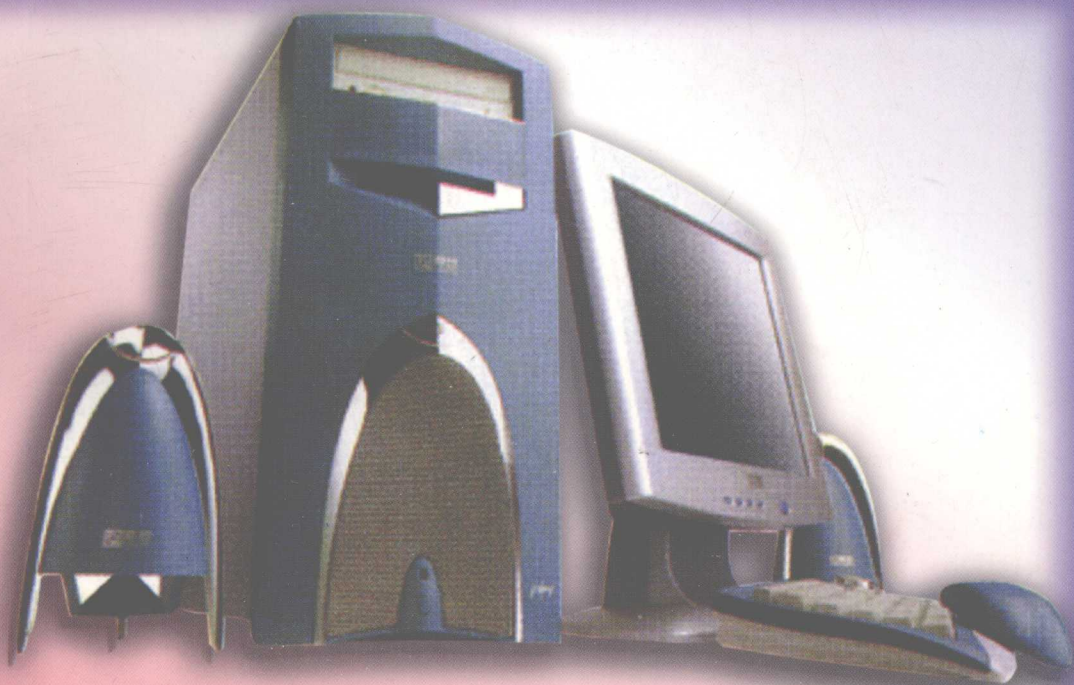


全国高等农林院校基础类课程教材

C语言程序设计 习题及实验指导

王忠芝 主编



中国林业出版社

52-5
全国高等农林院校基础类课程教材

C 语言程序设计 习题及实验指导

王忠芝 主编

中国林业出版社

图书在版编目 (CIP) 数据

C 语言程序设计习题及实验指导 / 王忠芝主编. —北京: 中国林业出版社, 2001.8

全国高等农林院校基础类课程教材

ISBN 7-5038-2853-6

I .C... II.王... 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 (2001) 第 053119 号

出版: 中国林业出版社 (100009 北京西城区刘海胡同 7 号)

E-mail: cfphz@public.bta.net.cn 电话: 66184477

发行: 中国林业出版社

印刷: 北京林业大学印刷厂

版次: 2001 年 8 月第 1 版

印次: 2001 年 8 月第 1 次

开本: 787mm×1092mm 1/16

印张: 13

字数: 316 千字

印数: 1~12000 册

定价: 19.00 元

《C 语言程序设计习题及实验指导》

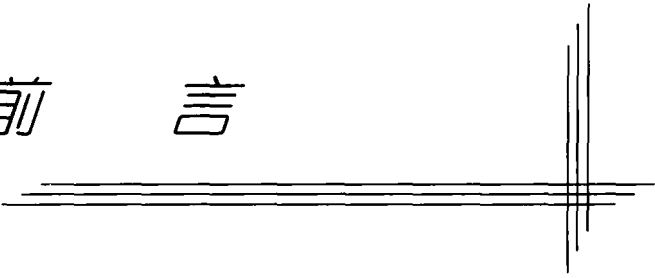
编写组

主 编 王忠芝

副主编 叶海建

编 者 蔡 娟 王春玲 李光辉 王久丽 陈志泊
徐秋红 李冬梅 王海兰 姚建成

前言



本书是供本专科生学习“C 语言程序设计”课程配套的习题和实验指导书。

本书的主要特点是注重学生的理解能力和一定的编程能力，并学习具体的操作内容，解决一些实际的问题，有很强的实用价值。

本书有大量的习题供学生复习和学习。

本书的内容适合于实验学时为 20~40 之间的 C 语言程序设计课程（总学时在 40~80 之间）。

本书内容丰富，概念清晰，实用性强，有大量的习题供学生复习和学习之用。对上机实验的题目做了题目分析，给出了参考程序。对实验的环境除了对 TC 做了较详细的介绍外，还对 VC 环境也做了介绍，供学生选用。另外给出了 C++ 的简单例子，便于学生进一步地学习面向对象的编程。

