

普通高等教育“十一五”国家级规划教材配套实验指导教材

C 语言程序设计实验指导及题解

黄迪明 许家珩 胡德昆 孙立欧 编著

电子科技大学出版社

图书在版编目 (CIP) 数据

C 语言程序设计实验指导及题解 / 黄迪明等编著. —成都: 电子科技大学出版社, 2008.7

普通高等教育“十一五”国家级规划教材配套实验指导教材

ISBN 978-7-81114-818-3

I. C… II. 黄… III. C 语言—程序设计—高等学校—教学参考资料

IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 075621 号

内 容 简 介

本书是普通高等教育“十一五”国家级规划教材《C 语言程序设计》的上机学习指导书。其主要针对主教材课后习题和本书实验项目进行了详细解答, 并且介绍了 Turbo 2.0/3.0 及 Visual C++ 6.0 编译系统的使用, 对 C 语言程序的编辑、编译、连接、运行和调试等方法进行了详细介绍和指导。全书共分四个部分, 主要内容包括: C 语言上机指导、C 语言实验项目、C 语言课程设计、教材习题和实验项目题解。

本书着眼于教材中的重点和难点知识分析以及习题解析, 为学习者指出了在 C 语言上机学习过程中需要注意的问题。希望学习者经过本书的指导学习以后, 在解决更多的疑难问题上得到指点和帮助。

普通高等教育“十一五”国家级规划教材配套实验指导教材

C 语言程序设计实验指导及题解

黄迪明 许家珩 胡德昆 孙立欧 编著

出 版: 电子科技大学出版社 (成都市一环路东一段 159 号电子信息产业大厦 邮编: 610051)

策划编辑: 吴艳玲

责任编辑: 吴艳玲

主 页: www.uestcp.com.cn

电子邮箱: uestcp@uestcp.com.cn

发 行: 新华书店经销

印 刷: 成都蜀通印务有限责任公司

成品尺寸: 185mm×260mm 印张 9.5 字数 240 千字

版 次: 2008 年 7 月第一版

印 次: 2008 年 7 月第一次印刷

书 号: ISBN 978-7-81114-818-3

定 价: 16.90 元

■ 版权所有 侵权必究 ■

- ◆ 本社发行部电话: 028-83202463; 本社邮购电话: 028-83208003。
- ◆ 本书如有缺页、破损、装订错误, 请寄回印刷厂调换。
- ◆ 课件下载在我社主页“下载专区”。

前 言

内容简介

本书是普通高等教育“十一五”国家级规划教材《C 语言程序设计》的上机学习指导书。其主要针对主教材课后习题和本书实验项目进行了详细解答，并且介绍了 Turbo 2.0 /3.0 及 Visual C++ 6.0 编译系统的使用，对 C 语言程序的编辑、编译、连接、运行和调试等方法进行了详细介绍和指导。全书共分四个部分，主要内容包括：C 语言上机指导、C 语言实验项目、C 语言课程设计、教材习题和实验项目题解。

本书着眼于教材中的重点和难点知识分析以及习题解析，为学习者指出了在 C 语言上机学习过程中需要注意的问题。希望学习者经过本书的指导学习以后，在解决更多的疑难问题上得到指点和帮助。

本书特色

1. 本习题指导针对主教材中的习题进行了全面的解答。在提供正确的源程序的基础上，增加了算法分析、结果分析和易错点提示等内容，对于学习者正确地、系统地理解和学习 C 语言很有帮助。
2. 在实验项目中，增加了一项综合性强、应用范围广的课程设计。这项设计不仅可以加深学习者对教材内容的领悟，同时也提高了学习者对 C 语言知识的综合运用能力。除此之外，还能激发学习者的学习热情和兴趣。
3. 本习题指导参考了很多资深程序员的编程经验。在讲解知识的同时，还提供了很多优秀的编程风格和习惯，更能培养学习者实际的软件开发能力。
4. 本习题指导的附录收集了很多编程的常见错误以及编译器的出错信息，使得本书在指导习题的同时，成为了一本 C 语言学习和开发的工具书。学习者通过本习题指导，可以在自身的软件开发能力方面得到全面的提高。

