



经典范例 50 讲系列

北京希望电子出版社 总策划
任志宏 程 超 编 著

C语言 经典范例 50 讲

中国物资出版社



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn



经典范例 50 讲系列

刘著

北京希望电子出版社

任志宏

刘著

刘著

刘著

刘著

刘著

C语言 经典范例 50 讲

中国物资出版社



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

图书在版编目 (CIP) 数据

C 语言经典范例 50 讲 / 任志宏, 程超编著. —北京: 中国物资出版社, 2004 年 12 月
(经典范例 50 讲系列)

国际标准书号 ISBN 7-5047-2162-X

I .C... II. ①任...②程... III. ①C 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字 (2004) 第 044161 号

责任编辑 黑俊贵 周凤明

责任印制 方鹏远

责任校对 王 莉

中国物资出版社出版发行

网址: <http://www.clph.cn>

社址: 北京市西城区月坛北街 25 号

电话: (010) 68589540 邮编: 100834

全国新华书店经销

北京双青印刷厂

开本: 787×1092mm 1/16 印张: 33.75 字数: 787 千字

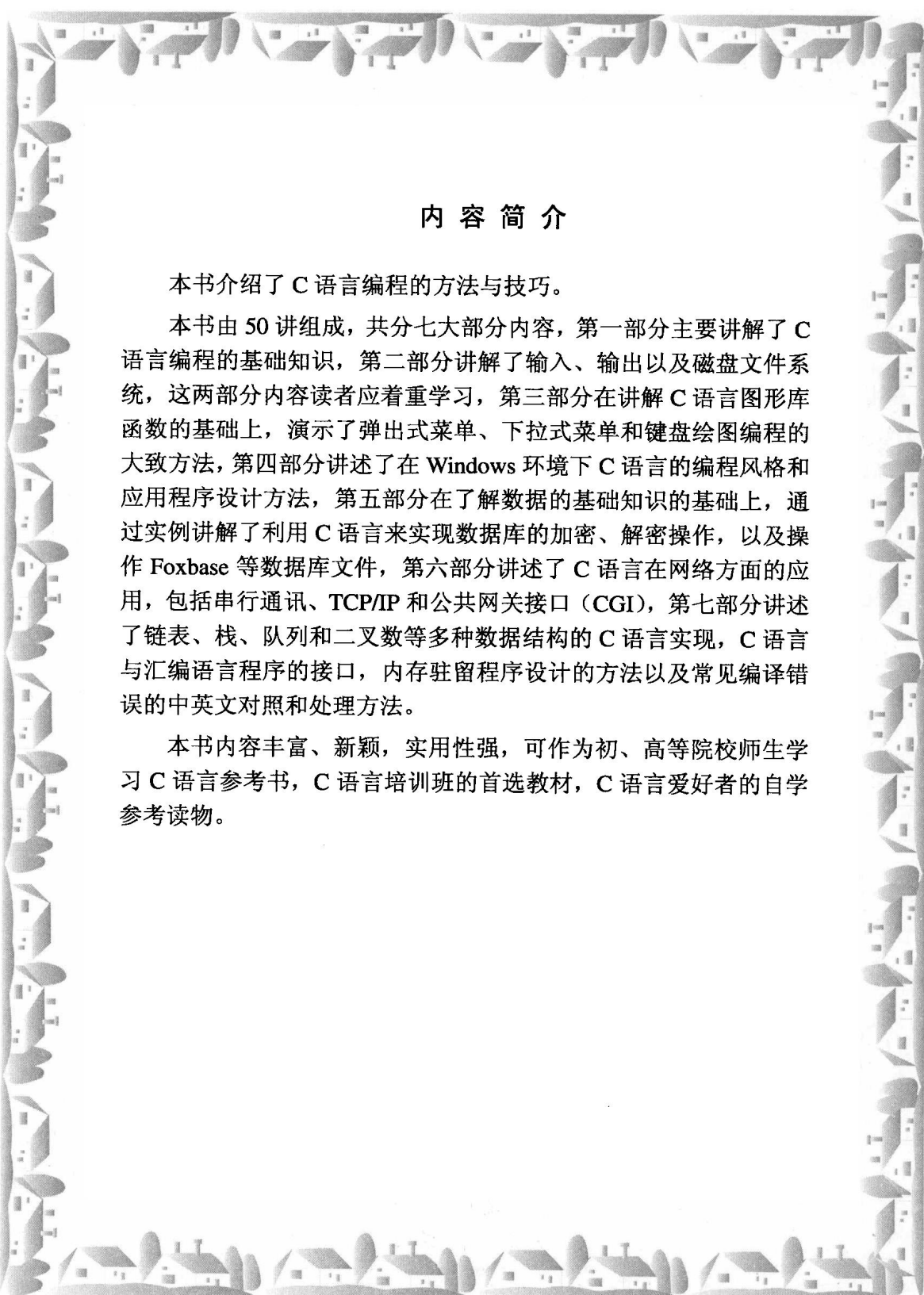
2004 年 12 月第 1 版 2004 年 12 月第 1 次印刷

书号: ISBN7-5047-2162-X/TP·0057

印数: 0001—5000 册

定价: 40.00 元

(图书出现印装质量问题, 本社负责调换)