本书由北京林业大学王忠芝担任主编并组织编写，由中国农业大学叶海建担任副主编。其他编写人员为蔡娟、王春玲、李光辉。参加本书部分工作的还有：王久丽、陈志泊、徐秋红、李冬梅、王海兰、姚建成、钟亚华、余扬、孙玉、乔用、秦桂文、张朋。

在本书的编写过程中，得到了北京林业大学教务处和中国林业出版社的支持，在此一并表示感谢！

编 者

2001 年 5 月

目 录

第 1 章 Turbo C 集成编译环境	(1)
1.1 Turbo C 简介	(1)
1.2 Turbo C 程序的执行实例	(16)
1.3 用 Windows 的“记事本”编辑 C 程序	(28)
1.4 常见错误分析	(30)
第 2 章 C 语言实验	(34)
实验 1 C 程序的运行环境和运行一个 C 程序的方法	(34)
实验 2 数据类型、运算符和表达式	(37)
实验 3 最简单的 C 程序设计	(43)
实验 4 逻辑运算和判断选取控制	(47)
实验 5 循环控制	(54)
实验 6 数 组	(60)
实验 7 函 数	(68)
实验 8 编译预处理	(75)
实验 9 指 针	(75)
实验 10 结构体和共用体	(82)
实验 11 位运算	(89)
实验 12 文 件	(90)
第 3 章 习题及其参考答案	(92)
习题 1	(92)
习题 2	(94)
习题 3	(100)
习题 4	(105)
习题 5	(111)
习题 6	(119)
习题 7	(126)

习题 8	(129)
习题 9	(133)
第 4 章 Visual C++6.0 开发环境	(144)
4.1 概 述	(144)
4.2 Visual C++6.0 集成环境	(146)
4.3 Visual C++6.0 的工具栏	(171)
4.4 项目工作区窗口	(173)
4.5 代码编辑窗口	(176)
4.6 一个简单的 Windows 应用程序	(176)
第 5 章 VC++环境应用举例	(182)
5.1 VC++的启动	(182)
5.2 在 VC++中执行一个 C 程序	(183)
5.3 在 VC++中执行一个 C++程序	(189)
参考文献	(197)

第 1 章

Turbo C 集成编译环境

1.1 Turbo C 简介

1.1.1 程序开发过程

(1) 分 析

一般说来, 一个具体的问题要涉及许许多多的方面, 这是问题的复杂性所在。为了便于求解, 往往要忽略一些次要方面。这种通过忽略次要方面, 而找出解题规律, 就称为建立模型。分析问题就是要搞清对问题相关领域的了解和所具备的知识。

(2) 设 计

设计分为两步: 总体设计和详细设计。

① 总体设计: 当问题较大时, 常常要把问题分为一些子问题, 当子问题还大时, 还应进一步分割。这样, 一个较大的问题就被分成一些容易求解的子问题, 每一个子问题将作为程序的一个模块。这是总设计的一个任务。但是, 仅仅分解还是不够的, 在总体设计时, 还要考虑如何组织程序模块。例如, 把一个“教学管理系统”分为“学生管理”模块、“统计报表”模块、“教学计划”模块、“教室管理”模块、“教师基本信息管理”模块。“学生管理”模块又可以分解为“学籍管理”模块、“成绩管理”模块等等。

② 详细设计: 详细设计就是设计每个模块的数据结构和算法。

(3) 编码以及编辑、编译和链接

程序编码就是用具体的程序设计语言表现各模块的算法和数据结构。这一阶段的关键是运用某一种具体的计算机语言, 来描述前面设计的数据结构及其算法。为此就要掌握一种计算机程序设计语言。

程序编码可以先写在纸上, 也可以直接在计算机上书写, 形成一个程序文件。在一定的环境下进行程序的书写以及修改的过程称为程序的编辑。

写出一个高级语言程序后, 并不是就可以立即拿来执行。要让机器直接执行, 还要将它翻译成由机器可以直接认识并执行的机器语言程序。为区别它们, 把用高级语言写的程序(文件)称为源程序(文件), 把机器可以直接辨认并执行的程序(文件)称为可执

