



高级C语言程序设计 技巧与实例

姚远 木水 等编

机械工业出版社

| |
|---|
| 2 |
| |

高级 C 语言程序设计 技巧与实例

姚 远 木 水 等编



机械工业出版社

(京) 新登字 054 号

本书系统地介绍了高级C语言程序设计技巧。对读者在编程时经常遇到的问题以专题的形式进行了详细的讨论,并且给出了大量的实例。本书共分12章,内容包括:C语言运行环境简介;结构化程序设计;测试与错误处理;标准C语言函数库;数据结构;C语言在排序和查找中的应用;绘制图形;动画游戏的设计;图象处理;用C语言解决数值分析问题;C语言接口;建立与鼠标器的接口。

本书叙述清晰,通俗易懂,使用方便,是使用C语言的用户极为有用的工具书,也是计算机应用人员和大中专院校师生必备的参考书。

图书在版编目 (CIP) 数据

高级C语言程序设计技巧与实例 / 姚远等编. —北京: 机械工业出版社, 1995

ISBN 7-111-04541-6

I. 高… II. 姚… III. C语言—程序设计 IV. TP312C

中国版本图书馆CIP数据核字(95)第03421号

出 版 人: 马九荣(北京市百万庄南街1号 邮政编码100037)

责任编辑: 王中玉 版式设计: 王 颖 责任校对: 肖新民

封面设计: 方 芬 责任印制: 路 琳

机械工业出版社印刷厂印刷·新华书店北京发行所发行

1995年8月第1版·1995年8月第1次印刷

787mm×1092mm $1/16$ ·26.25印张·643千字

0 001-4 000册

定价: 35.00元

JS378/16

前 言

C 语言是当今世界最流行的程序设计语言之一，目前 C 语言已用于各个方面的程序设计，无论是设计系统软件（操作系统，编译系统），还是应用软件（图形处理），数据处理（如企业管理）以及数值计算等都可以很方便地使用 C 语言。C 语言的应用所以如此广泛，是由于其本身所固有的特点，即吸收了汇编语言的精华，继承和发扬了其他高级语言的长处，如规模适中，语言简洁，占用存储空间少，可移植性好等等。因此使众多的程序设计者爱不释手。

本书的侧重点是理论与实践相结合，在简要介绍 C 语言基本程序设计知识的基础上，着重强调编程方法，并由浅入深地给出了大量的实例，从而使读者达到学以致用效果。本书取材新颖，内容丰富，基本上覆盖了 C 语言程序设计的各个方面。

本书以现在广泛使用的 Turbo C、Microsoft C、Quick C 为开发环境，所提供的实例虽然是针对上述三个开发环境的，但是读者只要做一些很少的修改就可以适用于您的开发环境。

参加本书编写的有姚远、木水、付晓铃、苏小芳、张亦奋、汪扬、王忠英等。

因作者才疏学浅，错误之处在所难免，敬请读者和同行批评指正。

作 者

1994 年 12 月

目 录

| | | | |
|------------------------------|----|-----------------------------|-----|
| 第 1 章 C 语言运行环境简介 | 1 | 4.10 字符串处理 <string.h> | 94 |
| 1.1 Turbo C 运行环境简介 | 1 | 4.11 日期与时间 <time.h> | 98 |
| 1.2 Quick C 运行环境简介 | 18 | 4.12 环境工具的限定 | 99 |
| 1.3 Microsoft C 运行环境简介 | 28 | 第 5 章 数据结构 | 103 |
| 第 2 章 结构化程序设计 | 35 | 5.1 引言 | 103 |
| 2.1 算法 | 35 | 5.2 自相关结构 | 103 |
| 2.2 伪代码 | 35 | 5.3 动态内存分配 | 103 |
| 2.3 控制结构 | 36 | 5.4 链表 | 104 |
| 2.4 If 选择结构 | 36 | 5.5 栈 | 111 |
| 2.5 If/Else 选择结构 | 36 | 5.6 队列 | 116 |
| 2.6 While 循环结构 | 39 | 5.7 树 | 121 |
| 2.7 算法制定 | 39 | 第 6 章 C 语言在排序和查找 | |
| 2.8 自顶向下逐步求精法 | 40 | 中的应用 | 127 |
| 2.9 结构化框图 | 44 | 6.1 引言 | 127 |
| 2.10 设计一个程序流程图 | 46 | 6.2 排序的基本概念 | 127 |
| 2.11 流程图法则 | 47 | 6.3 冒泡排序法 | 128 |
| 2.12 一个完整的实例 | 47 | 6.4 选择排序法 | 131 |
| 2.13 增量、减量以及赋值运算符 | 51 | 6.5 插入排序法 | 132 |
| 第 3 章 测试与错误处理 | 54 | 6.6 shell 排序法 | 133 |
| 3.1 错误的几种基本类型 | 54 | 6.7 Quicksort 排序法 | 135 |
| 3.2 如何测试程序 | 58 | 6.8 字符串数组的排序 | 137 |
| 3.3 个别模块测试的方法 | 67 | 6.9 结构的排序 | 138 |
| 3.4 错误处理方法及工具的使用 | 70 | 6.10 堆排序(HEAP SORT) | 139 |
| 第 4 章 标准 C 语言函数库 | 76 | 6.11 合并排序(MERGE SORT) | 142 |
| 4.1 错误 <errno.h> | 76 | 6.12 树排序(TREE SORT) | 144 |
| 4.2 一般定义 <stddef.h> | 76 | 6.13 查找 | 150 |
| 4.3 诊断 <assert.h> | 77 | 第 7 章 绘制图形 | 158 |
| 4.4 字符处理 <ctype.h> | 77 | 7.1 模式和调色板 | 158 |
| 4.5 数学 <math.h> | 78 | 7.2 写像素 | 160 |
| 4.6 非局部跳转 <setjmp.h> | 80 | 7.3 一个简单的检测程序 | 166 |
| 4.7 变量参数 <stdarg.h> | 81 | 7.4 存储和调用图象 | 171 |
| 4.8 输入/输出 <stdio.h> | 81 | 7.5 程序综合 | 183 |
| 4.9 一般实用程序 <stdlib.h> | 90 | 第 8 章 动画游戏的设计 | 204 |