鸣谢

本书由电子科技大学黄迪明、许家珩、胡德昆，西南科技大学孙立欧编写。在本书的编写过程中，还得到了杜海涛、张大愚、邹波、张紫微、刘鹏等人的热情帮助，在此表示诚挚的谢意！

由于我们的水平和时间有限，书中难免存在缺点和错误，敬请读者和同行专家指正。

编 者

2008 年 5 月于电子科技大学

目 录

第 1 章 C 语言程序的上机步骤.....	1
第 2 章 Turbo C 2.0/3.0 使用指南.....	4
2.1 Turbo C 简介	4
2.2 Turbo C 2.0 文件简介	4
2.3 Turbo C 2.0 的启动	5
2.4 Turbo C 2.0 集成开发环境的使用	5
2.5 Turbo C 2.0 的配置文件	12
2.6 Turbo C 调试系统	14
2.7 常见的编译错误和程序调试.....	18
2.7.1 常见的错误类型	18
2.7.2 程序调试.....	23
第 3 章 Visual C++环境下运行 C 程序.....	26
3.1 启动 VC++	26
3.2 新建/打开 C 程序文件.....	27
3.3 程序的编辑和保存.....	27
3.4 执行程序	28
3.5 关闭程序工作区.....	30
3.6 命令行参数处理.....	30
3.7 程序调试	31
第 4 章 实验概述	34
4.1 本课程实验的任务.....	34
4.2 本课程实验简介.....	34
4.3 本课程适用专业.....	34
4.4 本课程实验涉及核心知识点.....	34
4.5 本课程实验重点与难点.....	34

第 5 章 实验项目和课程设计	35
5.1 实验项目一	35
5.2 实验项目二	35
5.3 实验项目三	36
5.4 实验项目四	36
5.5 实验项目五	37
5.6 实验项目六	37
5.7 课程设计	37
5.7.1 课程设计一	37
5.7.2 课程设计二	38
5.7.3 课程设计三	38
5.7.4 课程设计四	38
第 6 章 主教材习题参考答案	39
6.1 习题 1	39
6.2 习题 2	42
6.3 习题 3	51
6.4 习题 4	66
6.5 习题 5	83
6.6 习题 6	91
6.7 习题 7	109
6.8 习题 8	117
6.9 习题 9	121
6.10 课程设计解答	132
6.10.1 设计一	132
6.10.2 设计二	133
6.10.3 设计三	135
6.10.4 设计四	137
附 录	139
附录一 C 语言常见错误总结	139
一、编程的常见错误	139
二、连接时的常见错误	140
三、运行时的常见错误	141
附录二 C 语言编译常见错误信息及处理方法	141

第1章 C语言程序的上机步骤

按照C语言语法规则编写的C程序称为C源程序。在主教材中，我们已经了解到，源程序由字母、数字及其他符号（标识符）等构成，在计算机内部用相应的ASCII码表示，并保存为扩展名为“.c”源程序文件（在本书中又被称为源代码）。源程序是无法直接被计算机运行的，因为计算机的CPU只能执行二进制的机器指令。这就需要将源程序的ASCII码先“翻译”成机器指令，然后计算机的CPU才能运行翻译好的程序，也就是ASCII码。源程序“翻译”过程由两个步骤实现：编译与连接。首先对源程序进行编译处理，即把每一条语句用若干条机器指令来实现，以生成由机器指令组成的目标程序。但目标程序还不能马上交给计算机直接运行，因为在源程序中，常用函数（如scanf、printf）运算并不是用户自己编写的，而是直接调用系统函数库中的库函数。因此，必须把“库函数”的处理过程连接到经编译生成的目标程序中，生成可执行程序，并经机器指令的地址重定位，才可由计算机运行，最终得到结果。