行程序（文件）。

这一过程一般分为两步：

第 1 步，在程序编辑过程中输入到计算机内部的是 ASCII 码。ASCII 码是计算机不能直接辨认的，为此，首先要将源程序文件翻译成机器可以辨认的程序文件——目标程序文件。这一过程称为翻译。在翻译过程中，还要对源程序中的语法和逻辑结构进行检查。翻译任务是由称做编译器的软件完成的。目标程序文件还不能被执行，它们只是一些可重定位的目标程序模块。

第 2 步，将目标程序文件中可重定位的目标程序模块以及程序所需的系统中固有的目标程序模块链接成一个完整的程序。经正确链接所生成的文件才是可执行文件。完成链接过程的软件称为链接器。

C 语言程序的编辑、编译和链接过程如下：

键盘输入 → 编辑 → 生成源程序文件 (.C) → 编译 → 生成目标程序文件 (.obj) → 链接 → 生成可执行文件 (.exe) → 执行 → 结果

目前，程序的编辑、编译、链接以及运行，都可以在一些集成环境下进行。我们编程所使用的 Turbo C 就是一种集成环境。

（4）程序测试

每一个人编写出一个程序后，在正式交付使用前，总要对程序进行测试。测试的目的是发现程序中的错误和验证程序的正确执行过程。

测试是以程序通过编译，没有语法和链接上的错误为前提的。在此基础上，通过让程序试运行一组数据，来验证程序的逻辑以及程序中的各语法有无错误。测试与调试是交替进行的，通过测试发现程序中的错误，通过调试进一步找出错误的位置并改正错误，这个过程一般需要重复多次，直到程序没有发现错误为止。

1.1.2 程序开发环境——Turbo C

C 语言的程序需要具体的 C 编译器和链接器的支持。不同的 C 编译器和链接器具有不同的特点和功能，形成了不同的 C 语言的版本。Turbo C 是 Borland 公司开发的一种 C 语言程序开发环境，它不仅提供了基于 ANSI C 的编译器和链接器，还提供了一个供程序员修改程序的方便的编辑器，形成一个集成化的 C 语言程序开发和运行环境。本书所介绍的开发环境是基于 Turbo C 2.0 的。它对机器的要求较低，可以适合更多的学习者使用。

Turbo C 定义了两种屏幕状态：程序开发状态和程序运行状态。两个状态是相互独立的，通常，Turbo C 位于开发环境状态，只有程序运行时才进入运行状态（也称用户屏幕）。

（1）在安装有 Turbo C 的计算机中启动 Turbo C

在安装有 Turbo C 的计算机中启动 Turbo C，有三种方法如下：

- ① 在 DOS 平台上，进入含有 Turbo C 的子目录后，输入命令 TC；
- ② 在 Windows 平台上，打开含有 Turbo C 的文件夹后，双击可执行文件 TC；
- ③ 在桌面上为 TC 建立一个快捷方式，直接双击该快捷方式，即可进入 Turbo C 2.0 集成开发环境。Turbo C 的界面如图 1.1 所示。

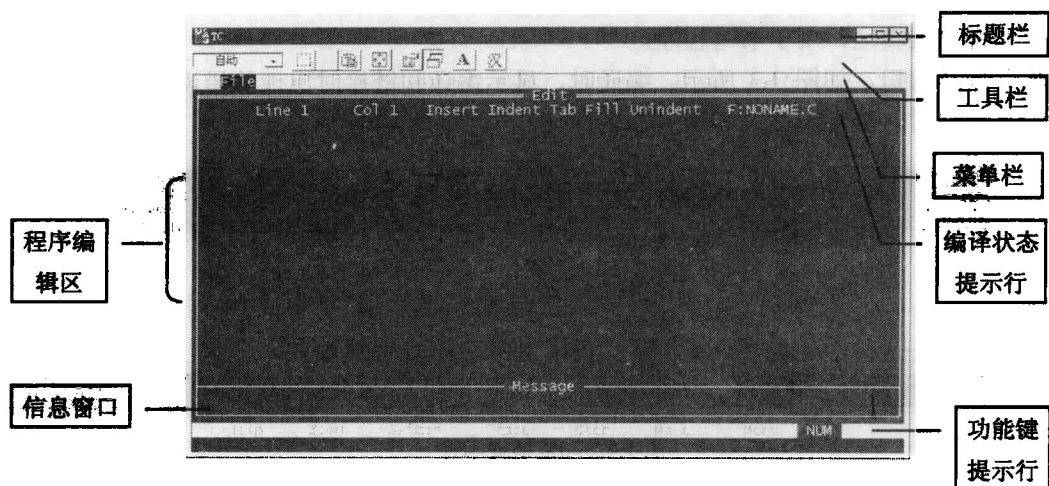


图 1.1 TC 界面

(2) Turbo C 的界面窗口介绍 (Windows 系统下)