| | | | |
|--|-----|---|-----|
| 8.1 “精灵”设计 | 204 | 10.10 求 $f(x) = x^3 - x^2 - 1.9$ 的根 | 317 |
| 8.2 动画场地 | 204 | 10.11 以 $x_{n+1} = (x_n + A/x_n) / 2$ | |
| 8.3 屏幕级的移动 | 205 | 来求 A 的平方根的近似值 | 318 |
| 8.4 “精灵”级的跑动 | 211 | 10.12 用辛卜生法求积分 | 320 |
| 8.5 动画数据的组织 | 213 | 10.13 用高斯法求多项式解 | 320 |
| 8.6 一个完整的动画程序 | 214 | 10.14 计算矩阵的相加、相减、相 | |
| 第 9 章 图象处理 | 237 | 乘、相除及其逆矩阵 | 323 |
| 9.1 概述 | 237 | 第 11 章 C 语言接口 | 329 |
| 9.2 图象的获取 | 240 | 11.1 C 语言与 UNIX 系统的接口 | 329 |
| 9.3 图象的存储 | 242 | 11.2 Turbo C 与 FORTRAN 语言 | |
| 9.4 图象的显示 | 245 | 的相互调用 | 335 |
| 9.5 图象的基本运算 | 248 | 11.3 C 语言与 FORTRAN 语言数 | |
| 9.6 灰度拉伸 | 262 | 据文件共享 | 340 |
| 9.7 图象的二值化 | 272 | 11.4 Turbo C 与 dBASE III 接口 | 341 |
| 9.8 边缘检测 | 276 | 11.5 Turbo C 与汇编语言程序接口 | 344 |
| 9.9 图象去噪 | 287 | 11.6 Turbo PROLOG 和 Turbo C | |
| 9.10 图象细化 | 294 | 的接口技术 | 350 |
| 9.11 小结 | 299 | 11.7 FORTRAN、PASCAL、C | |
| 第 10 章 用 C 语言解决数值分 | | 语言相互调用的实现 | 354 |
| 析问题 | 300 | 11.8 FoxBASE+ 与 Turbo C 的 | |
| 10.1 求 $3x+2y-7z=5(x, y, z < 20)$ | | 接口技术 | 358 |
| 的可能解 | 300 | 11.9 用 C 语言架起 ORACLE | |
| 10.2 牛顿法解 $x^2 - c = 0$ 的根 | 301 | 与 AutoCAD 的桥梁 | 363 |
| 10.3 用 SECOND 法解方程式 | | 11.10 使用 C 语言把 dBASE 数据 | |
| $x^2 - 56\sqrt{x} = 2.5$ | 303 | 转入 ORACLE 系统 | 364 |
| 10.4 解函数 $q(i, n, r, k)$ | 304 | 第 12 章 建立与鼠标器的接口 | 371 |
| 10.5 判断矩阵是否为特殊矩阵 | 307 | 12.1 鼠标的基本知识 | 371 |
| 10.6 求 $(x+a)^n$ 中各项的系数 | 310 | 12.2 虚实屏幕的对应 | 372 |
| 10.7 检验矢量 $(a_1, \dots, a_n), (b_1, \dots, b_n)$ | | 12.3 鼠标驱动程序 | 372 |
| 是否正交 | 313 | 12.4 高级鼠标函数 | 373 |
| 10.8 以递归方法求 10 阶以内的 | | 12.5 在绘图软件中应用鼠标输入 | 379 |
| 行列式值 | 314 | 12.6 附加说明 | 412 |
| 10.9 求 $ax^2 + bx + c = 0$ 的根 | 316 | 参考文献 | 413 |

