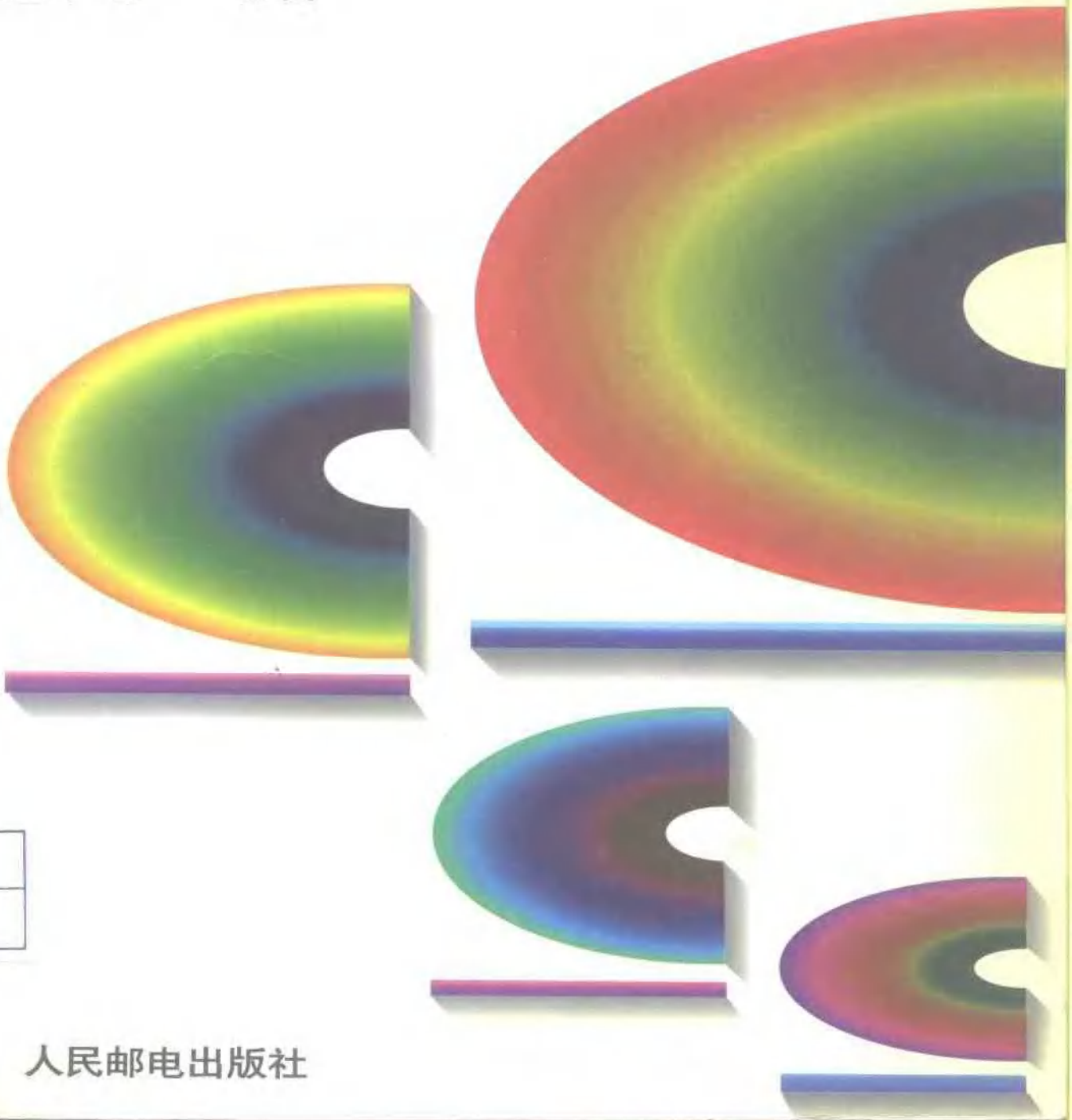
怎样学用Turbo C

程不功　　编著



人民邮电出版社

怎样学用 Turbo C

程不功 编著

人民邮电出版社

怎样学用 Turbo C/程不功编著. —北京:人民邮电出版社,1995.7
ISBN 7-115-05697-8

Ⅰ.怎… Ⅱ.程… Ⅲ.C 语言—程序设计 Ⅳ.TP312C

内 容 提 要

本书以广泛流行的 Turbo C 为例,由浅入深地介绍了 Turbo C 的基本概念、编程方法和技巧。全书共分十五章。第一至第七章是 Turbo C 的基础部分;第八至第十二章则介绍了有关 Turbo C 的窗口菜单程序设计、ROM BIOS 调用、文件操作、快速写屏及屏幕绘图实现方法等实用编程技巧;第十四、十五章则针对 C 语言大型程序的模块化设计等重要问题作了讨论。

全书文字精洁,通俗易懂,配有大量典型程序实例,并附有练习题,十分适合 C 语言自学使用,也可作为大专院校的教材或参考书。

怎样学用 Turbo C

ZEN YANG XUE YONG Turbo C

程不功 编著

责任编辑 贾安坤

人民邮电出版社出版发行

北京朝阳门内大街东厂胡同 11 号

北京顺义向阳胶印厂印刷

新华书店总店科技发行所经销

开本:787×1092 1/16 1995 年 8 月 第 一 版

印张 16.75 1995 年 8 月 北京第 1 次印刷

字数:413 千字 印数:1— 8 000 册

ISBN 7-115-05697-8/TP·214

定价:22.00 元

前　言

70 年代初，Unix 操作系统和 C 语言一诞生，就立即以它们崭新的面貌引起了计算机科学界的广泛注意，20 多年来的实践和考验又进一步证明了它们强大的生命力。目前 C 语言已经发展成为应用最为广泛，广大计算机人员最为喜爱的计算机语言之一。不少计算机专业工作者和计算机业余爱好者迫切希望能学会这种计算机语言，已经有些基础的则希望能进一步掌握它的应用。这本书就是在这样的背景下，为适应这两部分读者的希望而编写的。本书力求在下面几方面写出自己的特色：

