集成开发环境	(11)
§ 2.1 Turbo C 集成开发环境及使用	(11)
2.1.1 TC 的命令行开关	(11)
2.1.2 基本导航操作	(12)
2.1.3 TC 的热键	(13)
2.1.4 菜单结构及命名约定	(15)
2.1.5 主菜单	(15)
2.1.6 快速参考行	(16)
2.1.7 编辑窗口	(16)
2.1.8 编辑命令速成指南	(17)
2.1.9 在编辑窗口中操作源文件	(17)
2.1.10 信息窗口	(18)
2.1.11 观察窗口	(19)
2.1.12 集成调试程序	(19)
§ 2.2 菜单命令及其使用	(21)
2.2.1 File 菜单	(21)
2.2.2 Edit 命令	(23)
2.2.3 Run 菜单	(23)
2.2.4 Compile 菜单	(26)
2.2.5 Project 菜单	(28)
2.2.6 Options 菜单	(29)
2.2.7 Debug 菜单	(41)
第三章 集成环境下的 C 程序开发	(47)
§ 3.1 在集成开发环境中建立 Turbo C 可执行程序	(47)
§ 3.2 建立和运行单个源文件的可执行程序	(47)
3.2.1 建立和运行单源文件程序的步骤	(47)
3.2.2 浏览盘上产生的文件	(50)

3.2.3 将输出送往打印机	(50)
3.2.4 编辑一个 C 程序	(50)
3.2.5 存盘	(51)
3.2.6 排错和调试	(51)
§ 3.3 建立和运行含有多个源文件的 C 程序	(53)
3.3.1 建立和运行多源文件程序的步骤	(53)
3.3.2 多源文件编译时的错误跟踪	(54)
3.3.3 Project Make 的功用	(56)
§ 3.4 Make 的其他一些特性	(57)
3.4.1 外部目标文件和库文件	(57)
3.4.2 标准文件的取代	(58)
§ 3.5 MAKE 实用程序	(58)
第四章 Turbo C 程序设计初步	(59)
§ 4.1 程序设计的基本元素	(59)
§ 4.2 输出	(60)
4.2.1 printf 函数	(60)
4.2.2 其他输出函数:puts 和 putchar	(61)
§ 4.3 数据类型	(61)
4.3.1 浮点型	(61)
4.3.2 三种整型	(62)
4.3.3 无符号类型	(62)
4.3.4 定义字符串	(62)
4.3.5 标识符	(63)
§ 4.4 运算	(64)
4.4.1 赋值运算符	(64)
4.4.2 单目运算符和双目运算符	(64)
4.4.3 增 1(++)和减 1(--)运算符	(64)
4.4.4 按位运算符	(65)
4.4.5 组合运算符	(66)
4.4.6 地址运算符	(66)
§ 4.5 输入	(67)
4.5.1 scanf 函数	(67)
4.5.2 用 gets 和 getch 输入	(68)
§ 4.6 条件语句	(68)
4.6.1 关系运算符	(68)
4.6.2 逻辑运算符	(69)
4.6.3 关于表达式的进一步说明	(69)
4.6.4 if...else 语句	(70)
§ 4.7 循环	(71)
4.7.1 while 循环	(71)
4.7.2 for 循环	(72)
4.7.3 do...while 循环	(73)

§ 4.8 函数	(74)
4.8.1 程序分析	(75)
4.8.2 全量说明	(76)
4.8.3 函数说明	(76)
4.8.4 函数定义	(77)
§ 4.9 注释	(78)
第五章 Turbo C 进一步的程序设计技术	(79)
§ 5.1 数据结构	(79)
5.1.1 指针	(79)
5.1.2 数组	(83)
5.1.3 结构	(87)
§ 5.2 switch 语句	(88)
§ 5.3 控制流命令	(91)
5.3.1 return 语句	(91)
5.3.2 break 语句	(91)
5.3.3 continue 语句	(92)
5.3.4 goto 语句	(93)
5.3.5 条件表达式(?)	(93)
§ 5.4 流和 I/O 流	(93)
5.4.1 什么是流	(93)
5.4.2 文字流和二进制流	(94)
5.4.3 缓存流	(94)
5.4.4 预定义流	(94)
§ 5.5 C 程序设计风格	(95)
5.5.1 使用函数原型和全函数定义	(95)
5.5.2 使用 enum 定义	(96)
5.5.3 使用 typedef	(96)
5.5.4 说明 void 函数	(97)
5.5.5 扩充的使用	(97)
§ 5.6 C 程序设计中的常见问题	(98)
5.6.1 使用 C 字符串的路径名	(98)
5.6.2 指针的使用和误用	(98)
5.6.3 赋值号(=)和等号(==)的混淆	(100)
5.6.4 在 switch 语句中漏写 break 语句	(100)
5.6.5 数组下标	(100)
5.6.6 忘记传送地址	(101)
第六章 Turbo Pascal 与 Turbo C 的异同、转换和连接	(103)
§ 6.1 Turbo Pascal 与 Turbo C 的比较	(103)
6.1.1 程序结构	(103)
6.1.2 程序设计成分	(104)
6.1.3 数据结构	(116)
6.1.4 编程问题	(123)