第 1 章 C 语言运行环境简介

1.1 Turbo C 运行环境简介

自从 Borland 公司推出 Turbo C 后, 就得到读者的欢迎。为什么 Turbo C 会如此流行呢? 一是因为在 Turbo C 未公布之前, 在 PC 机上的 C 语言程序的编译器 (compiler) 或翻译器 (interpreter) 都无法满足读者的需求, 二是由于 Turbo PASCAL 的好用与快速给了读者良好的印象, 的确 Borland 公司出版的产品都是不错的, 值得去学习。

1.1.1 Turbo C 软盘内容

Turbo C 共有 4 片软盘, 它包括的文件及其功能描述如下:

软盘 1: 集成环境 (The Integrated Environment)

此软盘包含了集成环境版本的 Turbo C、求助文件及此套软件最新的事项和错误更正。

| | |
|------------|-------------------|
| TC.EXE | Turbo C 的集成环境 |
| TCHELP.TCH | Turbo C 求助文件 |
| README.COM | 最近更新的文件 |
| README | README.COM 应用文本文件 |

软盘 2: 命令行及工具 (Command - line & Utilities)

此软盘包含 Turbo C 的命令行编译程序及其他可能用到的工具程序。

| | |
|------------|---------------------|
| TCC.EXE | Turbo C 的命令行编译程序 |
| TLIND.EXE | Turbo C 的连接程序 |
| TCINST.COM | Turbo C 安装程序 |
| CPP.EXE | Turbo C 预处理程序 |
| MAKE.EXE | Turbo C 的 MAKE 实用程序 |
| TOUCH.EXE | Turbo C 文件重建日期实用程序 |

软盘 3: 文件及程序库 I (Include Files & Libs I)

此软盘包含浮点数程序库、头文件及给微小 (tiny)、小 (small)、大 (large) 存储器模式的执行程序库。

浮点数程序库包括:

| | |
|----------|----------------|
| ALLOC.H | 存储器管理函数 |
| ASSERT.H | 错误处理宏头文件 |
| BIOS.H | 头文件 |
| CONIO.H | DOS 所控制的 I / O |
| CTYPE.H | 字符分类宏 |
| DIR.H | 目录及路径名的结构、宏和函数 |
| ERRNO.H | 系统调用错误存储器 |

| | |
|--------------|------------------------|
| GCNTL.H | 文件控制信息文件 |
| FLOAT.H | 浮点数参数文件 |
| IO.H | 输入 / 输出结构及声明 |
| LIMITS.H | 环境参数文件 |
| MATH.H | 不同数学函数的声明 |
| MEM.H | 存储器管理函数 |
| PROCESS.H | 过程管理结构和声明 |
| SETJMP.H | Set jmp 及 Long jmp 的声明 |
| SHARE.H | 共享文件 |
| SIGNAL.H | 单一定义文件 |
| STDARG.H | 变量参数头文件 |
| STDDEF.H | 通用类型及宏文件 |
| STDIO.H | 标准输入 / 输出文件 |
| STDLIB.H | 标准子程序的声明 |
| STRING.H | 字符串管理子程序文件 |
| TIME.H | 时间及日期函数头文件 |
| SYS / STAT.H | 协助输入 / 输出头文件 |

微小 / 小 / 大存储器模式执行时实用程序

| | |
|-----------|------------------|
| COT.OBJ | 微小存储器模式启动程序目标码 |
| COS.OBJ | 小存储器模式启动程序目标码 |
| CS.LIB | 微小 / 小存储器模式库存子程序 |
| MATHS.LIB | 微小 / 小存储器模式数学子程序 |
| COL.OBJ | 大存储器模式启动程序目标码 |
| CL.LIB | 大存储器模式库存子程序 |
| MATHL.LIB | 大存储器模式数学子程序 |

软盘 4: 程序库 II

此软盘包含简洁 / 中 / 巨大存储器三种模式的程序库, 建立启动的文件和各种不同性质的其他文件。

简洁 / 中 / 巨大存储器模式的程序库

| | |
|-----------|----------------|
| COC.OBJ | 简洁存储器模式启动程序目标码 |
| CC.LIB | 简洁存储器模式库存子程序 |
| MATHC.LIB | 简洁存储器模式数学子程序 |
| COM.OBJ | 中存储器模式启动程序目标码 |
| CM.LIB | 中存储器模式库存子程序 |
| MATHM.LIB | 中存储器模式数学子程序 |
| COH.OBJ | 巨大存储器模式启动程序目标码 |
| CH.LIB | 巨大存储器模式库存子程序 |

| | |
|--------------|---------------------------------|
| MATHH.LIB | 巨大存储器模式数学子程序 |
| 建立启动的文件 | |
| CO.ASM | 启动子程序的原始文件 (CO?.OBJ) |
| BULES.ASI | 启动程序原始文件的包含文件 |
| BUILD-CO.BAT | 建立立即执行目标程序的批处理文件 |
| 各种不同性质的其他文件 | |
| CNVTCFG.EXE | Turbo.CFG 与 TCCONFIG.TC 之间的转换程序 |
| CPINIT.OBJ | 与 Turbo Prolog 连接的目标程序文件 |
| MATHERR.C | 交错的 Matherr 函数原始文件 |