第一，原理和应用相结合，软件和硬件相结合。毫无疑问，计算机语言的学习是以应用为目的，以软件为重点的，但结合应用讲清一些原理，结合软件介绍一些有关的硬件知识，能加深理解，为灵活应用打好基础。特别对于 C 语言来说，了解这些知识能进一步发挥语言的特长。

第二，叙述力求深入浅出，简明易懂。经验证明，通过程序学习计算机语言是最有效的方法。因此本书精选了大量经过验证的典型程序，并对这些程序作了简明的注解。根据作者多年的教学经验，对于那些概念容易模糊，程序容易产生错误的地方则不仅从正面，有时也从反面加以强调。

第三，每章后面均有选择、判断、思考题以及程序设计题，并在书末附上部分参考答案。读者如能认真完成这些练习将能大大加深和巩固所学知识。为了使读者写出的程序能与书上一致，我们选择了当前流行最广的 Turbo C 作为蓝本，实际上，只要掌握了它，其它类型的 C 也很容易学会。

全书分十五章，从第一到第七章是 Turbo C 的基础部分，第八、九章是关于屏幕窗口菜单设计，第十章是关于结构联合以及 ROM BIOS 调用，第十一章为快速写屏，第十二章为屏幕绘图，第十三章为文件操作，第十四、十五章补充一些其它重要问题。

本书特别适合那些希望能够通过自学较快较深入掌握 C 语言的读者。对于有一定基础的读者可以跳过前六章，直接从第七章开始。

本书也可作为大专院校的教材或参考书。

编写本书的过程中，陈斌启、程江、肖杨等同志在找资料、校对等方面做了大量的工作，在此一并表示谢意。

作者

1994 年 8 月　于长沙大学

目　录

第一章 概 论

第一节 C 语言的诞生和兴起

70 年代初，由美国贝尔实验室的 Thompson 和 D. M. Ritchie 合作开发的 UNIX 操作系统和 C 语言诞生了。像一对孪生姐妹，她们以自己崭新的面貌一开始就引起了人们的广泛注意。后来又经过不断改进和实践的考验，这对姐妹已迅速成长和成熟，显示出了强大的生命力，被公认为最优秀的操作系统和计算机语言之一。20 多年来，C 语言帮助了 UNIX 的成功，UNIX 的发展又推动了 C 语言的普及和发展。