6.1.5 Pascal 程序人员使用 C 时的常见错误	(127)
§ 6.2 Turbo Pascal 程序到 Turbo C 的转换	(129)
6.2.1 把 Turbo Pascal 循环转换为 C 循环	(129)
6.2.2 case 和 if 语句	(130)
6.2.3 结构和记录	(131)
6.2.4 一个手工转换的例子	(131)
6.2.5 实现自动转换的一个试验原型	(133)
§ 6.3 Turbo C 与 Turbo Pascal 的连接	(142)
第七章 Turbo C 和 Turbo Prolog 的接口技术	(146)
§ 7.1 Turbo C 与 Turbo Prolog 的连接步骤	(146)
7.1.1 对程序模块进行编译	(146)
7.1.2 对程序模块进行连接	(146)
7.1.3 其他注意事项	(147)
§ 7.2 Turbo C 与 Turbo Prolog 的连接示例	(147)
7.2.1 示例之一: 两个整数相加	(147)
7.2.2 示例之二: 使用数学库	(149)
7.2.3 示例之三: 使用流模式和存储分配	(152)
7.2.4 示例之四: 画三维条形图	(157)
第八章 Turbo C 高级程序设计技术	(164)
§ 8.1 存储模式	(164)
8.1.1 8086 寄存器	(164)
8.1.2 内存分段及地址计算	(165)
8.1.3 近指针、远指针和特大指针	(166)
8.1.4 Turbo C 的六种存储模式	(167)
8.1.5 混合模式程序设计: 地址修饰符	(168)
§ 8.2 多语言混合程序设计: 和汇编语言接口	(173)
8.2.1 C 和 Pascal 的参数传递顺序	(173)
8.2.2 汇编语言接口	(176)
8.2.3 从汇编语言程序调用 Turbo C	(178)
8.2.4 定义汇编语言子程序	(179)
8.2.5 寄存器使用约定	(182)
8.2.6 从汇编子程序调用 C 函数	(182)
§ 8.3 程序设计的低级支撑	(183)
8.3.1 伪变量	(183)
8.3.2 直接插入汇编代码	(185)
8.3.3 中断函数	(193)
§ 8.4 浮点库的使用	(195)
8.4.1 仿真 8087/80287 芯片	(196)
8.4.2 使用 8087/80287 数学协处理器	(197)
8.4.3 不使用浮点数的场合	(197)
8.4.4 87 环境变量	(198)
8.4.5 寄存器和 8087/80287	(198)

8.4.6 浮点出错处理	(199)
§ 8.5 警告和提示	(199)
8.5.1 Turbo C RAM 的使用	(199)
8.5.2 要慎用 Pascal 调用约定	(199)
8.5.3 在 DOS 3.2 和有浮点协处理器下使用 Turbo C	(199)
§ 8.6 Turbo C 的字符屏幕管理	(200)
8.6.1 基本概念	(201)
8.6.2 显示方式控制	(202)
8.6.3 字符输出	(203)
8.6.4 程序例	(203)
§ 8.7 Turbo C 的图形功能	(204)
8.7.1 基本概念	(204)
8.7.2 图形系统控制	(206)
8.7.3 颜色控制	(207)
8.7.4 绘图和着色	(208)
8.7.5 图形屏幕管理和视区设置	(212)
8.7.6 图形模式下的正文输出	(213)
8.7.7 图形模式中的错误处理	(214)
8.7.8 状态查询	(215)
第九章 集成环境下的 C 程序调试技术	(217)
§ 9.1 集成调试程序的工作方式	(217)
9.1.1 调试简单程序	(218)
9.1.2 设置和使用断点	(221)
9.1.3 使用 Ctrl-Break	(222)
9.1.4 单步执行函数调用	(222)
9.1.5 计算表达式	(222)
9.1.6 nextword 和 wordlen 函数	(223)
9.1.7 静心思考错误原因	(223)
9.1.8 回顾已完成的工作	(224)
9.1.9 Evaluate 窗口的默认表达式	(224)
9.1.10 改变所求表达式的值	(224)
9.1.11 受限变量名	(225)
9.1.12 格式区分符	(225)
§ 9.2 找出 wordlen 中的错误	(228)
9.2.1 改正错误	(229)
9.2.2 已完成的工作	(229)
9.2.3 断点的进一步讨论	(229)
§ 9.3 继续调试所给程序	(230)
9.3.1 编辑和删除观察表达式	(231)
9.3.2 窗口的扩大和转换	(231)
9.3.3 卷滚观察表达式	(232)
§ 9.4 调试打印循环	(232)