C语言程序的调试、运行步骤如图1.1所示。

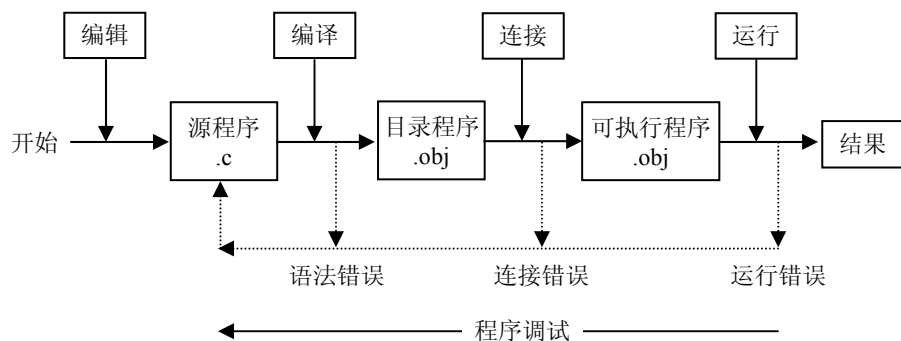


图 1.1 C 程序的运行步骤

在图 1.1 中，实线表示程序从开始编辑到得到结果的过程，虚线表示当某一步骤出现错误时的修改路线。程序在运行时，无论是出现编译错误、连接错误，还是运行结果不正确（源程序中有语法错误或逻辑错误），都需要修改源程序，并对它重新编译、连接和运行，直至将程序调试正确为止。

在积累了一定的程序设计经验后，我们会了解到：除了较简单的情况，一般的程序很难一次就能做到完全正确。在上机过程中，根据出错现象找出错误并改正称为程序调试。我们要在学习程序设计的实践过程中，逐步培养调试程序的能力。它不可能靠几句话讲清楚，要靠自己在上机中不断摸索总结，可以说是一种经验积累。

程序中的错误大致可分为三类:

1. 程序编译时检查出来的语法错误。
2. 连接时出现的错误。
3. 程序执行过程中的错误。

编译错误通常是编程者违反了 C 语言的语法规则,如保留字输入错误、花括号不匹配和语句少分号等。连接错误一般由未定义或未指明要连接的函数,或者函数调用不匹配等因素引起,对系统函数的调用必须要通过“include”说明。

对于编译连接错误, C 语言编译系统会提供出错信息,包括出错位置(行号)、出错提示信息等。编程者可以根据这些信息,找出相应错误所在并修改。有时系统提示了一大串错误信息,并不表示真的有这么多个错误,往往是因为程序中的一两个错误带来的,所以当纠正了几个错误后,应该重新编译连接一次,然后根据最新的出错信息继续纠正,这是程序调试的一个好方法。在本书中将会介绍更多的程序调试方法供学习者参考,但如果要较好地掌握,需要多多进行上机实践,积累调试的经验。

有些程序通过了编译连接,并能够在计算机上运行,但得到的结果和预期的结果不一样,这类错误被称为逻辑错误。这类在程序执行过程中的错误往往难以改正。错误的原因一部分是程序书写错误带来的,例如应该使用变量 x 的地方写成了变量 y,虽然没有语法错误,但意思完全错了;另一部分可能是程序的算法不正确,解题思路不对,得到的结果和预期的结果不一样,例如预期求两个整数的和,在程序中却写为两个整数的差,得到的结果肯定会和预期的不一样;还有一些程序计算结果有时正确,有时不正确,例如求一个输入整数除以 2 的商,如果将这个商定义为 int 型变量,那么在该整数为偶数时正确,奇数时就会错误,这些现象往往是编程时对各种情况考虑不周所致。

解决运行错误的首要步骤就是错误定位,即找到出错的位置和错误的原因,才能予以纠正。通常我们需要先设法确定错误的大致位置,然后通过 C 语言提供的调试工具找出真正的错误。但需要大家注意的是,在本书中,大部分的程序在调试时,调试工具都能直接找到程序的错误,但也有部分比较复杂的程序,当程序执行出错时,调试工具发现的错误未必就一定是程序中的真正错误,这些例子我们会在第三部分中为大家介绍。在将来的实际软件开发中,这样的情况将会出现得更多,所以有经验的程序员往往都认为,寻找程序的错误不能只依靠计算机,也需要我们自己掌握好的方法,在实际的程序调试中积累丰富的经验。以下就将介绍几种比较好的方法:

为了确定错误的大致位置,可以先把程序分成几大块,并在每一块的结束位置,手工计算一个或几个阶段性结果,然后用调试方式运行程序,到每一程序块结束时,检查程序运行的实际结果与手工计算是否一致,通过这些阶段性结果来确定各程序块是否正确。对于出错的程序块,可逐条仔细检查各语句,找出错误所在。如果出错块程序较长,难以一下子找出错误,可以进一步把该块细分成更小的块,按照上述步骤进一步检查。在确定了大致出错位置后,如果无法直接看出错误,可以通过单步运行相关位置的几条语句,逐条检查,这样肯定能找出错误的语句。

当程序出现计算结果有时正确有时不正确的情况时,其原因一般是算法对各种数据处理情况考虑不全面。解决办法最好多选几组典型的输入数据进行测试,除了普通的数据外,还

应包含一些边界数据和不正确的数据。比如确定正常的输入数据范围后，分别以最小值、最大值、比最小值小的值和比最大值大的值，多方面运行检查自己的程序。能够处理边界数据和不正确的数据，也是程序健壮性的体现。

本书第二、三章分别以 Turbo C 2.0 和 Visual C++ 6.0 为上机平台，对 C 程序编译、连接和调试进行简单介绍。建议一开始学习上机时，把注意力放在程序的编译、连接和运行，以能运行为目标，而把调试部分放到学习了教材第四章内容后再看。只有具有了一定的程序语句量，调试才有作用。

第 2 章 Turbo C 2.0/3.0 使用指南

2.1 Turbo C 简介

Turbo C 是美国 Borland 公司的产品, Borland 公司是一家专门从事软件开发、研制的大公司。该公司相继推出了 Turbo 系列软件, 如 Turbo BASIC, Turbo Pascal, Turbo Prolog, 这些软件很受用户欢迎。该公司在 1987 年首次推出 Turbo C 1.0 产品, 其中使用了全然一新的集成开发环境, 即使用了一系列下拉式菜单, 将文本编辑、程序编译、连接以及程序运行一体化, 大大方便了程序的开发。1988 年, Borland 公司又推出 Turbo C 1.5 版本, 增加了图形库和文本窗口函数库等, 而 Turbo C 2.0 则是该公司 1989 年出版的。Turbo C 2.0 在原来集成开发环境的基础上增加了查错功能, 并可以在 Tiny 模式下直接生成 “.COM ” (数据、代码和堆栈处在同一 64KB 内存中) 文件, 还可对数学协处理器 (支持 8087/80287/80387 等) 进行仿真。Borland 公司后来又推出了面向对象的程序软件包 Turbo C++, 它继承发展了 Turbo C 2.0 的集成开发环境, 并包含了面向对象的基本思想和设计方法。1991 年为了适用 Microsoft 公司的 Windows 3.0 版本, Borland 公司又将 Turbo C++ 进行了更新, 即 Turbo C 的新一代产品 Borland C++。

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, 并且它可在任何彩、单色 80 列监视器上运行。它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。

2.2 Turbo C 2.0 文件简介

下面对 Turbo C 2.0 的主要文件进行简单介绍:

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
C0?.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
*.BGI 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）。

2.3 Turbo C 2.0 的启动

安装完毕的Turbo C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，如图2.1所示。



图2.1 Turbo C 2.0集成开发环境

2.4 Turbo C 2.0 集成开发环境的使用

进入Turbo C 2.0集成开发环境后，屏幕显示画面如图2.1所示。

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。这四个窗口构成了Turbo C 2.0的主屏幕，以后的程序编辑、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。

1. 主菜单

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】键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】键可进入File菜单，该菜单包括以下内容：

- 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】键即可进入该项，而不需要先进入File菜单再选此项。

- 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】。

- 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。

- 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】。

- 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 2.0到DOS提示符下，此时可以运行DOS命令，若想回到Turbo C 2.0中，只要在DOS状态下键入EXIT即可。