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 语言移植到其它机器的工作大大简化，1983 年美国国家标准化协会（ANSI）对 C 的发展与扩充又制定了新的标准。从而使 C 发展成一种独立于 UNIX，独立于具体计算机类型的计算机语言。目前，C 语言已经能够运行于各种类型的机器和各种类型的操作系统环境中，甚至于在同一种机型下还提供了多种 C 语言编译系统，例如 IBM—PC 机，在 DOS 操作系统支持下的 C 语言编译器就有 C86、C88、MSC、Turbo C 和解释 C 等。

目前 C 语言已经风靡全球，成为当今世界上最为流行，广大程序设计者最为喜爱的计算机语言之一。

C 语言究竟具有哪些特点呢？

第一，C 语言是一种中级语言。

C 语言是计算机的中级语言，是指它既具有高级语言的通用性及易写易读的特点，又具有汇编语言（低级语言）的"位处理"，"地址操作"等能力，加上它的规模比较适中，因此称它为中级语言，也能突出它的特点。C 语言的这个特点使得它成为应用范围最为广泛的语言，它不仅能像 PASCAL、BASIC 等高级语言那样用于数据处理和管理软件的设计，还能像汇编语言那样用于计算机系统软件和控制软件的开发。

第二，C 语言是一种结构化语言。

函数是 C 语言程序的基本单位，函数之间除了必要的信息交换以外彼此独立。这种结构方式可以使程序层次清晰，便于维护、调试和使用，为模块化程序设计（一种先进的程序设计方法）提供了有力的支持。以 C 为基础发展而成的 C++ 语言目前已被认为是"面向对象"程序设计的比较理想的语言。

第三，语句简练、紧凑，语法规定少，数据类型丰富，编译后生成的代码质量高，运行速度快。

第四，可移植性好，用 C 语言编写的程序可以从一种环境不加或稍加改动就搬到另一种环境中运行。

尽管 C 语言有很多优点，但也存在一些缺点和不足，比如它的类型检验弱，转换比较随

便、优先级太多、不便记忆等。这些都对程序设计者提出了更高的要求、也给初学者增加了困难。

第二节　C 语言的基本结构

每个 C 语言程序都是由一个或多个函数组成的，函数是组成程序的基本单位。下面是一个最简单的程序。

```
/* Example */
main()
{
  printf("I learn Turbo C");
}
```

现在以上述程序为例、对 C 语言的函数结构加以说明：

1. 函数名：每个程序由一个或多个函数组成，每个函数都有一个名字、不论有几个函数、其中必然要有一个函数名是 main。不管 main() 函数放在什么位置，它总是程序中最先执行的函数。函数名后面的圆括号是放"参数"的、如果没有参数、也应写上圆括号、如本例中 main()。

2. 函数体：紧接着函数名之后、用大括号("{"与"}")括起来的部分、为函数体。函数体可能包括一句或多个语句(本例中只包括一句)。

3. 每条语句的最后必须用分号(;)结束。分号是 C 语句的结束符、也是各语句之间的分隔符。

4. 大写与小写：在 C 语言中、大写和小写字符是有区别的。C 语言程序中的语句都用小写字母书写。比如函数中的 main 和 printf 都要使用小写字符、不能用 MAIN 或者 PRINTF 代替。

5. 书写风格：书写程序时要注意书写的格式。比如上面的程序虽然也可以写成一行，如：

```
main(){printf("I learn Turbo C");}
```