① 标题栏：位于窗口的第一行，显示当前程序名 TC。标题栏左端的图标一方面表明它是一个 Turbo C 集成环境系统，另一方面点击它可以下拉一个控制菜单，用于 Turbo C 集成环境系统的关闭及控制窗口大小。这些控制功能也可用标题栏右端的三个按钮来完成。

② 工具栏：位于窗口的第二行，从左到右分别为“字体”、“标记”、“复制”、“粘贴”、“全屏幕”、“属性”、“后台”、“字体”、“汉字”。

③ 菜单栏：位于窗口的第三行，共有 8 个菜单：

File (Alt+F)	进行文件和目录操作
Edit (Alt+E)	编辑当前编辑窗口中显示的程序
Run (Alt+R)	控制程序的运行状态
Compile (Alt+C)	进行程序的编译、链接
Project (Alt+P)	对工程文件（由多个 C 文件组成的程序）进行管理
Options (Alt+O)	设置选项
Debug (Alt+D)	调试程序：显示变量值，查找函数，查看调用栈状态
Break/Watch(Alt+B)	调试程序：设置/清除断点，监测变量值变化

④ 程序编辑窗口，由三部分组成，即：编辑状态提示行、程序编辑区、信息窗口。

- 编辑状态提示行，如图 1.2 所示。“Line ”表示光标所在的行，“Col ”表示光标所在的列，“Insert ”表示处于插入状态，可以用键盘的 Insert 键进行设置和取消。“Indent”表示程序的行缩进方式，“F:\NONAME.C”表示盘符和文件名，表示现在编辑的文件是“F”盘“NONAME.C”文件。刚打开 Turbo C 时，系统默认“NONAME.C”文件名，自己一般在保存时重新给文件命名。

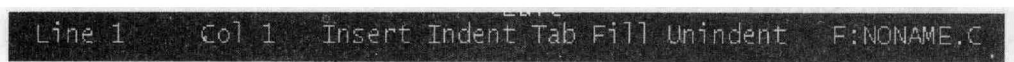


图 1.2 编辑状态提示行

- 程序编辑区，用于书写程序。
- 信息窗口，如图 1.3 所示。编译时，显示编译和链接时出现的错误和警告信息；调试时，显示表达式和变量的值。

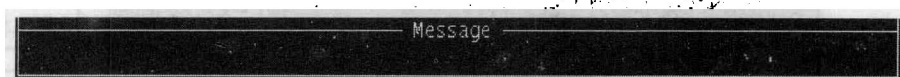


图 1.3 信息窗口

- ⑤ 功能键提示行，如图 1.4 所示。说明最常用功能键的含义。



图 1.4 功能键提示行

Turbo C 是基于键盘操作的，各种命令的执行和菜单的选择，都要使用箭头键和热键。

1.1.3 程序编译和链接操作

在“Compile”菜单中列出了编译和链接的操作。

. Compile to obj	对源程序进行编译生成目标文件 .OBJ
Make EXE file	对源程序进行编译和链接生成可执行 .EXE 文件
Link EXE file	将当前的.OBJ 文件和库进行链接生成可执行 .EXE 文件
Build all	重新编译链接 Project 中的全部程序，生成 .EXE 文件
Primary C file	指定文件作为编译对象，以替代编辑窗口中的文件
Get info	在弹出的窗口中显示有关当前文件的信息

在程序的编译、链接或执行中，遇到错误，会发出出错信息。

1.1.4 重要编辑热键

(1) 删除

删除光标所在行：Ctrl+Y

删除至行尾：Ctrl+QY

删除光标右侧一个单词：Ctrl+T

说明：Ctrl+Y 表示两个键同时按下；Ctrl+QY 表示按下 Ctrl 键的同时，先按下 Q 键，再按下 Y 键。以下的操作方法相同，不另作说明。

(2) 块操作

标记块：块首——Ctrl+KB，块尾——Ctrl+KK

标记单词：Ctrl+KT

块拷贝：Ctrl+KC

块删除：Ctrl+KY

(3) 其他

取消当前操作：Ctrl+U

自动缩进模式开关: Ctrl+QI
恢复重写错误信息: Ctrl+QW

1.1.5 Turbo C 菜单功能

特别要指出的是,除了 Edit 项外,每一个菜单项又对应一个子菜单。而选择 Edit 项目后,只是进入编辑器。

为了从主菜单中选择所需要的功能,可以用以下两种方式之一:

① 按 F10 键后,可以看到屏幕上部主菜单中的某个条目处出现亮块,此时,利用左右光标移动键(←与→)将此亮块移到所要选择的条目位置处,然后按回车键,即出现相应的子菜单。