§ 9.5 调试大型程序	(232)
9.5.1 寻找函数的定义	(233)
9.5.2 调用栈	(233)
9.5.3 返回执行位置	(233)
§ 9.6 关于多源文件	(233)
§ 9.7 调试程序命令和热键综述	(234)
§ 9.8 软件测试指南	(235)
9.8.1 开发标准方法	(235)
9.8.2 彻底地测试修改	(236)
9.8.3 预防性设计	(236)
9.8.4 自底向上调试	(236)
9.8.5 寻找同类错误	(237)
§ 9.9 调试直接插入的汇编代码	(237)
第十章 改进的 Turbo C 交互式编辑程序	(238)
§ 10.1 快速进入和退出编辑程序	(238)
§ 10.2 编辑窗口状态行	(238)
§ 10.3 编辑命令	(239)
10.3.1 基本光标移动命令	(240)
10.3.2 快速光标移动命令	(241)
10.3.3 插入和删除命令	(241)
10.3.4 块命令	(242)
10.3.5 其他编辑命令	(243)
§ 10.4 Turbo C 编辑程序与 Wordstar 之比较	(245)
第十一章 Turbo C 命令行	(247)
§ 11.1 命令行选择项与菜单选择项的对应关系	(247)
§ 11.2 编译选择项	(249)
11.2.1 存储模式选择项	(249)
11.2.2 定义	(249)
11.2.3 代码生成选择项	(250)
11.2.4 优化选择项	(251)
11.2.5 源选择项	(252)
11.2.6 出错报告选择项	(252)
11.2.7 段命名选择项	(253)
11.2.8 编译控制选择项	(254)
§ 11.3 连接选择项	(254)
§ 11.4 环境选择项	(254)
11.4.1 隐式库文件和显式库文件	(255)
11.4.2 库文件的搜索算法	(255)
§ 11.5 从命令行直接编译和连接 Turbo C 程序	(255)
11.5.1 命令行一般格式	(256)
11.5.2 可执行文件的产生	(256)

11.5.3 命令行的例子	(256)
§ 11.6 TURBOC.CFG 文件	(258)
§ 11.7 在 DOS 下直接运行 Turbo C 程序	(258)
第十二章 Turbo C 的用户定做	(259)
§ 12.1 定做程序 TCINST 的功用	(259)
§ 12.2 运行 TCINST	(259)
§ 12.3 TCINST 设置菜单	(260)
12.3.1 编译菜单	(261)
12.3.2 工程菜单	(261)
12.3.3 选择项菜单	(261)
12.3.4 Debug 菜单	(265)
12.3.5 编辑命令选择项	(265)
12.3.6 显示模式菜单	(268)
12.3.7 颜色设置菜单	(268)
12.3.8 改变窗口大小选择项	(269)
12.3.9 从 TCINST 程序退出	(269)
第十三章 Turbo C 语言参考	(270)
§ 13.1 注解	(270)
§ 13.2 标识符	(271)
§ 13.3 关键字	(271)
§ 13.4 常量	(272)
13.4.1 整型常量	(272)
13.4.2 字符常量	(272)
13.4.3 浮点常量	(273)
§ 13.5 字符串	(273)
§ 13.6 硬件特性	(274)
§ 13.7 类型转换	(275)
13.7.1 字符、整数与枚举	(275)
13.7.2 指针	(275)
13.7.3 算术转换	(275)
§ 13.8 运算符	(276)
§ 13.9 类型与类型修饰符	(276)
13.9.1 枚举类型	(276)
13.9.2 void 类型	(277)
13.9.3 带正负号修饰符 signed	(277)
13.9.4 常量修饰符 const	(277)
13.9.5 易变修饰符 volatile	(278)
13.9.6 修饰符 cdecl 和 pascal	(278)
13.9.7 修饰符 near, far 和 huge	(279)
§ 13.10 结构与联合	(279)
13.10.1 字边界	(279)