但这样的格式不便于阅读和理解。为增强程序的可读性、书写程序时、既要注意分行又要注意缩排、保持良好的书写风格。

6. 注释：注释语句放在符号/* */之间，如本例的/* Example */。注释允许放在程序的任何位置，它的作用仅在于便于人们阅读和理解程序、对程序本身的运行没有影响。为了使程序易于被人理解、一个好的程序、应该包括恰当的注释语句。

第三节　Turbo C 2.0 概述

如前所述，C 语言有多种不同的编译器、它们之间只有细小的区别、学会其中一种、对其它几种也能很快适应。本书选择了当前流行最广、操作最为简便的 Turbo C 作为上要讲述对象。为此、这里还需介绍一点有关 Turbo C 的情况。

一、Turbo C 的产生和发展

Turbo C 是美国 Borland 公司的产品、这是一个专门开发软件的公司、曾开发了一系列

Turbo 产品，如 Turbo BASIC、Turbo PASCAL、Turbo Prolog 等等。该公司 1987 年首次推出了 Turbo C 1.0，使用了令人耳目一新的"集成开发"环境，即将文本编辑、编译、连接、运行等一体化，大大方便了程序的开发工作。1988 年该公司又推出了 Turbo C 1.5 版本，增加了图形库和文本窗口函数库等。1989 年推出了 Turbo C 2.0 增加了查错及"协处理器"的仿真等功能。

二、基本配置

Turbo C 2.0 可运行于 IBM—PC 系列微机及其兼容机上。要求有操作系统 DOS 2.0 或更高版本的支持。内存至少 448 K，可在任何彩色、单色 80 列监视器上运行。

Turbo C 2.0 软盘内容简介

Turbo C 2.0 有两张高密软盘(或六张低密软盘)，包括的主要文件有：

INSTALL.EXE	安装程序文件
TC.EXE	集成编译器
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	信息文件
TCCONFIG.EXE	配置文件转换程序
MAKE.EXE	项目管理工具
TCC.EXE	命令行编译器
TLINK.EXE	连接器
TLIB.EXE	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	头文件
*.BGI	显示器图形驱动程序
*.C	例行程序(源程序)

其中？分别为：

T	Tiny(微模式)
S	Small(小模式)
M	Medium(中模式)
C	Compact(压缩模式)
L	Large(大模式)
H	Huge(巨大模式)

三、运行环境

可能有相当多的读者熟习 BASIC 语言，因此有必要先介绍一下运行 Turbo C 与运行

BASIC 的差别，然后再讲述开发和运行 Turbo C 程序的操作。这样，你就可以在后面章节的学习中实际运行一些程序了。经验证明：一边学习，一边运行有关程序是自学计算机语言最有效的方法。

1. 解释型与编译型的区别

程序是由若干语句所组成的，这些语句就是一条条指令，它规定了计算机的执行过程。人们从键盘键入的程序，称为"源程序"，源程序中使用的是人们容易理解和记忆的符号（如 if、while、repeat 等都是类似英文的语句），这些符号计算机自己却并不能识别。计算机能够识别的，只有二进制代码，其它符号一概不认识。为了使计算机执行这些程序，就有必要将源程序翻译成二进制代码，这个由二进制代码组成的程序，称为"机器代码程序"或者称"目标程序"。由源程序翻译成目标程序是由系统软件自动完成的。人们常把这个系统软件称为"编译器"。编译器的翻译过程可以采用两种不同的方式，一种叫"解释型"，另一种叫"编译型"。

（1）解释型方式：早期的 BASIC 语言几乎都采用解释方式。它的特点是，用于翻译的"解释程序"必须常驻内存，执行程序时，翻译一条语句执行一条，如果下次要执行时又要重复翻译和执行的过程。这样，不仅速度慢而且占用内存较多。

（2）编译型方式：FORTRAN、COBOL、C 等高级语言几乎全部采用编译型，它的特点是在程序运行以前，先一次性地将源程序翻译成机器代码，并将这个由机器代码组成的程序保存起来。以 C 语言为例，若源程序名为"exam.c"，翻译后存入的目标程序将自动命名为"exam.obj"（注意：只改变了程序名的后缀）。不论这个程序以后要运行几遍，只要程序本身没有改变，这个翻译工作不再重复（当然，程序如果改变了，必须重新编译），因此程序运行的速度比较快而且占用的内存也比较少。