② 直接按 Alt+主菜单条目中的首字母(分别为 F, E, R, C, P, O, D, B)此时就会出现相应的子菜单。例如,按 Alt+F 表示选择文件子菜单(File)。

当出现子菜单时,其中某个条目是高亮度的,此时可以利用上、下光标移动键(↑与↓)来移动该高亮度线,从而选择所需要的功能。

在主菜单或通过主菜单调用的任意一个子菜单中,按 Esc 键后将直接返回到活动窗口。下面简要介绍各子菜单的功能:

(1) 文件菜单(File)

当选中 File 菜单后,将出现一个子窗口,如图 1.5 所示。在这个子窗口中,有的条目右边还标出了实现该功能的热键。

所谓热键,是指为执行菜单中某一固定功能而设置的键。通过热键来实现某种功能,一般要比通过菜单选择更简单直接,但要求用户熟记这些键。例如,为了选择“文件”菜单的“Load”子菜单,可以直接按下 F3 键。

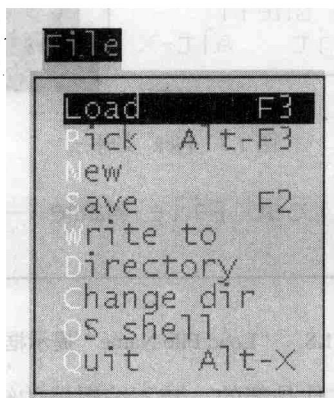


图 1.5 文件子菜单

各子菜单的功能如下:

① Load (加载): 把一个文件装入内存,即打开一个文件。当给定的文件名中有文件名通配符(*或?)时,将进行列表选择,如图 1.6 所示。

用键盘的移动键选择文件,按下回车键即可装入文件。在子目录处按下回车键可以进入子目录进行选择文件。

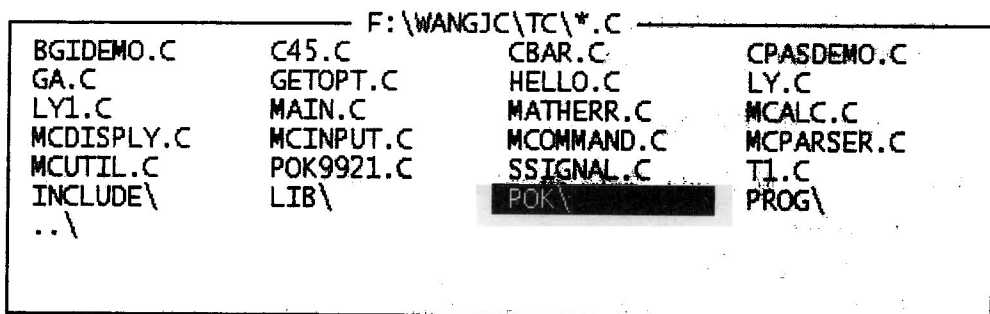


图 1.6 Load 文件

② Pick (选择): 将最近装入编辑窗口的 8 个文件列成表, 供用户选择, 选择后又装入编辑器, 光标放置在上次修改过的地方。如图 1.7 所示。若选了 “--Load File --”, 屏幕将出现 “Load File Name” 提示框。如图 1.8 所示。若给定的文件名中有文件名通配符 (* 或 ?) 时, 将进行列表选择, 如图 1.6 所示。其余步骤与 Load 操作相同。