1.1.2 安装及使用 Turbo C

以下说明安装和运行 Turbo C 的两个步骤：首先，在计算机上安装 Turbo C 系统；第二，使用基本步骤工作，包括编译和运行一个程序。为了能尽快地运行 Turbo C，重点在于说明建立一个基本的、通用的配置。系统开始工作以后可以加入另外的工具和选项。

1. 安装 Turbo C

将 Turbo C 一号盘插入驱动器 A，转到 A：提示符下，并键入命令：

```
Install
```

以后按照屏幕上的提示换盘插入软驱即可安装 Turbo C。

2. 命令行系统

使用传统的命令行方式开发程序过程如下：

(1) 建立源文件：使用最熟悉的正文编译器输入并保存两个源文件 `main.c` 和 `utility.c`。对于开发大型的 C 应用程序，最好使用多窗口、灵活的编译器，例如用 Brief，这样的编译器能很方便地在文件之间删除和复制代码。对于双软驱系统，应转到 B 提示符下，因为此处包含用户文件，对于硬盘系统，可以转入任一用户希望保存 C 文件的目录下，假定符合前面所述的路径和配置文件要求。

(2) 编译并连接程序：使用 `tcc.exe` 命令行编译程序编译和连接是很简单的。该程序不仅省去了单独的编译和连接步骤，而且自动地结合了适当的起始代码 (`cos.obj`) 和库例程 (`emu.lib`, `maths.lib` 和 `cs.lib`)。如果使用双软驱系统，前面创建的运行盘应在驱动器 A 中，用户应转入驱动器 B。对于硬盘系统，用户应继续转入存放这两个源文件的目录中。对任一种设置，在 DOS 提示符下键入下面的命令：

```
tcc main.c utility.c
```

编译程序将编译这两个 C 程序，然后自动地启动连接程序，传递以正确的命令行。结果文件是 `main.obj`, `utility.obj` 和 `main.exe`。现在键入：

```
main
```

注意每次运行 `tcc` 程序时，它都自动地重编译两个 C 文件，即使只修改了其中之一。

3. 集成开发环境

本节介绍 Turbo C 集成环境，给出开发实例程序的基本步骤，然后是学习和使用该系

统的一些基本要点。

要开发一个实例程序，有以下步骤：

(1) 启动集成环境：键入 tc 启动 Turbo C 的集成开发环境。

(2) 在编辑器中建立源文件：启动 Turbo C 后，按 Alt-E 进入编辑器。该编辑器所用的命令几乎等同于 Turbo PASCAL 和 Sidekick，类似于 Wordstar。首先键入源程序，输入完成后，按 Alt-F 进入文件菜单，然后按 S 保存该文件到盘上。系统询问是否替换缺省名 noname.c，此时键入 main.c 并按回车。

存入文件后，仍处于文件菜单。此时按 N 清掉编辑器准备输入一个新的文件。

在程序编译以前还有一个文件需要建立。如果此时调回 main.c 执行编译和连接，Turbo C 无法知道存在另一个文件与该程序相关，报告以下信息：

```
Undefined symbol'-RepeatStr' in module main.c
```

因此，用户必须建立一个工程 (project) 文件来通知 Turbo C 组成该程序的模块。工程文件类似于 make 文件。建立该文件很简单，只要两行就可组成：

```
main.c
utility.c
```

以文件名 main.prj 保存该文件。然后按 Alt-P，再按 P，接着键入 main 使 Turbo C 知道该工程文件名，注意如果整个程序仅由一个文件组成时不需要建立工程文件。

(3) 产生 .exe 文件：建立了工程文件后，可在任何时候编译和连接整个系统，通过简单地按 Alt-C 调出编译菜单，按 M (或按 F9) 产生 .exe 文件。此时生成可执行文件 main.exe。

注意使用了工程文件后，哪个文件当前处于编辑器中是无关紧要的，当要求 Turbo C 根据一个工程文件生成一个程序时，它自动地存取所有需要的文件，并输出 .obj 和 .exe 文件到磁盘上 (不同于 Turbo PASCAL、Turbo C 不编译到内存，而总是写 .obj 和 .exe 文件到磁盘上。)

Turbo C 的工程机制不仅自动地生成多文件的应用程序，而且产生最新的 .exe 文件。类似于 make 工具，它检查文件修改的日期，不重编译未修改的源代码文件。本书为所有的例子程序提供工程文件，并对该工具的另外一些特征作了说明。