内 容 简 介

本书介绍了 C 语言编程的方法与技巧。

本书由 50 讲组成，共分七大部分内容，第一部分主要讲解了 C 语言编程的基础知识，第二部分讲解了输入、输出以及磁盘文件系统，这两部分内容读者应着重学习，第三部分在讲解 C 语言图形库函数的基础上，演示了弹出式菜单、下拉式菜单和键盘绘图编程的大致方法，第四部分讲述了在 Windows 环境下 C 语言的编程风格和应用程序设计方法，第五部分在了解数据的基础知识的基础上，通过实例讲解了利用 C 语言来实现数据库的加密、解密操作，以及操作 Foxbase 等数据库文件，第六部分讲述了 C 语言在网络方面的应用，包括串行通讯、TCP/IP 和公共网关接口 (CGI)，第七部分讲述了链表、栈、队列和二叉数等多种数据结构的 C 语言实现，C 语言与汇编语言程序的接口，内存驻留程序设计的方法以及常见编译错误的中英文对照和处理方法。

本书内容丰富、新颖，实用性强，可作为初、高等院校师生学习 C 语言参考书，C 语言培训班的首选教材，C 语言爱好者的自学参考读物。

前 言

C 语言是国际上广泛流行的、很有发展前途的计算机高级语言。它适合于作为系统描述语言，既用来编写系统软件，也可用来编写应用软件。

早期的操作系统等系统软件主要是用汇编语言编写的（包括 UNIX 操作系统在内）。由于汇编语言依赖于计算机硬件，程序的可读性和可移植性都比较差。为了提高可读性和可移植性，最好改用高级语言，但一般的高级语言难以实现汇编语言的某些功能（汇编语言可以直接对硬件进行操作），例如：对内存地址的操作、位操作等。人们设想能否找到一种既具有一般高级语言特性，又具有低级语言特性的语言，集它们的优点于一身。于是，C 语言就在这种情况下应运而生了。

C 语言是在 B 语言的基础上发展起来的，它的根源可以追溯到 ALGOL 60。1960 年出现的 ALGOL 60 是一种面向问题的高级语言，它离硬件比较远，不宜用来编写系统程序。1963 年英国的剑桥大学推出了 CPL (Combined Programming Language) 语言。CPL 语言在 ALGOL 60 的基础上接近了硬件一些，但规模比较大，难以实现。1967 年英国剑桥大学的 Martin Richards 对 CPL 语言作了简化，推出了 BCPL (Basic Combined Programming Language) 语言。1970 年美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础，又作了进一步简化，设计出了很简单的而且很接近硬件的 B 语言（取 BCPL 的第一个字母），并用 B 语言写第一个 UNIX 操作系统，在 PDP-7 上实现。1971 年在 PDP-11/20 上实现了 B 语言，并写了 UNIX 操作系统。但 B 语言过于简单，功能有限。1972 年至 1973 年间，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）。C 语言既保持了 BCPL 和 B 语言的优点（精练、接近硬件），又克服了它们的缺点（过于简单、数据无类型等）。最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工作语言而设计的。1973 年，K.Thompson 和 D.M.Ritchie 两人合作把 UNIX 90% 以上代码用 C 改写（UNIX 第 5 版。原来的 UNIX 操作系统是 1969 年由美国的贝尔实验室的 K.Thompson 和 D.M.Ritchie 开发成功的，是用汇编语言写的）。

后来，C 语言多次作了改进，但主要还是在贝尔实验室内部使用。直到 1975 年 UNIX 第 6 版公布后，C 语言的突出优点才引起人们普遍注意。1977 年出现了不依赖于具体机器的 C 语言编译文本《可移植 C 语言编译程序》，使 C 移植到其他机器时所做的工作大大简化了，这也推动了 UNIX 操作系统迅速地在各种机器上实现。例如，VAX，AT&T 等计算机系统都相继开发了 UNIX。随着 UNIX 的日益广泛使用，C 语言也迅速得到推广。C 语言和 UNIX 可以说是一对孪生兄弟，在发展过程中相辅相成。1978 年以后，C 语言已先后移植到大、中、小、微型机上，已独立于 UNIX 和 PDP 了。现在 C 语言已风靡全世界，成为世界上应用最广泛的几种计算机语言之一。