若给定一个文件名, 就打开该文件 (文件存在) 或建立该文件 (文件不存在)。

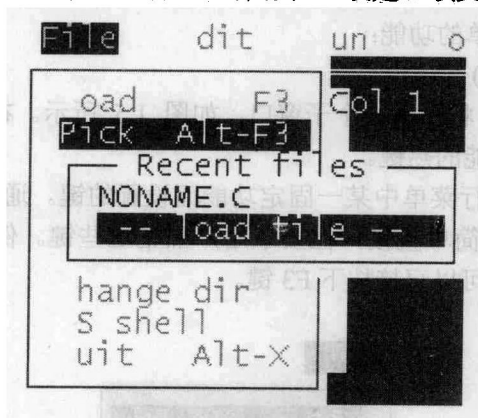


图 1.7 Pick 文件

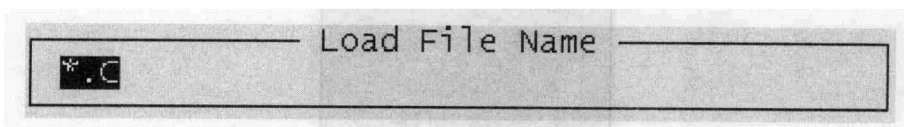


图 1.8 “Load File Name” 提示框

③ New (新文件): 说明文件是新的, 装入编辑器的缺省文件名为 “NONAME.C”。

④ Save (存盘): 将编辑器中的内容存盘。若文件名为 “NONAME.C”, 而又要存盘, 编辑器会询问是否要改名。

⑤ Write to (存盘): 把编辑器中的内容写入指定的文件中。若该文件已经存在, 则导致重写, 会覆盖原来的文件。

⑥ Directory (目录): 显示目录与所需文件列表 (若按回车键则选择当前目录)。

⑦ Change Dir (改变驱动器): 显示当前目录, 改变驱动器与目录。

⑧ OS Shell (暂时退出): 暂时退出 Turbo C, 转到 DOS 状态。在 DOS 状态下用 EXIT 命令又可返回 Turbo C。此功能对于在想运行 DOS 命令但又不想退出 Turbo C 时非常有用。

⑨ Quit (退出): 退出 Turbo C。

(2) 编辑菜单 (Edit)

调用内部编辑器, 可以输入修改程序。在编辑器中按 F10 键可返回主菜单 (或用 Alt 加所需主菜单命令的首字母), 但此时编辑器的内容仍保持在屏幕上。

在主菜单中按 Esc 键 或 E 键, 即可回到编辑器 (按 Alt+E 也可以, 且在任何时候都起作用)。

(3) 运行菜单 (Run)

当选中 Run 菜单后, 将出现一个子窗口, 如图 1.9 所示。在这个子窗口中, 有的条目右边标出了实现该功能的热键。

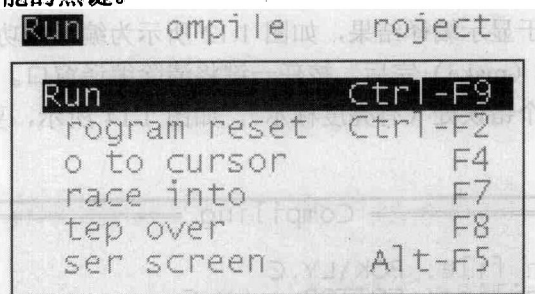


图 1.9 Run 菜单

① Run (运行): 运行当前程序。

② Program reset (程序重启动): 中止当前调试, 释放分配给程序的空间, 关闭已打开的文件。

③ Go to cursor (执行到光标处): 使程序从执行长条运行到编辑窗口中的光标所在行。若光标所在行不含可执行代码语句, 则显示一个 Esc 框作警告, 此时按 Esc 键, 如图 1.10 所示。



图 1.10 Esc 框

④ Trace into (跟踪进入): 运行当前函数中的下一个语句。若此语句不含调试器可访问的函数调用, 则停在下一条可执行语句上; 但若此语句含有调试器可访问的函数调用, 则停在函数定义的开始。

⑤ Step over (单步执行): 执行当前函数的下一语句, 即使遇到调试语句可访问的函数调用也不会跟踪进入下一级函数中。

⑥ User screen (用户屏幕): 切换到用户屏幕。

(4) 编译菜单 (Compile)

当选中 Compile 菜单后, 将出现一个子窗口, 如图 1.11 所示。

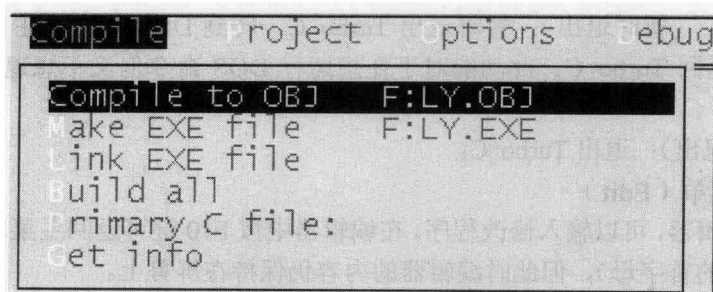


图 1.11 Compile 菜单

① **Compile to OBJ** (编译生成目标码): 本命令的热键为 **Alt+F9**。

本命令将一个 .C 源文件编译成 .OBJ 文件, 同时显示生成的文件名。 .OBJ 文件由源文件名产生; 或在没有指定文件名时, 由上次装入编辑器的文件名产生。 Turbo C 在编译时弹出一个窗口, 用于显示编译结果, 如图 1.12 所示为编译成功, 如图 1.13 为源程序出现错误。在编译/组装 (**make**) 完后, 按任一键将清除编译窗口。此时, 若发现有错误, 则转到消息窗口的第一个错误处 (有亮度标示), 如图 1.14 所示, 要把光标从消息窗口转到编辑窗口按 **F6** 键。

