

信息与电子学科百本精品教材工程

新编计算机类本科规划教材

C语言程序设计 实验指导及题解

黄迪明 主编

<http://www.phei.com.cn>



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

新编计算机类本科规划教材

C 语言程序设计

实验指导及题解

黄迪明 主编

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是《C 语言程序设计》的配套辅导书,主要介绍了 Turbo C 2.0/3.0 及 Visual C++ 6.0 编译系统的使用,并对 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。全书分为 4 个部分,主要包括:C 语言上机指导,C 语言实验项目,主教材习题题解,以及实验项目题解。

本书着眼于教材中的重点和难点知识分析以及习题解析,并为读者指出了在 C 语言学习过程中需要注意的问题。

本书可作为高等院校学生学习 C 语言程序设计的辅导教材,以帮助学生学习和提高编程技能。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

C 语言程序设计实验指导及题解/黄迪明主编. —北京:电子工业出版社,2005.7

新编计算机类本科规划教材

ISBN 7-121-01342-8

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(2005)第 056418 号

策划编辑:章海涛

责任编辑:章海涛

印 刷:北京牛山世兴印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:10.5 字数:224 千字

印 次:2005 年 7 月第 1 次印刷

印 数:4000 册 定价:16.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。
联系电话:(010) 68279077。质量投诉请发邮件至 zltz@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

本书相关的主教材《C 语言程序设计》是电子信息类“九五”规划教材《C 语言程序设计》的修订版，是一本程序设计的入门教材，以美国国家标准 C 语言（ANSI C）为基本内容，以当前广为使用的 Turbo C 编译系统为实现的版本，全面系统地介绍了 C 语言及其程序设计方法。

为了帮助读者更好地学习，我们精心编写了本书作为《C 语言程序设计》的配套教材。本书的主要内容包括：C 语言上机指导，C 语言实验项目，主教材习题解答，以及实验项目题解。本书着眼于教材中的重点、难点知识的分析以及习题解析，并由浅入深地为读者解析了 C 语言的高级程序设计实例。

本书由电子科技大学黄迪明、胡德昆、许家珩编写，还得到了杜海涛、王波、刘家芬、陈琼等人的热情帮助，在此表示诚挚的谢意！

限于我们的水平和时间，书中难免存在缺点和错误，敬请读者和同行专家指正。

作 者

2005 年 3 月于电子科技大学

目 录

第一部分 上机指导

第 1 章 C 语言程序的上机步骤	3
第 2 章 Turbo C 2.0/3.0 使用指南	5
2.1 Turbo C 简介	5
2.2 Turbo C 2.0 文件简介	5
2.3 Turbo C 2.0 的启动	6
2.4 Turbo C 2.0 集成开发环境的使用	6
2.5 Turbo C 2.0 的配置文件	13
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
2.8 Turbo C 对于大型程序的开发	26
2.8.1 软件工程化方法	26
2.8.2 大型程序的组织方法	27
2.8.3 建立、运行多文件程序的步骤	27
2.8.4 使用 TLIB 库管理程序	29
第 3 章 Visual C++ 下运行标准 C 程序	31
3.1 启动 Visual C++	31
3.2 新建/打开 C 程序文件	31
3.3 程序保存	32
3.4 执行程序	32
3.5 关闭程序工作区	34
3.6 命令行参数处理	34
3.7 程序调试	35

第二部分 上机实验

第 1 章 实验概述	41
第 2 章 实验内容	42

第三部分 主教材习题参考答案

习题 1	49
习题 2	52
习题 3	62

习题 4	68
习题 5	81
习题 6	89
习题 7	97
习题 8	114
习题 9	118
习题 10	126
习题 11	130

第四部分 实验项目参考答案

1. 实验项目一	143
2. 实验项目二	143
3. 实验项目三	144
4. 实验项目四	145
5. 实验项目五	148
6. 实验项目六	149
7. 实验项目七	155

第一部分 上机指导

程序设计是实践性很强的过程，任何程序最终都必须在计算机上运行，以检验程序的正确性。因此学习程序设计一定要重视上机实践环节，一方面，通过上机读者可以加深对C语言的有关概念的理解，巩固理论知识；另一方面，也可以培养读者程序调试的能力与技巧。