以 1978 年发表的 UNIX 第 7 版中的 C 编译程序为基础，Brian W.Kernighan 和 Dennis M.Ritchie (合称 K&R) 合著了影响深远的名著《The C Programming Language》，这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础，它被称为标准 C。1983 年，美国国家

标准化协会（ANSI）根据 C 语言问世以来各种版本对 C 的发展和扩充，制定了新的标准，称为 ANSI C。ANSI C 比原来的标准 C 有了很大的发展。K&R 在 1988 年修改了经典著作《The C Programming Language》，按照 ANSI C 的标准重新写了该书。1987 年，ANSI C 又公布了新标准--87 ANSI C。目前流行的 C 编译系统都是以它为基础的。

本书的体系结构是针对初中层次读者特点精心安排的。本书在讲解 C 语言的基础上，主要着重于对实例的分析和讲解，在每一讲都包含 1~2 个范例，范例主要是针对每一讲的要点和难点而编写的，其中一些范例实用性很强。读者在阅读完每讲的基础知识后，可以通过范例进一步掌握所学内容，加深理解。为了读者学习方便，本书附送光盘上包含所有的范例源代码，源代码全部都在 Turbo C 调试通过。

本书共分七大部分，第一部分主要讲解了 C 语言编程的基础知识，第二部分讲解了输入、输出以及磁盘文件系统，这两部分内容读者应着重学习，第三部分在讲解 C 语言图形库函数的基础上，演示了弹出式菜单、下拉式菜单和键盘绘图编程的大致方法，第四部分讲述了在 Windows 环境下 C 语言的编程风格和应用程序设计方法，第五部分在了解数据的基础知识的基础上，通过实例讲解了利用 C 语言来实现数据库的加密、解密操作，以及操作 Foxbase 等数据库文件，第六部分讲述了 C 语言在网络方面的应用，包括串行通讯、TCP/IP 和公共网关接口（CGI），第七部分讲述了链表、栈、队列和二叉数等多种数据结构的 C 语言实现，C 语言与汇编语言程序的接口，内存驻留程序设计的方法，以及常见编译错误的中英文对照和处理方法。

本书不要求读者具有专门的计算机专业知识的基础，完全能够学习并掌握 C 语言的基本内容。对于已经初步学习过 C 语言的读者来说，可以直接从第三部分开始阅读。对于 C 语言的高级开发，如图形、图像、网络、数据库等，会涉及到汇编语言和 DOS 操作系统的中断和功能调用，请读者另行参考其他相关书籍。

本书由任志宏、程超组织编写，其他参与本书部分编写、录排、审核等工作的人员还有：李韵、翁颖钧、万英、罗曼莉、任宇斌、崔卓、李春明、周莹、赖擎、于自跃、韩存兵、张巧莉、张莉、田野、刘晨宏、龚建、龚志翔、季宁、胡方霞、李正希、陈发吉、周树成、陈博、张德强、黄司渭、温尚清、夏兵等。龚超同志对全稿进行复审。在此还要感谢北京希望电子出版社的周凤明编辑，周老师的辛勤劳动使本书质量更上一层楼！

由于时间仓促，且经验和水平有限，文中难免有不妥之处，我们殷切地期望读者能给我们提出中肯的意见，以便于提高水平，把更好的图书呈现给大家！

作者

目 录

第0讲 预备知识.....1	8.2 范例.....89
0.1 C语言的特点.....1	8.3 本讲总结.....92
0.2 C语言的开发环境.....2	第9讲 函数的作用域规则.....93
第1讲 变量、常量、运算符和表达式.....13	9.1 背景知识.....93
1.1 背景知识.....13	9.2 范例.....103
1.2 范例.....28	9.3 本讲总结.....105
1.3 本讲总结.....30	第10讲 一维数组.....106
第2讲 结构化编程基础.....32	10.1 背景知识.....106
2.1 背景知识.....32	10.2 范例.....110
2.2 范例.....41	10.3 本讲总结.....112
2.3 本讲总结.....43	第11讲 二维数组和多维数组.....113
第3讲 判断选取控制和分支控制结构.....44	11.1 背景知识.....113
3.1 背景知识.....44	11.2 范例.....116
3.2 范例.....47	11.3 本讲总结.....120
3.3 本讲总结.....51	第12讲 字符数组.....121
第4讲 简单循环控制结构.....52	12.1 背景知识.....121
4.1 背景知识.....52	12.2 范例.....123
4.2 范例.....57	12.3 本讲总结.....125
4.3 本讲总结.....60	第13讲 数组应用实例.....126
第5讲 复杂循环控制结构.....61	13.1 背景知识.....126
5.1 背景知识.....61	13.2 范例.....126
5.2 范例.....62	13.3 本讲总结.....129
5.3 本讲总结.....64	第14讲 字符串的基本知识.....131
第6讲 子过程与函数过程.....65	14.1 背景知识.....131
6.1 背景知识.....65	14.2 范例.....133
6.2 范例.....77	14.3 本讲总结.....136
6.3 本讲总结.....81	第15讲 字符串应用实例.....137
第7讲 函数的嵌套调用.....82	15.1 背景知识.....137
7.1 背景知识.....82	15.2 范例.....137
7.2 范例.....83	15.3 本讲总结.....139
7.3 本讲总结.....86	第16讲 指针.....140
第8讲 函数的递归调用.....88	16.1 背景知识.....140
8.1 背景知识.....88	16.2 范例.....147