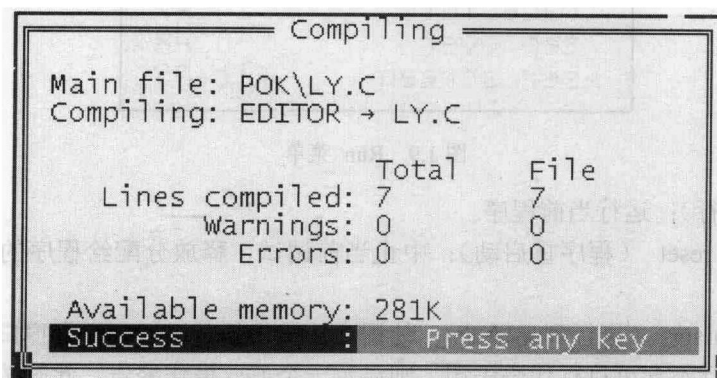


图 1.12 编译成功

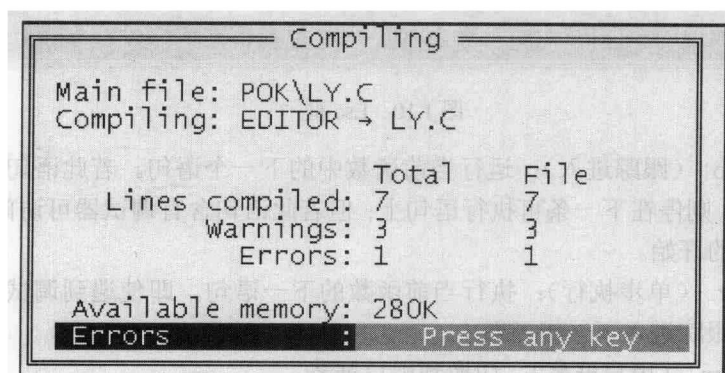


图 1.13 源程序出现错误

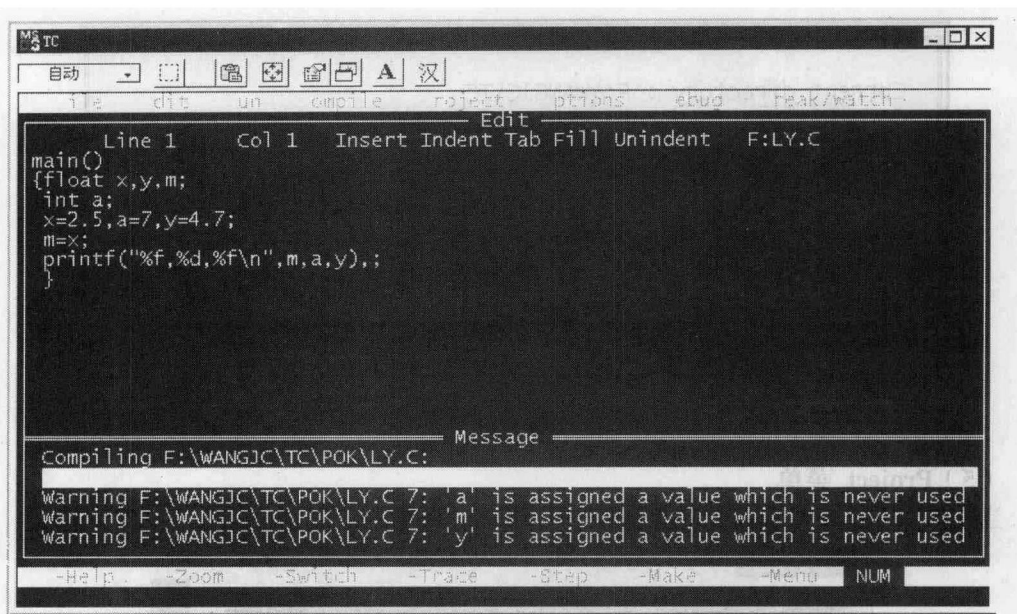


图 1.14 消息窗口

② **Make EXE file**（生成执行文件）：本命令的热键为 F9。

本命令调用 **Project/Make** 来生成 .EXE 文件，并显示所生成的 .EXE 文件名。.EXE 文件名是依次由下列文件名产生的：

- Project/Project Name 说明的项目文件名；
- 或 Project C File 说明的文件名；
- 或 上次装入窗口的文件名。

③ **Link EXE file**（链接执行文件）：把当前.OBJ 文件与库文件（既可以是缺省的也可以是定义在当前项目 Project 文件中的）链接在一起，生成.EXE 文件。