(4) 运行程序：要运行程序，在系统中按 Alt-R。注意 Alt-R 命令还自动完成将可执行文件变成最新版本的编译和连接过程。因此不耐烦的用户可直接按 Alt-R，而不用前述命令生成 .exe 文件。

1.1.3 Turbo C 菜单选择项

在 Turbo C 下，有 6 种不同的存储器模式，可依需求来选择：

(1) Tiny (微小)：这是最小存储器模式，4 个段寄存器 (Segment Registers) 都在相同的位置上。总共有 64KB 的存储器可以给你的程序码、数据和数组使用。此 4 个段寄存器是程序码段 (Code Segment, CS)、数据段 (Data Segment, DS)、堆栈 (Stack Segment, SS) 和附加段 (Extra Segment, ES)。

(2) Small (小)：程序码各数据段不一样，也不重叠，所以可以有 64KB 的数据段。而

堆栈及附加段都在相同的段里。对一般用途而言,这已是相当好的。

(3) Medium (中): 数据段被限制在 64KB, 但程序码段不在此限制, 其可以大至 1MB, 对于程序太大而不需用太多数据者而言, 这是最好的模式。

(4) Compact (简洁): 与中存储器模式相反, 程序码段只有 64KB, 而数据段可大到 1MB。对数据多而程序短的情况, 则此为最佳的存储器模式。

(5) Large (大): 程序码段和数据段都有 1MB 的大小, 只有在非常大的应用程序下才使用到此模式。

(6) Huge (巨大): 一般来说 Turbo C 限制全部的静态数据只能到 64KB, 然而巨大存储器模式允许静态数据大于 64KB。

一般来说, 在小存储器模式下应用程序已相当够用了, 因此本书以小存储器模式为主将其复制到软盘上, 此后你要使用 Turbo C 只要用此编译软盘 (Compile disk) 及工作软盘 (Work disk) 即可。将下列 7 个文件复制到编译软盘:

| | | | |
|--------|----------|---------|-----------|
| TC.EXE | STDIO.H | COS.OBJ | CS.LIB |
| | STDARG.H | | MATHS.LIB |
| | | | EMU.LIB |

另外工作软盘必须含有系统 (即能够开机), 然后将一些 C 程序复制过来, 必要时亦可将求助文件 (TCHELP.TCH) 拷贝进来。

请用你的工作软盘开机 (或用与其 DOS 版本相同的系统也可以), 注意将工作软盘放入驱动器 A, 编译 Turbo C 软盘 (即含上述 7 个文件) 放入驱动器 B, 然后在 A>下键入 B:TC, 如下所示:

```
A > B:TC <enter>
```

即可进入 Turbo C 系统。

Turbo C 的菜单包括: 主菜单 (Main Menu), 编辑窗口 (Edit Window), 信息窗口 (Message Window) 和快速参考行 (Quick Reference Line) 4 个部分, 如图 1-1 所示。

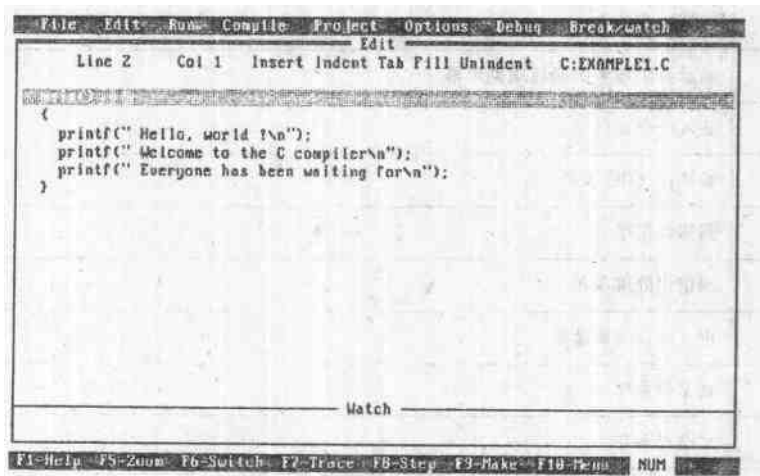


图 1-1

在主菜单中有 7 个选项让你选择: File, Edit, Run, Compile, Project, Options, Debug。只要利用左右箭头或按高亮度的大写字母, 如 F, E, R, C, P, O, D 去选择一个菜单项, 然后按 Enter 键。也可以利用 Home 键跳到第一个选择项, 同样地, 按“End”键可以跳到屏幕最后一个选择菜单项。

如你要求助按 F1 即可, 看完后要按 Esc 回到原先的菜单, 按 F6 可以切换窗口, 可从编辑窗口变换到信息窗口, 反之也可以, 按 F5 可使窗口变大或变小, 而按 F10 可以回到主菜单。