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 程序称为源程序。源程序由字母、数字及其他符号等构成，在计算机内部用相应的 ASCII 码表示，并保存在扩展名为“.c”的文件中。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。这就需把 ASCII 码源程序先翻译成机器指令，然后计算机的 CPU 才能运行翻译好的程序。源程序翻译过程由两个步骤实现：编译和连接。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计算机直接运行，因为在源程序中，输入、输出以及常用函数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。

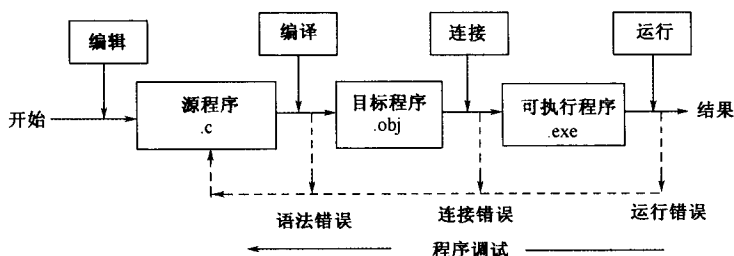


图 1.1 C 程序的运行步骤

图 1.1 中，虚线表示当某一步骤出现错误时的修改路线。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。

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 3 类：

- 程序编译时检查出来的语法错误。
- 连接时出现的错误。
- 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、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通过“include”说明。

对于编译或连接错误，C 语言系统会提供出错信息，包括出错位置（行号）等出错提示信息。编程者可以根据这些信息，找出相应错误所在。有时系统提示的一大串错误信息并不表示真的有这么多个错误，往往是因为前面的一两个错误带来的。所以纠正了几个错误后，应该重新

编译、连接一次，然后根据新的出错信息继续纠正。

有些程序通过了编译、连接，并能够在计算机上运行，但得到的结果不正确，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，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。还有一些程序有时计算结果正确，有时不正确，这种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，才能予以纠正。

通常，我们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。

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，从而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，难以立刻找出错误，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，如果无法直接查出错误，可以通过单步运行相关位置的几条语句，逐条检查，这样可以找出错误的语句。

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。解决办法是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。

本部分的第 2 章和第 3 章分别以 Turbo C 2.0/3.0 和 VC++ 6.0 为上机平台，对 C 程序编译、连接和调试进行了简单介绍。建议读者一开始学习上机时，把注意力放在程序的编译、连接和运行上，以能运行为目标，把调试部分放到学习了主教材第 5 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。该公司相继推出了一套 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，其中使用了焕然一新的集成开发环境，使用了一系列下拉菜单，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 1989 年推出的。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”文件（数据、代码、堆栈处在同一块 64KB 内存中），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++，它继承和发展 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，推出 Turbo C 的新一代产品 Borland C++。

Turbo C 2.0 可运行于 IBM PC 系列上，包括 XT，AT 及 IBM 兼容机，此时要求 DOS 2.0 或更高版本的支持，并至少需要 448KB 的 RAM，可在任何彩、单色 80 列监视器上运行。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 TCHELP.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 例行程序 (源文件)。

上面的“?”表示: 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, 下面还建立了两个子目录\LIB 和\INCLUDE, 分别存放库文件和所有头文件。运行 Turbo C 2.0 时, 只要在 DOS 模式下的\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。

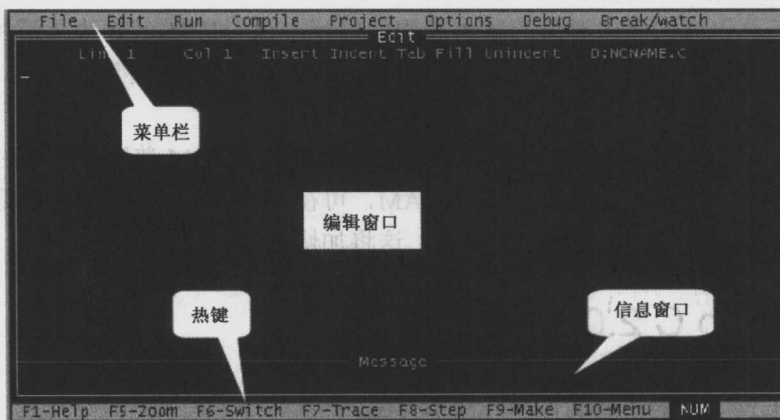


图 2.1 Turbo C 2.0 集成开发环境

2.4 Turbo C 2.0 集成开发环境的使用

进入 Turbo C 2.0 集成开发环境中后, 屏幕上显示如图 2.1 所示: 最上面一行为 Turbo C 2.0 主菜单, 中间窗口为编辑区, 下面是信息窗口, 最下面一行为参考行。这 4 部分构成了 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。