但是，光有编译，程序还不能运行，还必须进行"连接"。为什么要进行连接呢？这里有几方面的原因：第一，程序可能要用到某些函数，如 sin()、cos()、"输入"、"输出"等，其中很多是通过系统的"库函数"完成的，因此你的程序必须与库函数连接起来。第二，你的程序本身可能比较大，划分成若干个较小的程序，分别进行编译和调试比较方便，但是为了一道运行，必须先进行连接。这类情况我们将在后面的章节里详细讨论。另外，程序在内存中最后的定位工作，也要在这个阶段完成。经过连接后的程序我们称之为"可执行程序"，简称为"映象"，这个程序将以另一个名字存储起来。现在仍以 exam.obj 为例，经过连接后存入的文件名将自动使用 exam.exe（后缀用的是 .exe）。

归纳起来，从键入源程序到运行该程序，需要经过四步：

——通过编辑产生源程序（程序名为 *.c）

——通过编译产生目标程序（程序名为 *.obj）

——通过连接产生可执行程序（程序名为 *.exe）

——运行可执行程序

这些步骤是一个有序的过程，它们中的每一步都是在前一步完成的基础上进行的，在前一步完成以前后一步无法开始。

2. 开发和运行 Turbo C 的两种环境

有两种不同的环境可以用来开发和运行 Turbo C 程序：

一种是"C 命令行"环境，即在 DOS 操作系统的环境下，键入命令逐步完成编辑、编译、链接和运行等项工作。这种方法操作起来比较复杂，但有的情况下是必需的，例如当要把汇编语言的语句嵌在 C 语言程序中一道运行时，就必须采用这种方法。关于命令行的操作方法参见

附录D。

另一种开发环境是"集成"环境。这个环境的英文为"Integrated Development Environment"，简称IDE。在这个环境中，开发程序所需要的操作都以"菜单"和"窗口"的方式集中显示在屏幕上，操作比较简便、直观，不必记忆众多的操作命令，内面是一个理想的开发和学习C语言的环境。本书将主要讲述在这种环境下开发和运行Turbo C程序的方法。

四、在集成环境下的基本操作

1. Turbo C 2.0的安装和启动

Turbo C 2.0的安装非常简单，只要将包括Turbo C 2.0的两张高密软盘（或六张低密软盘）中的第一张插入A驱动器，在DOS的提示"A>"下键入以下命令：

A>INSTALL ↙

屏幕上将显示三种选择：

(1) 在硬盘上创建一个新目录来安装整个Turbo C系统。

(2) 对Turbo C 1.5版本进行更新。

(3) 为只有两个软盘而无硬盘的系统安装Turbo C。

一般情况下选择"1"。后面的操作只要按对盘号的提示，顺次插入各个软盘，就可以完成全部安装工作。安装过程中，自动在C盘的根目录下建立了一个TC子目录，在TC的下面又建立了两个子目录LIB和INCLUDE。在LIB子目录下面存放着"库文件"，在INCLUDE子目录下面存放着所有的"头文件"。