13.10.2 位域	(280)
§ 13.11 语句	(280)
§ 13.12 外部函数定义	(280)
13.12.1 函数类型修饰符	(280)
13.12.2 函数原型	(282)
§ 13.13 作用域规则	(284)
§ 13.14 编译程序控制行	(285)
13.14.1 词法单位替换	(285)
13.14.2 文件嵌入	(286)
13.14.3 条件编译	(286)
13.14.4 行控制	(286)
13.14.5 出错指令	(287)
13.14.6 pragma 编译指令	(287)
13.14.7 空编译指令	(288)
13.14.8 预定义宏名	(288)
13.14.9 Turbo C 预定义宏	(288)
§ 13.15 过时成份	(289)
附录 A Turbo C 语法的 BNF 描述	(290)
A.1 词法	(290)
A.1.1 词法单位	(290)
A.1.2 关键字	(290)
A.1.3 标识符	(290)
A.1.4 常量	(291)
A.1.5 字符串常量	(292)
A.1.6 运算符	(292)
A.1.7 标点字符	(292)
A.2 语法	(292)
A.2.1 表达式	(292)
A.2.2 说明	(294)
A.2.3 语句	(295)
A.2.4 外部定义	(295)
A.3 预处理指令	(295)
附录 B 编译出错信息	(297)
B.1 严重错误	(297)
B.2 一般错误	(297)
B.3 警告	(306)
参考资料	(309)

第一章 Turbo C 的安装和启动

本章概要地介绍 Turbo C 语言的起源、特点、现有版本及中西文 Turbo C、Turbo C 的程序结构和组成 Turbo C 2.0 系统的文件配置,最后介绍 Turbo C 系统的安装和启动。

§ 1.1 Turbo C 语言简介

1.1.1 C 语言的起源

C 语言以其良好的可移植性和广泛的实用性而著称,深受国内外广大用户的欢迎,成为当今颇具影响的程序设计语言之一。

C 语言的出现是和 UNIX 操作系统紧密联系在一起的。它的许多重要思想来自于早在 1968 年发表的 CPL (Combined Programming Language) 语言以及 Martin Richards 在 1969 年研制的 BCPL 语言和由 K. Thompson 在 1970 年研制成的以 BCPL 语言为基础的 B 语言。K. Thompson 用 B 语言为 PDP-7 计算机书写了第一个 UNIX 操作系统。1972 年, D. M. Ritchie 在 B 的基础上研制出 C 语言,并用 C 语言为 PDP-11 计算机写成了第一个 UNIX 操作系统。1977 年出现了独立于机器的 C 语言编译文本《可移植 C 语言编译程序》,这就大大简化了把 C 语言编译程序移植到新环境所需做的工作,从而推动了 UNIX 操作系统迅速地在各种机器上的实现。

多年来, B. Kernigha 和 D. Ritchie 在《The C Programming Language》一书(以下简称《K&R》)中定义的 C 语言,一直被作为 C 的非官方标准,然而《K&R》中并未定义一个完整的标准 C 语言。随着微型计算机的日益普及,出现了各种 C 语言编译程序和工具,虽然在源程序级上这些工具能够保持兼容,但由于没有统一的标准,这些工具间也存在不一致之处,为了改变这种情况,美国国家标准局(ANSI)于 1983 年成立了 X3J11 专门委员会来制定 C 的 ANSI 标准,并于 1986 年提出了一个 C 语言标准草案。

1.1.2 Turbo C 2.0 的新特点

Turbo C 语言,无论是程序开发环境,还是语言本身都具有独特的风格,它完全支持《K&R》中定义的 C 语言,而且实现了大部分 ANSI 规定,并作了若干扩充,它通过增加原先标准提高灵活性,扩充了 C 语言,使得 Turbo C 更加实用。

自 1987 年 Turbo C (1.0 版本)问世以来,已经有了重要的发展,相继推出了 1.5 版本, 2.0 版本。以及支持面向对象的程序设计的 Turbo C++ 1.0、Borland C++ 2.0 版本。它们是在 Turbo C 的基础上扩充实现的,因此,学习 Turbo C 也是学习 C++ 的基础。2.0 版本是 Turbo C 的最新版本,它在 1.5 版本的基础上又作了重大改进,并增加许多新的功能,具有下列新特点:

- (1) 具有一个集编辑、编译、调试、运行一体化的集成开发环境;
- (2) 集成调试程序可对程序进行单步执行,单步跟踪,设置断点,对表达式监视和求值

等;