主菜单在 Turbo C 2.0 主屏幕最上面, 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 键加上该项中第一个字母 (即大写字母), 就可进入该项的子菜单中。

1. File (文件)

按 Alt+F 组合键可进入“File”菜单，该菜单包括以下内容：

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 键即可进入该项，而不需要先进入“File”菜单再选此项。

2) Pick (选择)

该菜单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，默认文件名为 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，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将编辑区中的文件存盘，若该文件已存在，则询问是否覆盖。

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。

9) Quit (退出)

该菜单用于退出 Turbo C 2.0，返回到 DOS 操作系统。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)

按 Alt+E 组合键可进入“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。读者按 F1 键可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，获得 Turbo C 2.0 编辑命令的帮助信息。
- F5，扩大编辑窗口到整个屏幕。
- F6，在编辑窗口与信息窗口之间进行切换。
- F10，从编辑窗口转到主菜单。

编辑命令简介：

- PageUp，向前翻页。

- PageDn, 向后翻页。
- Home, 将光标移到所在行的开始。
- End, 将光标移到所在行的末尾。
- Ctrl+Y, 删除光标所在处的一行。
- Ctrl+T, 删除光标所在处的一个词。
- Ctrl+K+B, 设置块开始。
- Ctrl+K+K, 设置块末尾。
- Ctrl+K+V, 块移动。
- Ctrl+K+C, 块复制。
- Ctrl+K+Y, 块删除。
- Ctrl+K+R, 读文件。
- Ctrl+K+W, 存文件。
- Ctrl+K+P, 块文件打印。
- Ctrl+F1, 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/[, 查找 Turbo C 2.0 双界符的后匹配符。
- Ctrl+Q/], 查找 Turbo C 2.0 双界符的前匹配符。

说明:

① Turbo C 2.0 的双界符包括以下几种:

- 花括号{, }。
- 尖括号<, >。
- 圆括号(,)。
- 方括号[,]。
- 注释符/*, */。
- 双引号"。
- 单引号'。

② 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 组合键为自动缩进开关的控制键。

3. Run (运行)

按 Alt+R 组合键可进入“Run”菜单, 该菜单有以下内容:

1) Run (运行程序)

运行由“Project/Project name”项指定的文件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。

2) Program reset (程序重启)

中止当前的调试, 释放分配给程序的空间。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,若用该选项,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到函数内部。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。

4. Compile (编译)

按 Alt+C 组合键可进入“Compile”菜单,该菜单有以下内容:

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,并显示生成的文件名。其中,EXE 文件名是下面几项之一:

- 由“Project/Project name”说明的项目文件名。
- 若没有项目文件名,则为由 Primary C file 说明的源文件。
- 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。该命令不做过时检查。

注意:上面的几条命令要做过时检查,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,在以后的编译中若没有项目文件名,则编译此项中规定的主 C 文件;若编译中有错误,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。

5. Project (项目)

按 Alt+P 组合键可进入“Project”菜单,该菜单包括以下内容:

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 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 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，可直接写上扩展名 “.obj”，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”。

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 时，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。

6. Options (选择菜单)

按 Alt+O 组合键可进入 “Options” 菜单，该菜单对初学者来说要谨慎使用。

1) Compiler (编译器)

本选项有许多子菜单，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。

- Model——有 Tiny, Small, Medium, Compact, Large, Huge 六种模式供用户选择。
- Define——打开一个宏定义框，用户可输入宏定义。多重定义时可用分号，赋值可用等号。
- Code generation——有一些任选项，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——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 文件中加入行号以供调试时用。
 - ◆ OBJ debug information——在 OBJ 文件中产生调试信息。

- Optimization 的任选项如下：
 - ◆ 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、调整循环和开关语句来压缩代码。
- Source 的任选项如下：
 - ◆ 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 的任选项如下：
 - ◆ 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，默认值分别为 CODE, DATA 和 BSS。

2) Linker (连接器)

本菜单设置有关连接的选择项，包括以下内容：

- Map file menu——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默认库。
- 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。

3) 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。

- 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——编译前是否清除消息窗口中的信息。
- 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 时则不存。
- 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备份文件（BAK 文件）。
- Tab size——设置制表键大小，默认为 8。