16.3 本讲总结	148	25.2 范例	235
第 17 讲 指针和数组	150	25.3 本讲总结	241
17.1 背景知识	150	第 26 讲 控制台 I/O	242
17.2 范例	158	26.1 背景知识	242
17.3 本讲总结	160	26.2 范例	244
第 18 讲 指针和字符串	161	26.3 本讲总结	246
18.1 背景知识	161	第 27 讲 缓冲文件系统	247
18.2 范例	165	27.1 背景知识	247
18.3 本讲总结	169	27.2 范例	261
第 19 讲 函数和指针	170	27.3 本讲总结	266
19.1 背景知识	170	第 28 讲 非缓冲文件系统	267
19.2 范例	174	28.1 背景知识	267
19.3 本讲总结	178	28.2 范例	268
第 20 讲 指针的指针	179	28.3 本讲总结	269
20.1 背景知识	179	第 29 讲 常用图形和屏幕函数	270
20.2 范例	182	29.1 背景知识	270
20.3 本讲总结	184	29.2 范例	288
第 21 讲 结构、联合、枚举	185	29.3 本讲总结	291
21.1 背景知识	185	第 30 讲 编写弹出式菜单	292
21.2 范例	195	30.1 背景知识	292
21.3 本讲总结	199	30.2 范例	292
第 22 讲 日期与时间	200	30.3 本讲总结	308
22.1 背景知识	200	第 31 讲 编写下拉式菜单	309
22.2 范例	203	31.1 背景知识	309
22.3 本讲总结	208	31.2 范例	309
第 23 讲 位与字节	209	31.3 本讲总结	321
23.1 背景知识	209	第 32 讲 利用键盘绘图	322
23.2 范例	217	32.1 背景知识	322
23.3 本讲总结	220	32.2 范例	322
第 24 讲 编译预处理	221	32.3 本讲总结	343
24.1 背景知识	221	第 33 讲 实现 BMP 图像的显示	344
24.2 范例	228	33.1 背景知识	344
24.3 本讲总结	230	33.2 范例	346
第 25 讲 动态分配	232	33.3 本讲总结	353
25.1 背景知识	232	第 34 讲 渐入、渐出的屏幕显示方法	354

34.1 背景知识	354	42.2 范例	429
34.2 范例	354	42.3 本讲总结	432
34.3 本讲总结	362	第 43 讲 数据结构的基本概念	433
第 35 讲 WINDOWS 中 C 语言编程	363	43.1 背景知识	433
35.1 背景知识	363	43.2 范例	440
35.2 范例	372	43.3 本讲总结	442
35.3 本讲总结	376	第 44 讲 链表的 C 语言表示	443
第 36 讲 创建 WINDOWS 实例		44.1 背景知识	443
程序 GENERIC	377	44.2 范例	449
36.1 背景知识	377	44.3 本讲总结	460
36.2 范例	377	第 45 讲 栈、队列的 C 语言表示	461
36.3 本讲总结	382	45.1 背景知识	461
第 37 讲 FOXBASE 数据库加密	383	45.2 范例	466
37.1 背景知识	383	45.3 本讲总结	473
37.2 范例	384	第 46 讲 二叉数的 C 语言表示	474
37.3 本讲总结	386	46.1 背景知识	474
第 38 讲 C 与 FOXBASE 的接口	387	46.2 范例	479
38.1 背景知识	387	46.3 本讲总结	483
38.2 范例	387	第 47 讲 C 与汇编语言的接口	484
38.3 本讲总结	393	47.1 背景知识	484
第 39 讲 用 TURBO C 直接读取		47.2 范例	488
FOXPRO 方法	394	47.3 本讲总结	493
39.1 背景知识	394	第 48 讲 内存驻留程序设计	494
39.2 范例	395	48.1 背景知识	494
39.3 本讲总结	399	48.2 范例	501
第 40 讲 串行通信	400	48.3 本讲总结	519
40.1 背景知识	400	第 49 讲 图形模式下显示汉字	520
40.2 范例	402	49.1 背景知识	520
40.3 本讲总结	412	49.2 范例	523
第 41 讲 用 SOCKETS 实现网络实时通信	413	49.3 本讲总结	526
41.1 背景知识	413	第 50 讲 C 语言编程规范	527
41.2 范例	421	50.1 背景知识	527
41.3 本讲总结	423	50.2 范例	529
第 42 讲 CGI 应用程序	424	50.3 本讲总结	530
42.1 背景知识	424		