- 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所有菜单均采用这种方法进行操作，以下不再说明。

（2）Edit（编辑）菜单

按【Alt】+【E】键可进入编辑菜单，若再回车，则光标出现在编辑窗口，此时用户可以进行文本编辑。可用【F1】键获得有关编辑方法的帮助信息。

与编辑有关的功能键如下：

【F1】键：获得Turbo C 2.0编辑命令的帮助信息

【F5】键：扩大编辑窗口到整个屏幕

【F6】键：在编辑窗口与信息窗口之间进行切换

【F10】键：从编辑窗口转到主菜单

编辑命令简介：

【PageUp】键：向前翻页

【PageDn】键：向后翻页

【Home】键：将光标移到所在行的开始

【End】键：将光标移到所在行的结尾

【Ctrl】+【Y】键：删除光标所在的一行

【Ctrl】+【T】键：删除光标所在处的一个词

【Ctrl】+【K】 【B】键：设置块开始

【Ctrl】+【K】 【K】键：设置块结尾

【Ctrl】+【K】 【V】键：块移动

【Ctrl】+【K】 【C】键：块拷贝

【Ctrl】+【K】 【Y】键：块删除

【Ctrl】+【K】 【R】键：读文件

【Ctrl】+【K】 【W】键：存文件

【Ctrl】+【K】 【P】键：块文件打印

【Ctrl】+【F1】键：如果光标所在处为Turbo C 2.0库函数，则获得有关该函数的帮助信息

【Ctrl】+【Q】键：查找Turbo C 2.0双界符的后匹配符

【Ctrl】+【Q】 【】键：查找Turbo C 2.0双界符的前匹配符

说明：

a. Turbo C 2.0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]

注释号 /* 和 */

双引号 "

单引号 '

b. Turbo C 2.0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】+【O】 【L】键为自动缩进开关的控制键。

(3) Run (运行) 菜单

按【Alt】+【R】键可进入Run菜单，该菜单有以下各项：

● 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。如果对上次编译后的

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】。

- 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】。

- 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】。

- 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】。

- 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】。

- 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】。

(4) Compile (编译) 菜单

按【Alt】+【C】可进入Compile菜单,该菜单有以下几个内容:

- 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】。

- 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。其中.EXE文件名是下面几项之一。

- a. 由 Project/Project name 说明的项目文件名。
- b. 若没有项目文件名,则是由 Primary C file 说明的源文件。
- c. 若以上两项都没有文件名,则为当前窗口的文件名。

- 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。

- 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。该命令不进行过时检查。

(上面的几条命令要进行过时检查,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。

- Get info

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

(5) Project (项目) 菜单

按【Alt】+【P】键可进入Project菜单，该菜单包括以下内容：

- 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和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：

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

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。

- Break make on（中止编译）

由用户选择是否在有Warning（警告）、Errors（错误）和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/Project name中的项目文件名。

- Remove messages（删除信息）

把错误信息从信息窗口中清除掉。

（6）Options（选择菜单）

按【Alt】+【O】键可进入Options菜单，该菜单对初学者来说要谨慎使用。

- Compiler（编译器）

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。这些子菜单如下：

- Model：共有Tiny、small、medium、compact、large和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。

- ◆ Calling convention: 可选择C或Pascal方式传递参数。
- ◆ Instruction set: 可选择8088/8086或80186/80286指令系列。
- ◆ Floating point: 可选择仿真浮点、数学协处理器浮点或无浮点运算。
- ◆ Default char type: 规定char的类型。
- ◆ Alignment: 规定地址对准原则。
- ◆ Merge duplicate strings: 进行优化用, 将重复的字符串合并在一起。
- ◆ Standard stack frame: 产生一个标准的栈结构。
- ◆ Test stack overflow: 产生一段程序运行时检测堆栈溢出的代码。
- ◆ Line number: 在.OBJ文件中放进行号以供调试时用。
- ◆ OBJ debug information: 在.OBJ文件中产生调试信息。

■ Optimization

- ◆ Optimize for: 选择是对程序小型化还是对程序速度进行优化处理。
- ◆ Use register variable: 用来选择是否允许使用寄存器变量。
- ◆ Register optimization: 尽可能使用寄存器变量以减少过多的取数操作。
- ◆ Jump optimization: 通过去除多余的跳转和调整循环与开关语句的办法以压缩代码。

■ Source

- ◆ Identifier length: 说明标识符有效字符的个数, 默认为32个。
- ◆ Nested comments: 是否允许嵌套注释。
- ◆ ANSI keywords only: 是只允许ANSI关键字还是也允许Turbo C 2.0关键字。