④ **Build All**（建立所有文件）：重建项目中的所有文件。本命令类似 **Compile/Make File**，只是它是无条件执行的，而 **Compile/Make File** 只重建那些非过时的文件。本命令首先将所有的 Project 文件中的.OBJ 文件的日期与时间置为 0，然后再组装（Make）。这样，若用户因 Ctrl+Break 键中断了 Build All 命令，只要用 **Compile/Make EXE File** 即可恢复。

⑤ **Primary C file**（主 C 文件）：当编译含多个.H 头文件的单个.C 文件时，Primary C File 命令是很有用的（但并非必要的）。若在编译过程中发现错误，包含错误的文件（.H 或.C）将被自动装入编辑器，可对其进行修改。但必须注意，.H 文件只有在已将 **Options/Environment/Message Tracking** 缺省设置改为 All Files 时才能自动装入，而原缺省设置不会自动加载.H 文件。即使.C 文件不在编辑器，但只要一按 Alt+F9，.C 主文件即被重新编译。

⑥ **Get Info**（获得信息）：**Compile/Get Info** 开辟一窗口如图 1.15 所示，给出如下信息：

当前命令、源文件、文件大小等信息。

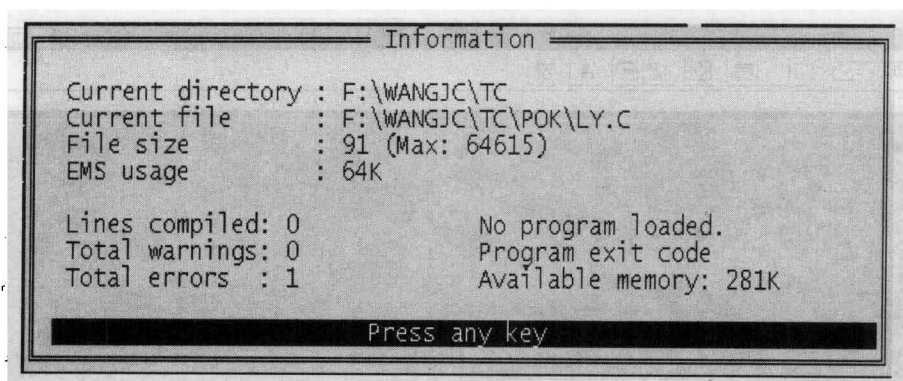


图 1.15

(5) Project 菜单

当选中 Project 菜单后, 将出现一个子窗口, 如图 1.16 所示。

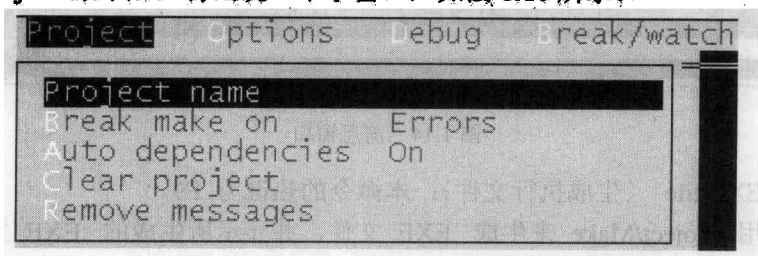


图 1.16 Project 菜单

① Project name : 选择一个包含将要编译、链接的文件名的 Project 文件, 项目名也将是以后要建立的.EXE 文件名。典型的项目文件具有.PRJ 扩展名。

② Break make on : 提供用户说明中止 Make 的缺省的条件, 如警告 (Warnings)、错误 (Errors)、致命错误 (Fatal Errors)。

③ Auto dependencies (自动依赖): 这是一个开关。当置为 On 时, 项目组装 (Project-Make) 自动检查每个项目表中在磁盘上有相应.C 文件的那些.OBJ 文件的依赖关系。所谓自动依赖关系检查是指: 项目组装打开.OBJ 文件, 寻找包含在源代码的那些文件的有关信息。这种信息总是在编译源模块时即被 TC 放进.OBJ 文件了。此时, 把每一个组成.OBJ 文件的源文件的日期/时间信息与.OBJ 中的进行比较, 若不同, 则重编译.C 源文件。

若 Auto dependencies 开关置为 off, 则不进行这种检查。

④ Clear Project (清除 Project): 该命令清除项目文件名, 重置消息窗口 (Message window)。

⑤ Remove messages (删除信息): 该命令把错误消息从消息窗口中清除掉。

(6) Options 菜单

当选中 Options 菜单后, 将出现一个子窗口, 如图 1.17 所示。

① Compile (编译器): 本命令将产生一个子菜单, 为用户提供说明硬件配置、存储模型、调试技术、代码优化、诊断消息控制以及宏定义等, 如图 1.18 所示。