如果不是在菜单上, 而你只要按 Alt 加上 F, E, R, C, P, O, D 也可以在主菜单的 7 项中进行选择, 例如按 Alt-E 则在编辑的窗口下, Alt-F 则在文件菜单下。请注意, 如果要退出 Turbo C 系统回到 DOS 时, 必须进入文件菜单 (File Menu), 然后选择 Quit 命令退出。当你退出 Turbo C 系统时, 你的工作文件没有存起来, 此时 Turbo C 会询问你是否要将此文件存起来。但你也可以按 Alt-X 退出 Turbo C 系统, 此时与上述先选择菜单中的 File; 再选择 Quit 命令一样。

下面是一些 Turbo C 的常用键:

| 键 值 | 功 能 |
|---------|---------------------|
| F1 | 显现求助窗口根据你现在的位置的相关信息 |
| F2 | 将正在编辑的文件存入软盘 |
| F3 | 装入一个文件 |
| F5 | 改变窗口大小 |
| F6 | 变换窗口 |
| F7 | 到前一个错误位置 |
| F8 | 到下一个错误位置 |
| F9 | 执行 Make 实用程序 |
| F10 | 回到主菜单中 |
| Alt-F1 | 显示你所参考的最后求助屏幕 |
| Alt-F3 | 装入一个文件 |
| Alt-F10 | 编译成 .OBJ 文件 |
| Alt-C | 到编译菜单 |
| Alt-D | 到错误处理菜单 |
| Alt-E | 进入文件编辑菜单 |
| Alt-F | 到文件菜单 |
| Alt-O | 至选择菜单 |
| Alt-P | 到专题菜单 |

(续)

| 键 值 | 功 能 |
|-------|--------------------|
| Alt-R | 执行你的程序 |
| Alt-X | 退出 Turbo C, 回到 DOS |

1. File 选项

如果现在不在主菜单上, 按 F10 回到主菜单, 然后选择 F 按 Enter 键此时会出现如图 1-2 的画面:

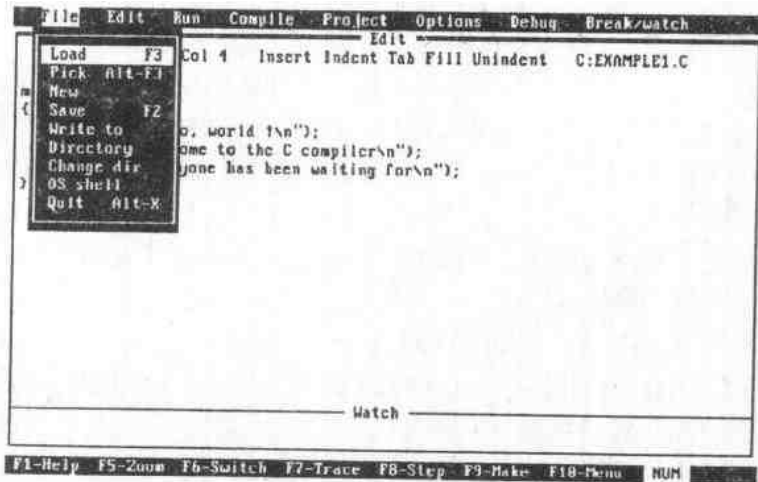


图 1-2

(1) Load: 装入一个文件(亦可按 F3 键), 选择此项后, 系统会问你 load 的文件名称, 此时可用通配符 (Wild Card *) 得到选择文件的打印, 或者直接指定一个文件名称。

(2) Pick: 从先前装入的 8 个文件显示中选取一个文件 (亦可以按 Alt-F3 键), 被选的文件会装入编辑器中, 如果你从 Pick 的显示中选取“-load file-”则与 File / Load 或 F3 键完全一样。

(3) New: 设置新文件, 内定的新文件名是 NONAME.C。

(4) Save: 将编辑器中的文件存入磁盘 (亦可按 F2 键), 若文件名是 NONAME.C, 则系统会问你是否要改名字再存入。

(5) Write to: 将文件写入一新文件中或覆盖掉已存在的文件。

(6) Directory: 显示你所需要的目录及文件集合

(7) Change dir: 改变目录。

(8) DOS shell: 暂时退出 Turbo C 系统, 回到 DOS 系统(这就是为什么工作软盘要有开机文件的原因, 否则你必须更换一片有系统的软盘才能回到 A >)。当你要再进入 Turbo C 时, 则只要进入 exit 命令即可。

(9) Quit: 退出 Turbo C, 回到 DOS 下 (亦可按 Alt-X)。

2. Edit 选项

当你在主菜单 (Main Menu) 中选择 E 这个选择项后再按“Enter”键, 则此时会进入编辑窗口。无论你在系统的任何地方, 甚至在信息窗口 (Message Window), 你都可以进入编辑窗口, 只要按 F10 键让其进入菜单, 接着按 E, 或者直接按 Alt-E 亦可。当你在编辑窗口时, 在屏幕的顶端提供你已在编辑中文件的一些信息:

| Line | Col | Insert | Indent | Tab | C: FILENAME .EXT |
|------------------|-----|--------|--------|-----|---|
| Line | | | | | 光标停在文件的那一行。 |
| Col | | | | | 光标停在文件的那一列。 |
| Insert | | | | | 表示启用插入模式, 而开始与停止的开关是 Ctrl-V 或 Insert 键。 |
| Indent | | | | | 自动缩排启用, 其启用与停止的开关是 Ctrl-OI。 |
| Tab | | | | | 表示 Tab 键可以使用, 其启用与停止的开关是 Ctrl-OT。 |
| C: FILENAME .EXT | | | | | 表示在驱动器 C 中所编辑的文件名和扩展名。 |