- (3) 支持 Borland 新的独立调试程序 Turbo Debugger;
- (4) 编译和连接的速度又有提高, 比以前快 20%—30%;
- (5) 有了更快的内存分配函数和串函数;
- (6) EMS(扩展存储器系统)可用作编辑缓冲区;
- (7) 浮点仿真、浮点运算速度更快;
- (8) 新增加了 signal 和 raise 函数;
- (9) 新增加的 __emit__ 函数允许用户在编译时插入机器代码到应用程序中去;
- (10) BGI 图形库中增加了许多新的函数, 如可加载的驱动程序和字体文件;
- (11) 支持命令行上的通配符 *, ? 等;
- (12) 连接程序可对极小模式程序建立 .COM 文件;
- (13) 支持 long double 常数和变量;
- (14) 编辑程序提供块自动缩进/回退及最佳填充等新的编辑功能;
- (15) MAKE 实用程序提供程序依赖关系的自动检查;
- (16) 新增加了三个实用程序:
 - ①THELP: 用在 DOS 下得到与 Turbo C2.0 内容相关的帮助系统的内存驻留程序;
 - ②CINSTXFR.EXE: 用于把 Turbo C1.5 版本集成开发环境的配置文件转换为 2.0 版本对应的文件;
 - ③OBJXREF.EXE: 按指定目录顺序搜索所有目标文件和库文件。

1.1.3 中西文 Turbo C

我们在使用 Turbo C 的基础上, 分别对 1.0 版, 1.5 版和 2.0 版进行了汉化。

为了适应不同机型的特点, 针对具有不同显示特性的机种, 对 Turbo C 2.0 版设计和实现了三个中西文版本, 即 PC 版(适合 IBM PC XT/AT 机, 操作系统为 CCDOS), 长城版(适合长城 CEGA 显示器的 0520CH、长城 286/386 机), 386 版(适合 PC/286 或 386 机, EGA 显示器, 操作系统 UCDOS)。几种中西文版本的共同特征是:

- (1) 保持西文版本的全部特点并兼容;
- (2) 在集成开发环境内能编辑、调试带有汉字输入/输出的 C 程序, 包括:
 - 汉字可以作为字符串进行操作;
 - 内部编辑程序可以对汉字进行输入、显示及编辑, 所有原来的编辑功能, 都能用来对汉字操作;
 - 部分提示信息 and 帮助信息的汉化;
 - 汉字的输入/输出操作与 CCDOS 或 UCDOS 一致;
- (3) 中西文 Turbo C 能对带汉字的 C 源程序进行正确的编译, 并生成正确的执行代码;
- (4) 长城版和 386 版所有窗口彩色特性与西文版本保持一致;
- (5) 中西文 Turbo C 系统及其编译后产生的可执行程序能在 CCDOS 或 UCDOS 下运行。

汉化工作主要是针对 Turbo C 的集成开发环境编译程序版本进行的, 从而使得 Turbo C 倍受国内广大 C 程序人员的青睐。

此外, 我们已用汉化的中西文 Turbo C 完成了七·五国家重点科技攻关项目“NEW 知

识工程语言”以及“智能刺绣编程系统”。

§ 1.2 Turbo C 程序结构

C 语言不像其他语言，只有很少的关键字，一般为 32 个(其中 27 个来自《K&R》的定义，另 5 个由 ANSI 标准增补)，Turbo C 增添了 11 个，共 43 个关键字(参见表 1.1)，这些关键字，加上语法规则，就构成了 C 编译程序。C 语言是大小写敏感的，即字母的大小写是有区别的，所有关键字都采用小写。关键字不能用作其他用途，例如变量名、函数名不能与关键字相同。

表 1.1 Turbo C 的关键字

由 ANSI 标准推荐的 32 个关键字			
auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while
Turbo C 扩展的关键字			
asm	_cs	_ds	_es
_ss	cdecl	far	huge
interrupt	near	pascal	

一个 C 程序由一个或多个函数组成，其中至少有一个函数是 main()，它是执行程序时最先被调用的函数。C 程序的一般形式如图 1.1 所示。

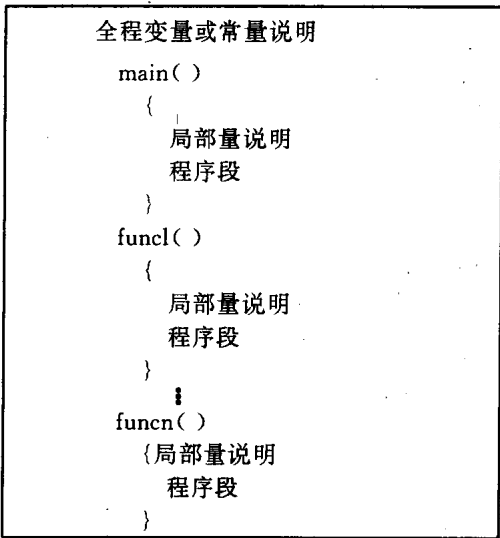


图 1.1 C 程序结构