2. 在集成环境下的简要操作

首先用命令cd \tc进入TC子目录，然后键入下列命令：

C> tc↙

或者

C>tc 文件名↙

即可进入Turbo C的集成开发环境，在屏幕上显示如图1－1所示内容。

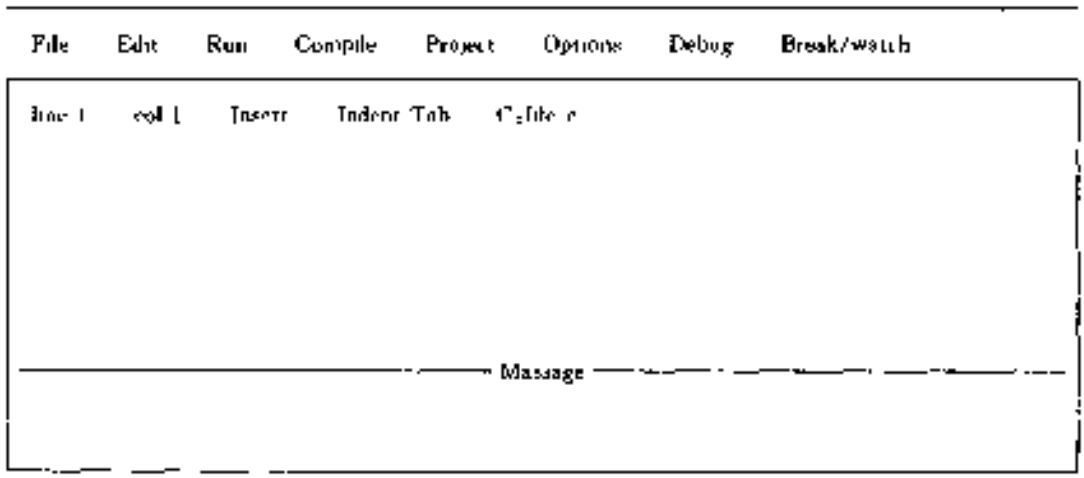


图1－1 集成环境的屏幕显示

图中第一行是"主菜单"，包括的内容及其含意如下：

File	Edit	Run	Compile	Project	Options	Debug	Break 'Watch
（文件）	（编辑）	（运行）	（编译）	（项 目）	（选 择）	（调试）	（断点及监视）

"主菜单"中的每一项，除 Edit 以外都有"子菜单"，按下[Alt]键同时按下项目的第一个字母，光标就会落在该选项上，例如按下 [Alt]+[F] 键就选择了 File 项，按下[Alt]+[E] 就选择了 Edit 项等。再按[Enter]键，屏幕上将显示该项的子菜单。也可以先按 [F10] 键，光标将出现在主菜单上，然后用光标移动键[←]或[→]选择好项目后回车，效果与前面的操作完全一样。

下面通过一个简单的例子来说明开发一个程序的基本操作，读者在掌握了这些基本操作以后如果需要进一步了解其它操作时，再参阅附录 A。

(1) 编辑一个程序：

在 DOS 的提示下键入命令：

```
C:> tc file c↙
```

将进入上述集成环境，如果磁盘上原来已经存有这个文件，则把该文件调入内存，并显示在编辑窗口上。假定现在磁盘上没有这个文件，则将创立一个新文件，文件名是 file.c。现在键入下列程序：

```
main()
  {
    printf(" This is an example.");
  }
```

(2) 将程序存入磁盘：

选择主菜单的 File 中子菜单的 Save 项，或者直接按[F2]键，程序将自动存入磁盘。

(3) 编译、连接和运行：

在 Turbo C 的集成环境中，编译、连接和运行的操作可以合并成一步，也可以分成几步进行，具体用那种方式合适，要看具体情况，比如小程序，用一步到位的操作方便一些，对大程序，则应采用分步的方法，其原因将在 14 章讨论。