一般常用的编辑命令如下:

- (1) 利用 Up / Down, Left / Right, PgUp / PgDn 键来移动光标。
- (2) 利用 Ctrl-Y 键删除某一行。
- (3) 利用 Ctrl-T 键删除一个字 (Word)。
- (4) 利用 Ctrl-KB 键(开始), Ctrl-KK (结束) 来对某一个块做标记。
- (5) 利用 Ctrl-KV 键移动某一个块。
- (6) 利用 Ctrl-KC 键拷贝某一个块。
- (7) 利用 Ctrl-KY 键删除某一个块。

当你进入编辑窗口时, Turbo C 会自动命名一个名叫 NONAME.C 的文件。在编辑之前, 无论是新建或编辑一个文件, 若不需要信息窗口, 你可以使用 F5 键将编辑窗口扩大为全屏幕, 再按一次 F5 键, 则回到先前的分区屏幕状态。

3. Run 选项

执行命令会编译 (Compile)、连接 (Link) 刚刚装入 (Load) 的文件, 产生一个可执行文件 (.EXE 文件) 并立即执行。

4. Compile

当你在主菜单中按 C 时, 则进入编译菜单, 在编译菜单中有 5 种功能, 如图 1-3 所示。

- (1) Compile to OBJ: 将原始文件编译成目标文件 (.OBJ) (若选择此项必须再 Link 比较麻烦, 通常单一程序都选用 Make EXE file 或 Run)。
- (2) Make EXE file: 编译成可执行文件 (.EXE)。
- (3) Link EXE file: 连接目标文件 (.OBJ) 与可执行文件 (.EXE)。
- (4) Build all: 重新建立所有在项目 (Project) 中的文件, 不管它们是否已过时, 这个选择功能与 Make 相类似, 所不同的是 Build all 是无条件的, 而“Make”只重建非现在的文

件。

(5) Primary C file: 当你使用 Compile to OBJ 时, 利用这个选择项去指定要被编译成 OBJ 的 C 文件。

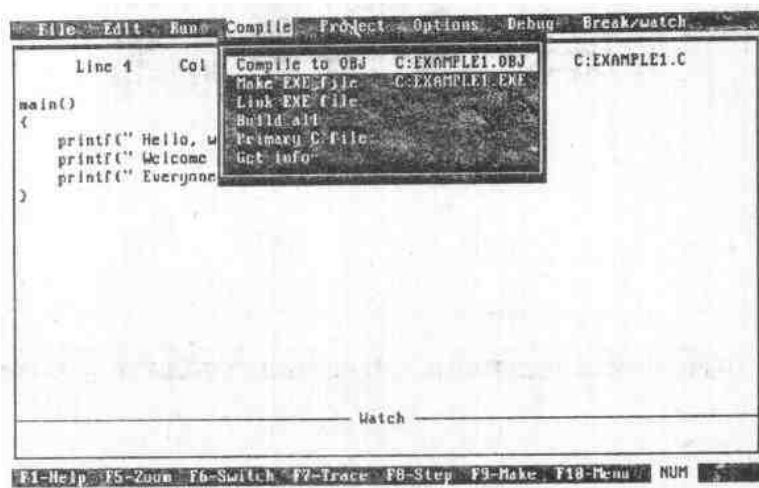


图 1-3

5. Project 选项

此功能允许你组合多个的 .OBJ 文件成为一个完整的程序 (即具有子程序类型)。Project 有三个选择项, 如图 1-4 所示。



图 1-4

(1) Project name: 选一项目名称, 当新建一个项目时, 项目的名称被给定为 .EXE 或 .MAP。

(2) Break make on: 当编译一个文件后, 若有警告 (Warnings)、错误 (Errors)、重大错误 (Fatal errors) 出现时是否应使用 Make 操作停止, 如图 1-5 所示。