第0讲 预备知识

0.1 C语言的特点

C语言有很多独到的特点，不然它怎么会流行得这么广呢？让我们来看一看C语言到底有什么样的特点。

1. 程序简洁

用C语言编写的程序非常简洁，C语言只有32个关键字，9种控制语句，程序主要由小写字母组成，书写格式自由。它压缩了其他高级语言中的冗余的部分，让我们来看一下C语言同Pascal语言和Basic语言的比较，如图0-1所示。

表0-1 C语言同Pascal语言和Basic语言的比较

C语言	Pascal语言	Basic语言	含义
{ }	BEGIN END	无	复合语句
if (条件) 语句	IF (条件) THEN 语句	IF (条件) THEN 语句	条件语句
int x;	VAR X:INTEGER	DIM X	定义一个整型变量
int y[5];	VAR Y:ARRAY[0..4]OF INTEGER	DIM X (5)	定义一个整型数组
int f ()	FUNCTION f () :INTEGER	SUB F	定义返回值为整型的函数
int *p	VAR P:^INTEGER	无	定义一个整型指针
x+=3	x=x+3	x=x+3	赋值语句
i++	i=i+1	i=i+1	自增语句

2. 运算符丰富

C语言中共有34种运算符，可以实现在其他高级语言中难以实现的功能。

3. 数据结构丰富

C语言中的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型等，能够用来实现非常复杂的数据结构。如线性表、链表、栈、队列、树、图等。

4. 具有结构化的控制语句

如if () else、while、switch等。并用函数作为程序模块，是理想的结构化程序设计语言。

5. 语法限制不太严格

如对数组下标越界不作检查，各种类型的变量可以通用。

6. 直接访问物理内存

C 语言允许直接访问物理内存，能实现对位的操作。

7. 目标代码质量高

C 语言生成的目标代码质量高，只比汇编语言生成的代码效率低 20% 左右。

8. 可移植性好

C 程序的可移植好，基本上不用修改就可用于其他型号的计算机。

以上是 C 语言的特点，我们将在今后的学习过程中一点点地去体会。

0.2 C 语言的开发环境

0.2.1 Turbo C 2.0 的安装和启动

Turbo C 2.0 的安装非常简单，只要将 1#盘插入 A 驱动器中，在 DOS 的 A> 下键入：A>INSTALL 即可，此时屏幕上显示三种选择：

(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。

这里假定按第一种选择进行安装，只要在安装过程中按对盘号的提示，顺序插入各个软盘，就可以顺利地进行安装，安装完毕后将在 C 盘根目录下建立一个 TC 子目录，TC 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。运行 Turbo C 2.0 时，只要在 TC 子目录下键入 TC 并回车即可进入 Turbo C 2.0 集成开发环境。

0.2.2 Turbo C 2.0 集成开发环境的使用

进入 Turbo C 2.0 集成开发环境中后，屏幕上显示如图 0-1 所示。



图 0-1 Turbo C 集成开发环境主界面

其中, 顶上一行为 Turbo C 2.0 主菜单, 中间窗口为编辑区, 接下来是信息窗口, 最底下一行为参考行。这 4 个窗口构成了 Turbo C 2.0 的主屏幕, 以后的编程、编译、调试以及运行都将在这个主屏幕中进行。下面详细介绍主菜单的内容。

主菜单在 Turbo C 2.0 主屏幕顶上一行, 显示下列内容: File Edit Run Compile Project Options Debug Break/watch。

除 Edit 外, 其他各项均有子菜单, 只要用 Alt 加上某项中第一个字母 (即大写字母), 就可进入该项的子菜单中。

1. File (文件) 菜单

按 Alt+F 可进入 File 菜单。该菜单包括以下内容, 如图 0-2 所示。



图 0-2 File 菜单

(1) 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。该项的热键为 F3, 即只要在主菜单中按 F3 即可进入该项, 而不需要先进入 File 菜单再选此项。

(2) 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, 选择后将该程序装入编辑区, 并将光标置在上次修改过的地方。其热键为 Alt+F3。

(3) New (新文件)

说明文件是新的, 缺省文件名为 NONAME.C, 存盘时可改名。

(4) 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, 其热键为 F2。

(5) Write to (存盘)

可由用户给出文件名来将编辑区中的文件存盘, 若该文件已存在, 则询问是否要覆盖。

(6) Directory (目录)

显示目录及目录中的文件, 并可由用户选择。

(7) Change dir (改变目录)

显示当前目录，用户可以改变显示的目录。

(8) Os shell (暂时退出)

暂时退出 Turbo C 2.0 到 DOS 提示符下，此时可运行 DOS 命令，若想回到 Turbo C 2.0 中，只要在 DOS 状态下键入 EXIT 即可。

(9) Quit (退出)

退出 Turbo C 2.0，返回到 DOS 操作系统中，其热键为 Alt+X。

说明：以上各项可用光标键移动色棒进行选择，回车则执行。也可用每一项的第一个大写字母直接选择。若要退到主菜单或从它的下一级菜单列表框退回均可用 Esc 键，Turbo C 2.0 的所有菜单均采用这种方法进行操作，以下不再说明。

2. Run (运行) 菜单

按 Alt+R 可进入 Run 菜单，该菜单有以下内容，如图 0-3 所示。

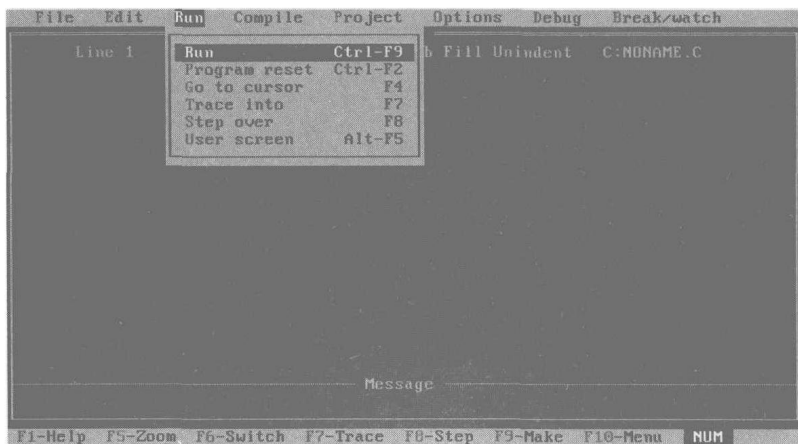


图 0-3 Run 菜单

(1) Run (运行程序)

运行由 Project/Project name 项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，其热键为 Ctrl+F9。

(2) Program reset (程序重启)

中止当前的调试，释放分给程序的空间，其热键为 Ctrl+F2。

(3) 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。其热键为 F4。

(4) 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 项，则执行长条将跟踪到该子函数内部去执行，其热键为 F7。

(5) 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，其热键为 F8。

(6) User screen (用户屏幕)

显示程序运行时在屏幕上显示的结果。其热键为 Alt+F5。

3. Compile (编译) 菜单

按 Alt+C 可进入 Compile 菜单, 该菜单有以下几个内容, 如图 0-4 所示。

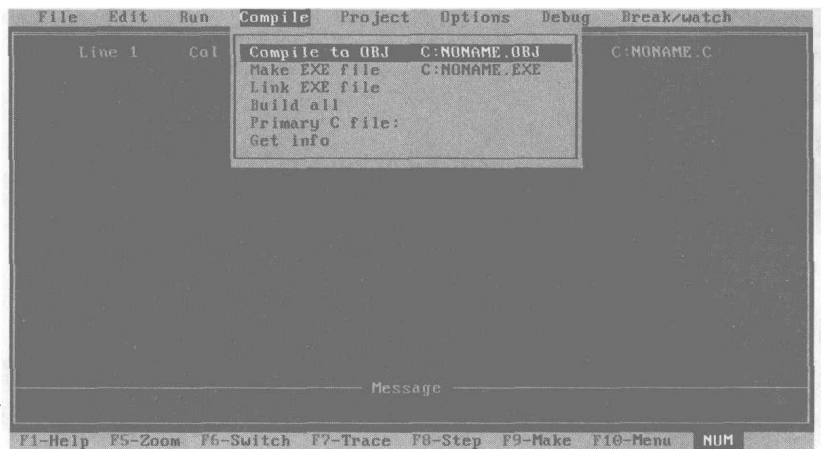


图 0-4 Compile 菜单

(1) 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。其热键为 Alt+F9。

(2) Make EXE file (生成执行文件)

此命令生成一个 .EXE 文件, 并显示生成的 .EXE 文件名。其中, .EXE 文件名是下面几项之一。

- (1) 由 Project/Project name 说明的项目文件名。
- (2) 若没有项目文件名, 则由 Primary C file 说明的源文件。
- (3) 若以上两项都没有文件名, 则为当前窗口的文件名。

(3) 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。

(4) 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)。

(5) Primary C file (主 C 文件)

当在该项中指定了主文件后, 在以后的编译中, 如没有项目文件名, 则编译此项中规定的主 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。

(6) 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。

4. Project (项目) 菜单

按 Alt+P 可进入 Project 菜单, 该菜单包括以下内容, 如图 0-5 所示。

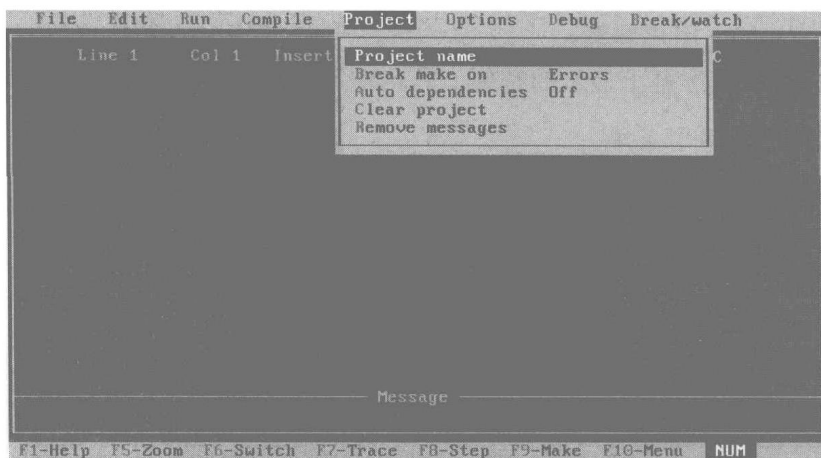


图 0-5 Project 菜单

(1) 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, 要将这三个文件编译装配成一个 file.exe 的执行文件, 可以先建立一个 file.prj 的项目文件, 其内容如下:

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

此时将 file.prj 放入 Project name 项中, 以后进行编译时将自动对项目文件中规定的三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。例如有 file1.obj, file2.c, file3.c 三个文件, 将不对 file1.c 进行编译, 而直接连接。

说明: 当项目文件中的每个文件无扩展名时, 均按源文件对待, 另外, 其中的文件也可以是库文件, 但必须写上扩展名.LIB。

(2) Break make on (中止编译)

由用户选择是否在有 Warning (警告)、Errors (错误)、Fatal Errors (致命错误) 时或 Link (连接) 之前退出 Make 编译。

(3) 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。

(4) Clear project (清除项目文件)

清除 Project/Project name 中的项目文件名。

(5) Remove messages (删除信息)