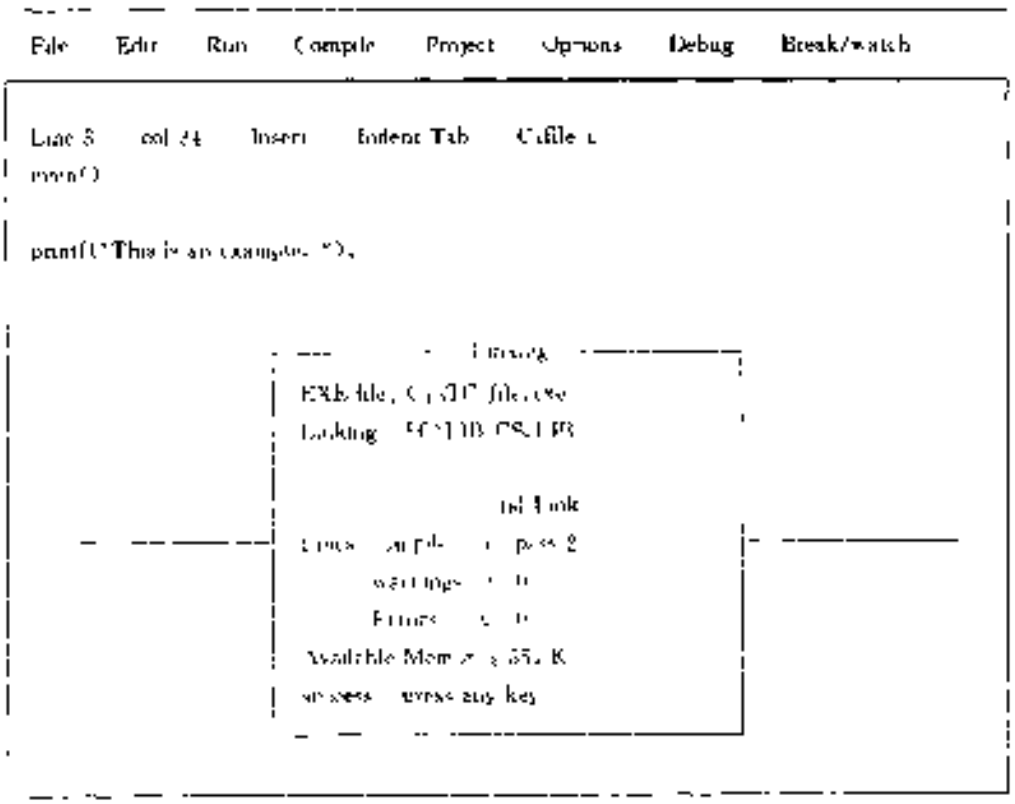
用合并的方法操作时，可选择 Run 子菜单中的第一项(或直接按[F9]键)来完成，系统将自动地进行编译、连接和运行等全部工作。

(4) 编译：

下面介绍分步时的操作：选择 Compile 中子菜单中的第一项将单独编译上述程序，并将编译过程中的信息显示在屏幕中央，这些信息包括：正在编译的文件名、编译的行号、多少警告信息(Warnings)、多少错误信息(Errors)、占用的内存数(Available Memory)等。在信息窗口的最后一行显示结论，如果提示 Success 则表示编译成功；如提示 Error，则表示编译失败。

(5) 连接：

只有在编译成功的基础上才能进行这一步，选择 Compile 子菜单中的"Link EXE file"项，系统将各目标程序包括有关的库文件连接到一起，组成一个可执行程序。连接过程中不断将有关信息显示的中央窗口上，情况如图 1—2 所示。



F1 Help F5 Zoom F6 Message F9 Make F10 Main Menu

图 1-2 出现在屏幕上的窗口

(6) 运行程序。

当编译和连接都通过后，可以在 Run 运行程序，上述程序的结果将显示：

This is an example.

有时显示的结果一闪而过又回到 Turbo C 环境下，为了能看清结果可选择 Run 项子菜单中的"User screen"项(或者直接按[Alt]+[F5])，暂时退出集成环境，看清结果后，再按任一键返回集成环境。

(7) 改正程序中的错误：

当编译过程中信息窗口提示有错误时，按任一键将弹到编辑窗口，同时程序中的某行将呈现高亮度，同时在信息窗口将出现一行或多行错误提示，其中一行说明了该行语法的出错。

例如前面的程序中，若遗漏了右边的双引号，高亮度将出现在 printf 行上，编译器将指出这一错误。不仅如此，因为既然没有右边的引号，编译器认为后面的符号均属字符串的一部分，因此还可能引发出其它错误。这种情况表明，程序中的某个错误可能带来一连串的错误提示，但只要改正了这一个错误，一连串的错误信息也就不存在了。

为了改正程序中的错误，按[F6]键后修改错误，以本例来说，只须在相应的位置上加上双引号就可以了。

改正后的程序必须重新编译。

(8) 退出集成环境：

按[Alt]+[X]键或者选择 File 中的 Quit 项即可退出集成环境，回到操作系统环境。