(3) Clear project: 删除项目名称并且重设信息窗口。

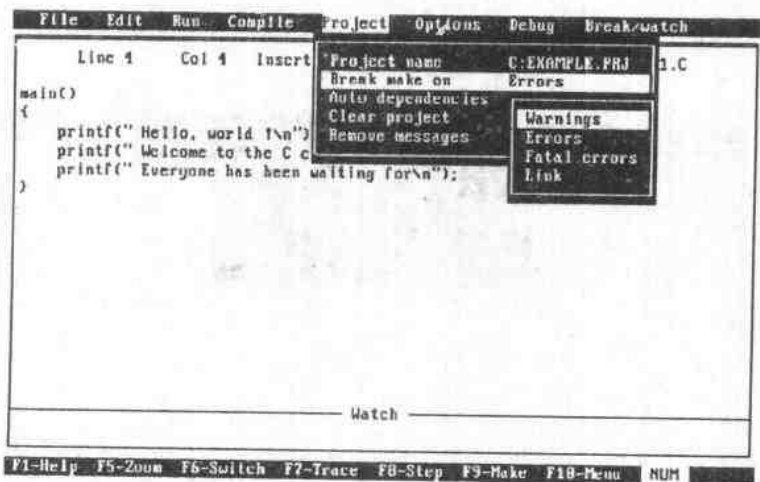


图 1-5

6. Options 选项

选择菜单(见图 1-6)包含了决定集成环境如何工作的一些假定, 这些假定包括编译器和连接器的选择功能, 程序库与包含文件的目录及程序执行时的一些命令行参数。



图 1-6

(1) Compile: 在 Compile 中的选择项又有 7 个小项(见图 1-7), 简述如下:

1) Model: 在前面我们曾谈及 Turbo C 有 6 种存储器模式可选择, 分别为 Tiny、Small、Medium、Compact、Large 及 Huge, 如图 1-8 所示, 依需求情况选择一适当的存储器模式。

2) Defines: 定义宏, 此功能可以将宏传递给预处理程序。如有一定义的宏如下:

```
BETA__BEST; one = 1; compiler = Turbo C
```

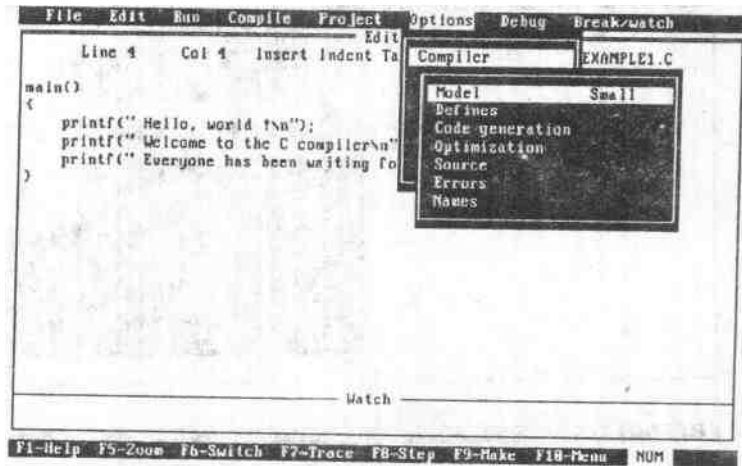



图 1-7

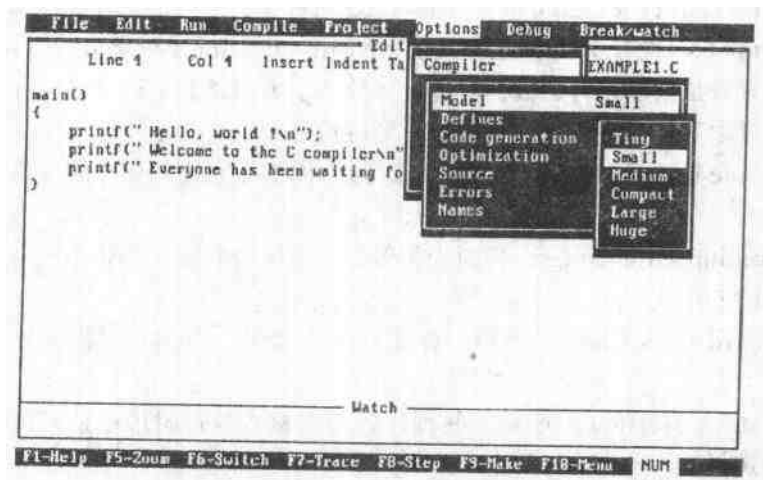


图 1-8

它表示定义符号 BETA__BEST, 并设 one 为 1, 及 compiler 等于字符串“Turbo C”。

3) Code generation: 告诉编译器以不同的方法准备目标码, 在此功能下又有 11 个小项如图 1-9。

① Calling convention: 使 Turbo C 对函数的调用产生一个 C 的调用序列或 PASCAL 调用序列。你可有两种选择: C 或 PASCAL。

② Instruction set: 允许你指定一个不一样目的的 CPU, 此为两个不同指令 (8088 / 80286) 之间转换。缺省是 8088 / 8086 程序码。

③ Floating point: 有 3 种选择功能: A 为 8087 / 80287, 这会产生直接的 8087 上程序码; B 为 None, 会使 Turbo C 假设都不会用及浮点运算; C 为 Emulation, 这会使 Turbo C 自动检查看看你是否 8087, 若有则用它, 若无则会精确模拟 8087, 但速度较慢。