把错误信息从信息窗口中清除掉。

5. Options (选择菜单)

按 Alt+O 可进入 Options 菜单, 该菜单对初学者来说要谨慎使用, 如图 0-6 所示。

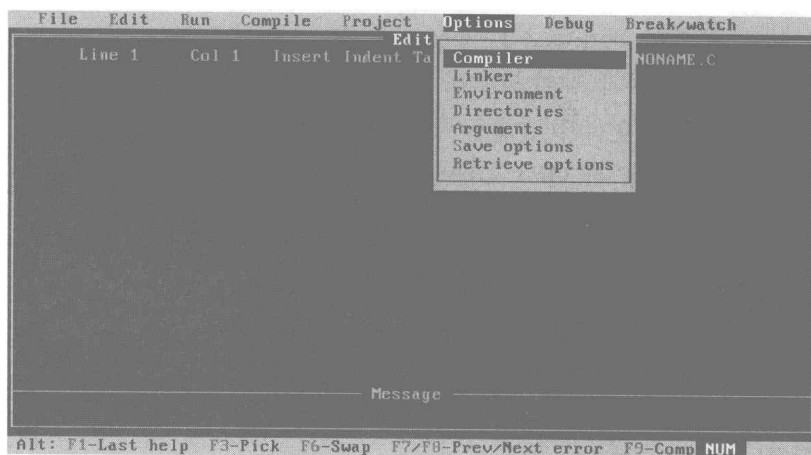


图 0-6 Options 菜单

(1) Compiler (编译器), 如图 0-7 所示。

本项选择又有许多子菜单, 可以让用户选择硬件配置、存储模型、调试技术、代码优化、对话信息控制和宏定义。

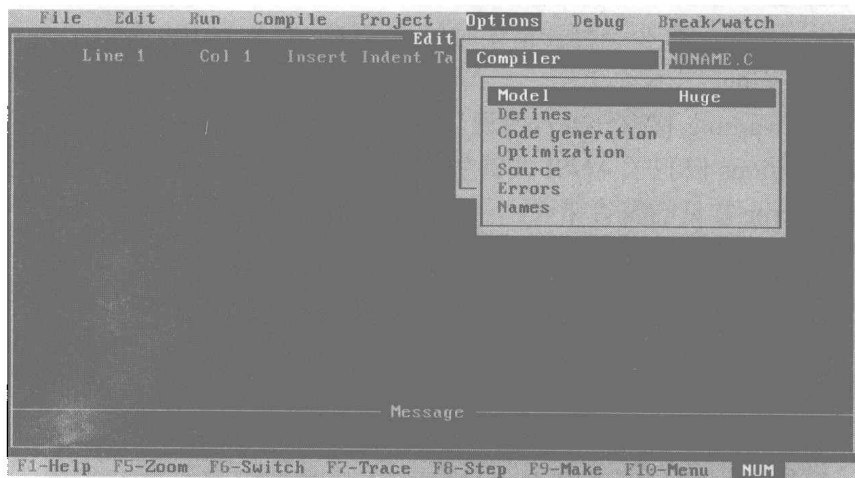


图 0-7 Compiler 子菜单

这些子菜单如下:

- Model 共有 Tiny, small, medium, compact, large, huge 六种不同模式可由用户选择。
- Define 打开一个宏定义框, 用户可输入宏定义。多重定义可用分号, 赋值可用等号。
- Code generation 它又有许多任选项, 这些任选项告诉编译器产生什么样的目标代码。
 - a) Calling convention 可选择 C 或 Pascal 方式传递参数。
 - b) Instruction set 可选择 8088/8086 或 80186/80286 指令系列。
 - c) Floating point 可选择仿真浮点、数学协处理器浮点或无浮点运算。
 - d) Default char type 规定 char 的类型。
 - e) Alignment 规定地址对准原则。
 - f) Merge duplicate strings 作优化用, 将重复的字符串合并在一起。

- g) Standard stack frame 产生一个标准的栈结构。
- h) Test stack overflow 产生一段程序运行时检测堆栈溢出的代码。
- i) Line number 在.OBJ 文件中放进行号以供调试时用。
- j) OBJ debug information 在.OBJ 文件中产生调试信息。
- Optimization
 - a) Optimize for 选择是对程序小型化还是对程序速度进行优化处理。
 - b) Use register variable 用来选择是否允许使用寄存器变量。
 - c) Register optimization 尽可能使用寄存器变量以减少过多的取数操作。
 - d) Jump optimization 通过去除多余的跳转和调整循环与开关语句的办法, 压缩代码。
- Source
 - a) Identifier length 说明标识符有效字符的个数, 默认为 32 个。
 - b) Nested comments 是否允许嵌套注释。
 - c) ANSI keywords only 是只允许 ANSI 关键字, 还是也允许 Turbo C 2.0 关键字。
- Error
 - a) Error stop after 多少个错误时停止编译, 默认为 25 个。
 - b) Warning stop after 多少个警告错误时停止编译, 默认为 100 个。
 - c) Display warning
 - d) Portability warning 移植性警告错误。
 - e) ANSI Violations 侵犯了 ANSI 关键字的警告错误。
 - f) Common error 常见的警告错误。
 - g) Less common error 少见的警告错误。
- Names 用于改变段(segment)、组(group)和类(class)的名字, 默认值为 CODE, DATA, BSS。

(2) Linker (连接器), 如图 0-8 所示。

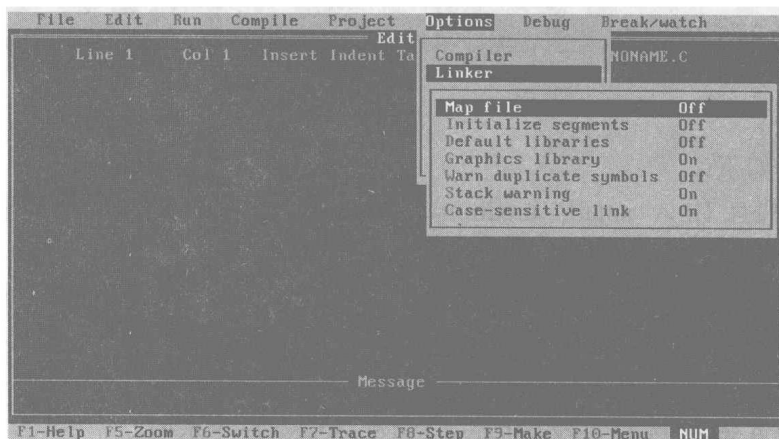


图 0-8 Linker 子菜单

本菜单设置有关连接的选择项, 它有以下内容:

- Map file menu 选择是否产生.MAP 文件。
- Initialize segments 是否在连接时初始化没有初始化的段。