■ Error

- ◆ Error stop after: 多少个错误时停止编译, 默认为25个。
- ◆ Warning stop after: 多少个警告错误时停止编译, 默认为100个。
- ◆ Display warning: 显示警告信息。
- ◆ Portability warning: 移植性警告错误。
- ◆ ANSI Violations: 侵犯了ANSI关键字的警告错误。
- ◆ Common error: 常见的警告错误。
- ◆ 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和BSS。

● Linker (连接器)

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。
- Default libraries: 是否在连接其他编译程序产生的目标文件时去寻找缺省库。
- Graphics library: 是否连接graphics库中的函数。

- Warn duplicate symbols: 当有重复符号时产生警告信息。
- Stack warning: 是否让连接程序产生No stack的警告信息。
- Case-sensitive link: 是否区分大、小写字。

- Environment (环境)

本菜单规定是否对某些文件自动存盘及制表键和屏幕大小的设置。

- Message tracking: 跟踪邮件。
- Current file: 跟踪在编辑窗口中的文件错误。
- All files: 跟踪所有文件错误。
- Off: 不跟踪。
- Keep message: 编译前是否清除Message窗口中的信息。
- Config auto save: 选on时, 在Run、Shell或退出集成开发环境之前, 如果Turbo C 2.0的配置被改过, 则所做的改动将存入配置文件中; 选off时不存。
- Edit auto save: 是否在Run或Shell之前, 自动存储编辑的源文件。
- Backup file: 是否在源文件存盘时产生后备文件(.BAK文件)。
- Tab size: 设置制表键大小, 默认为8。
- Zoomed windows: 将现行活动窗口放大到整个屏幕, 其热键为【F5】。
- Screen size: 设置屏幕文本大小。

- Directories (路径)

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。
- Arguments (命令行参数): 允许用户使用命令行参数。
- Save options (存储配置): 保存所有选择的编译、连接、调试和项目到配置文件中, 缺省的配置文件为TCCONFIG.TC。
- Retrieve options: 装入一个配置文件到TC中, TC将使用该文件的选择项。

(7) Debug (调试) 菜单

按【Alt】+【D】可选择Debug菜单, 该菜单主要用于查错, 它包括以下内容:

- Evaluate (估值)
 - Expression: 要计算结果的表达式。
 - Result: 显示表达式的计算结果。
 - New value: 赋给新值。
- Call stack: 在Turbo C debugger 时用于检查堆栈情况。
- Find function: 在运行Turbo C debugger时用于显示规定的函数。

- Refresh display: 如果编辑窗口偶然被用户窗口重写了可用此恢复编辑窗口的内容。

(8) Break/watch (断点及监视表达式)

按【Alt】+【B】键可进入Break/watch菜单, 该菜单有以下内容:

- Add watch: 向监视窗口插入一监视表达式。
- 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。
- Clear all breakpoints: 清除所有断点。
- View next breakpoint: 将光标移动到下一个断点处。

上面我们只对TC 中主要的功能进行了介绍, 对于其他的操作读者可以自己试验, 或参考有关Turbo C 手册。

(9) 常用快捷键小结

- 【F2】键: 保存程序
- 【F3】键: 调入程序文件
- 【F4】键: 程序运行到光标处暂停
- 【Alt】+【F5】键: 查看运行结果
- 【F7】键: 单步执行 (可进入函数)
- 【F8】键: 单步执行 (不能进入函数)
- 【Ctrl】+【F2】键: 结束程序调试运行
- 【Alt】+【F3】键: 调入最近曾经用过的文件
- 【F5】键: 放大/缩小窗口
- 【F6】键: 窗口切换
- 【Ctrl】+【F7】键: 增加查看变量
- 【Ctrl】+【F8】键: 把光标所在行设为断点或取消断点
- 【F9】键: 编译、连接程序
- 【Alt】+【F9】键: 编译程序
- 【Ctrl】+【F9】键: 编译、连接、执行程序
- 【F10】键: 菜单选择
- 【F1】键: 帮助
- 【Alt】+【x】键: 退出系统

2.5 Turbo C 2.0 的配置文件

所谓配置文件是包含Turbo C 2.0有关信息的文件, 其中存有编译、连接的选择和路径等信息。可以用下述方法建立Turbo C 2.0的配置: