21世纪

高等院校计算机系列教材

C程序设计简明教程实验指导与实训

吴年志 主 编

李发海 薛 礼 陈雪刚 副主编

中国水利水电出版社

www.waterpub.com.cn

21世纪高等院校计算机系列教材

C程序设计简明教程
实验指导与实训

吴年志　主编

李发海　薛　礼　陈雪刚　副主编

中国水利水电出版社

内 容 提 要

本书是《C 程序设计简明教程》（王晓东　主编）一书配套使用的教学参考书，主要包括：C 程序设计环境介绍，重点介绍了 Turbo C 的程序开发环境，也简单介绍了一些 Microsoft Visual C++ 6.0 的开发环境；C 程序设计的实验篇；C 程序设计的实训篇；《C 程序设计简明教程》一书的部分课后习题参考答案。

书中的实验和实例都在 Turbo C 2.0 环境下进行了验证并通过。书中的实验和习题内容丰富，实训部分知识点回顾简明清晰，工程项目从提出问题、分析问题到解决问题思路清晰，具有启发性和综合性，不仅紧密配合理论教学，而且具有很强的实用价值。

本书既可以作为学习 C 语言和上机实践的参考书，也可以作为高等学校计算机程序设计的教学用书或培训教材。

图书在版编目（CIP）数据

C 程序设计简明教程实验指导与实训/吴年志主编.
北京：中国水利水电出版社，2007
（21 世纪高等院校计算机系列教材）
ISBN 978-7-5084-4314-0

Ⅰ. C…　Ⅱ. 吴…　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（2006）第 152194 号

书　　名	C 程序设计简明教程实验指导与实训
作　　者	吴年志　主编　李发海　薛　礼　陈雪刚　副主编
出版　发行	中国水利水电出版社（北京市三里河路 6 号　100044） 网址：www.waterpub.com.cn E-mail：mchannel@263.net（万水） sales@waterpub.com.cn 电话：（010）63202266（总机）、68331835（营销中心）、82562819（万水）
经　　售	全国各地新华书店和相关出版物销售网点
排　　版	北京万水电子信息有限公司
印　　刷	北京市天竺颖华印刷厂
规　　格	787mm×1092mm　16 开本　14.75 印张　359 千字
版　　次	2007 年 1 月第 1 版　2007 年 1 月第 1 次印刷
印　　数	0001—4000 册
定　　价	22.00 元

前　言

计算机程序设计课程是一门对实践性要求特别强的课程。为了配合《C 程序设计简明教程》，我们特地组织了实验教学经验丰富的教师编写了这本书。

本书内容包括四大部分。第一部分主要介绍 C 语言程序开发环境，重点介绍 Turbo C 的程序开发环境和上机调试过程，对部分要求高的读者也简单介绍了 Microsoft Visual C++ 6.0 的开发环境。第二部分是实验篇，重点介绍上机实验要求和《C 程序设计简明教程》对应章节的上机实验内容，包括 12 个实验，每个实验都包括实验目的、实验内容和实验要求，通过实验来领会教材中的内容。第三部分是实训篇，主要帮助读者回顾每一章节的重点内容，培养和提高读者分析问题、解决问题的能力，这部分主要包括知识点回顾、实训项目以及习题和解答。通过回顾、实训和习题，帮助读者理解和掌握教材中的每一个知识点，并提高动手实践能力。习题部分配有参考答案，方便读者参考。第四部分是部分课后习题解答，主要是针对《C 程序设计简明教程》课后习题中的部分重点和难点题目的参考程序设计。

本书第一部分以及第二部分（实验篇）和第三部分（实训篇）中的第 1～5 章由吴年志编写，第 6～8 章由李发海编写，第 9～10 章由薛礼编写，第 11 章由陈雪刚编写，第四部分（部分习题解答）由《C 程序设计简明教程》的作者王晓东提供，最后由吴年志、李发海和薛礼统稿并校对。在本书的编写过程中，得到了阎菲副教授的热情支持和指导，在此表示衷心的感谢！

由于时间仓促，加之编者水平有限，书中难免有错误和不妥之处，恳请专家与读者指正。如有意见和建议，可以通过电子邮件联系：wunianzhi@163.com。

编　者

2006 年 11 月

目　　录

实验篇

实训篇

C 语言程序开发环境介绍

C 语言源程序必须经过某种编译工具翻译成为目标机器语言程序才能够在计算机上执行。然而随着程序编写规模的扩大，顺利编写出正确的程序并非一件容易的事情，早期的许多编译工具仅提供翻译功能，已满足不了应用的要求，编程人员需要一种功能全面并高度集成的编译环境。

经过近十年的努力，现在许多软件厂商都推出了成熟的集成化编译开发环境，使用这些软件产品使 C 语言程序的开发人员能够在统一的界面和功能环境下解决应用问题，较大地提高了开发效率。在此，我们介绍微机上最为流行的几种由 Borland 公司、Microsoft 公司推出的编译系统。

Turbo C 2.0（简称 TC20）是 Borland 公司于 1989 年推出的 C 语言集成环境的编译器，具有编译速度快、代码优化效率高等优点，在当时深受喜爱，具有深远的影响。TC20 支持 DOS 系统下应用软件的开发，同时提供了集成编译方式和命令行编译方式，其集成开发环境的设计思想成为现在主要的编译开发环境的流行风格。

继 TC20 之后，Borland 公司对 C 语言编译环境的开发一直处于行业的领航位置，又先后推出了 TC++1.0、BC++2.0 等一系列支持面向对象 C++语言的编译产品。Borland C++3.1（简称 BC31）是 Borland 公司在 1990 年推出的有深远意义的开发工具，它结合了以往产品中几乎所有的优点，提供了 DOS 下最完善的开发环境。BC31 支持鼠标功能，支持多窗口操作，而且可以突破 DOS 的内存限制，充分利用 PC 机的扩展内存；除此之外，BC31 还提供了最早的在 Windows（win31）下运行的支持 Windows（win31）程序开发的集成环境。

在此之后，Borland 公司与 Microsoft 公司的 C++编译产品一直成为领域内的主要产品，其技术上的进步更是令其他对手无法企及，直至今天这种格局仍然保持着。在 Windows 成为微机上主流的操作系统之后，支持 DOS 应用的开发环境就不再显得具有生命力，因此，在 BC31 之后 Borland 推出的 BC++4.0、BC++5.0 以及当前广泛使用的 C++ Builder 系列产品都不再提供 DOS 下运行的集成式开发环境，而都是专注于 Windows 应用系统的开发方面。在 MSC 7.0 之后，Microsoft 公司也终于推出了全新的 Windows 下的优秀的开发工具——VC++系列产品，其优秀的设计思想与 Windows 操作系统的紧密结合，确立了 VC++产品的庞大用户群。从目前来讲，C++ Builder 产品和 VC++6.0 产品各有优势，在应用软件系统的快速开发方面 C++ Builder 是相当优越的，而在用于同 Windows 系统结合或访问控制底层硬件等产品的开发时，VC++取得了更多的认可。

0.1 Turbo C 集成开发环境的使用

Turbo C 是 DOS 上的一个 C 语言系统，它也能以全屏幕方式或窗口方式运行在各种 Windows 环境中。Turbo C 符合 ANSI C 标准。Turbo C 的组成部分包括 C 语言编译系统、连接系统、ANSI C 的标准函数库和一批扩充库函数等。另一重要组成部分是一个集成化程序开

发环境，用户可方便地在其中编辑、编译、调试和运行自己的程序。

这里介绍 Turbo C 2.0 集成开发环境的基本使用方法。

今天微机上的 C 语言系统很多，有些新系统的功能更强大。但 Turbo C 环境有许多特点，在基本的 C 程序设计课程教学中被广泛使用。本系统既简单又功能完整，开始使用时需要理解的概念少，容易入门，特别适合初学者。Turbo C 系统对计算机要求低，在任何微机上都能很好地运行，编译加工速度快，对于做程序练习和一般的程序开发都很合适。另外，Turbo C 的编程和调试环境也很完整，反映了集成化开发环境的特点。掌握了这个程序开发工具的使用方法，就掌握了一个实用的程序开发工具，也能为进一步学习使用其他编程工具打下很好的基础。

Turbo C 2.0 版编程环境的一个主要缺点是不支持鼠标操作，也不能同时编辑多个文件。Turbo 系列的后续版本还扩充为 C/C++语言环境，后来还扩充为 C/C++语言环境，它们以及 Borland C/C++系统的基本使用方式都与 Turbo C 2.0 系统很类似。

如果读者没用过 Turbo C 程序开发环境，正式使用前应阅读 0.1.1、0.1.2、0.1.3 节，这几节介绍了该系统的基本使用方法和用它编程的基本工作过程。0.1.4 节介绍程序调试方面的问题；0.1.5 节介绍系统的项目管理功能，主要用于较大程序的开发工作。最后一节介绍 Turbo C 系统的另一些情况。

0.1.1 Turbo C 集成开发环境的启动和退出

如果所用的计算机上已经安装好 Turbo C 系统，我们就可以使用了。为保证使用中不出现意外，使用同一计算机的人在工作中不互相冲突，应当先做一点准备工作。

1. 准备工作

为防止在使用中无意地破坏 Turbo C 系统本身（例如不当地删除或改写了系统文件），最好在自己的子目录中使用它。为此，首先要做的工作是：

（1）建立一个自己的工作目录。

（2）将 Turbo C 系统目录中的 TCCONFIG.TC 复制到自己的目录中，如找不到该文件，可以在启动 Turbo C 后，用 Options/Save Options 菜单命令在自己的子目录里建立配置文件 TCCONFIG.TC，有关操作见后。TCCONFIG.TC 是系统的配置文件，把它复制到自己的子目录下，既可以根据需要对 Turbo C 的工作方式做一些设定，又不会改变系统的工作方式，不会影响别人使用。

（3）检查操作系统路径变量 PATH，保证 Turbo C 的系统目录在路径中，然后就可以启动 Turbo C 集成开发环境了。

2. 集成开发环境的启动

在 DOS 系统提示符下键入 TC 并回车，数秒钟后 Turbo C 开发环境就会启动完毕，屏幕中央将显示系统版权信息。此时按任何键都将使开发环境进入正常工作状态（编辑程序文件的状态）。

以这种方式启动时，被编辑的程序文件自动命名为 NONAME.C。系统进入工作状态后的屏幕情况如图 0.1 所示。如果要编辑某个特定文件，可以在启动系统后将该文件装入（方法见后），也可以直接把文件名作为 TC 命令的参数。如果系统的设定被改动过，启动后的情况也可能与这里介绍的有差别，例如直接装入了某个默认的文件等。

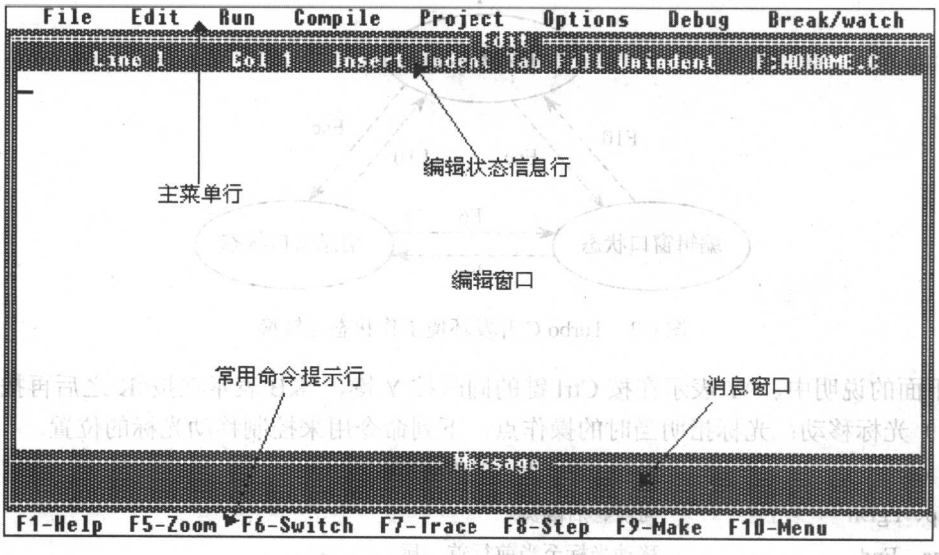

图 0.1 Turbo C 程序开发环境启动后的屏幕情况

在集成开发环境正常工作时，屏幕上的整个显示区域被划分成几个部分：

（1）最上部是系统命令主菜单行，排列着若干菜单命令按钮。

（2）最下面是常用命令提示行，其中说明了一些常用命令所对应的功能键。

（3）中间主要部分分为两个区域：上面区域是编辑窗口，被编辑的文件将显示在这里。下面区域有两个用途，有时作为消息窗口（标题为 Message，启动后就是这样），显示系统工作中产生的信息；有时作为程序调试的监视窗口（这时标题是 Watch），显示程序调试中的有关信息。

3. 系统的退出

在开发环境中工作时按 Alt+X 键即可退出 Turbo C。要求退出时系统可能提问："是否要保存正在编辑的文件"（如果文件被修改过），这时应该按照实际需要回答（Y/N）。也可以用菜单命令退出开发环境。

0.1.2 集成开发环境的使用

在集成环境工作过程中，任何时候都有惟一的一个活动部分，该部分将接受用户的键盘输入（命令等）并做出响应。环境中可能的活动部分有三个，分别是：命令主菜单、编辑窗口、消息（监视）窗口。当某部分处在活动状态时，我们就说开发环境处在这个状态。就是说，集成开发环境的可能状态有三个：主菜单状态、编辑状态和消息（监视）状态。用键盘的功能键可以令系统在不同状态之间转换。如图 0.2 所示是活动状态转换图，箭头表示状态转换，旁边标出了实现有关转换所用的功能键。

例如，在编辑状态下（编辑窗口活动时）按 F10 键，系统将转到主菜单状态，再按 Esc 键可使系统转回，其余类推。

1. 编辑状态的命令以及编辑器的使用

编辑状态下的集成环境是一个全屏幕的程序编辑器，它提供的编辑命令分成几组。这里只介绍常用的编辑命令，这些命令对于编辑程序而言已经足够。

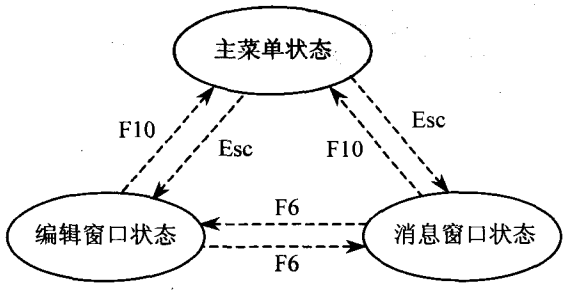

图 0.2 Turbo C 开发环境工作状态与转换

在下面的说明中，^Y 表示在按 Ctrl 键的同时按 Y 键，^KB 表示在按^K 之后再按 B 键。

（1）光标移动：光标指明当时的操作点，下列命令用来控制移动光标的位置。

→、←、↑、↓	向右、左、上、下移动光标
PgUp、PgDn	向前、后翻页
Home、End	移动光标至当前行首、尾
^QR、^QC	移动光标至文件首、尾

（2）插入删除：输入方式分为插入和改写两种，用功能键切换。

Ins、Insert	使输入方式在插入/改写方式间切换
Enter	换行符号
Del、Delete	删除光标处的字符
BackSpace（←）	退格键，删除光标前的一个字符

（3）块操作命令：一个块就是程序文件中的一段连续字符，块可以整体地复制、搬移、删除、写入到文件等。要进行编辑内容的块操作，首先要定义块：^KB、^KK 命令分别把当时的光标位置标记为块的开始和结束位置。

^KB、^KK	将光标位置标记为块的开始、结束位置
^KC	把被标记的块复制到当前光标位置
^KV	把被标记的块剪切到当前光标位置
^KY	将被标记的块删除
^KH	隐去或显示当前的块标记
^KW，^KR	将标出的块写入文件，从文件读入到光标位置

（4）查找和替换：查找替换操作需要提供其他辅助信息，包括被查找的字符串等，还要指定查找方式。有关方式的信息可以通过功能键 F1 查询联机帮助信息。

^QF	查找某一个字符串
^QA	查找一个字符串并用另一个字符串替换
^L	重复前一次查找

编辑窗口最上一行是编辑状态信息行，这里显示的不是被编辑程序的内容，而是当时的编辑状态信息，包括：当前光标所处的行列数、输入处于插入状态或改写状态、是否采用自动对齐（Indent）方式等。

自动对齐方式对写出格式清晰的程序非常有用。系统中可以设定按 Tab 键一次光标跳的格数，在自动对齐方式下回车，光标将自动与上一行的非空白位置对齐，按退回键（BackSpace，

键盘上标←）将使光标退到再前面的一个对齐位置。利用这几个键，可以方便地维持良好的程序格式。

按 Tab 键的默认跳格数为 8 个字符，可以自己设置，建议将跳格数设置为 4，这样可以避免光标移得太远。设定方式是：依次按 Alt+O、E、T、4 及回车键，按 Esc 键退出一层菜单后按 S 命令保存。应该把自己的系统设置存放在自己的子目录中，以免影响别人的使用。按这些键实际上是要求系统执行菜单命令，详情在后面解释。

编辑过程中还可以用功能键向系统发命令，主要功能键在屏幕最下一行提示，包括：

F1	查阅系统帮助信息
F5	扩大/缩小编辑窗口
F6	活动窗口转换
F7	跟踪执行，用于程序调试
F8	单步跟踪执行
F9	对当前程序生成可执行代码
F10	转到系统命令菜单

按下 Alt 键并保持几秒钟，提示行将显示出另一组功能键命令，包括：

F1	重显示前面最后一次的帮助信息
F3	显示用过的几个文件，供重新装入
F6	活动窗口切换
F7	查找程序中前一个错误的位置
F8	查找程序中下一个错误的位置
F9	编译当前文件

上述功能键在系统处于消息窗口状态时同样可以使用。除了上面这些以外，在编辑状态下还有一些功能键，常用的有：

F2	将当前文件存盘，并继续工作
F3	装入一个文件，或编辑一个新文件
Ctrl+F1	关联查询帮助信息，以光标位置的标识符作为查询索引
Ctrl+F9	运行当前生成的可执行文件

2. 菜单状态的命令

系统菜单用于发各种操作命令。在系统的菜单状态下发命令的方式有两种。在子菜单弹出的状态下，同样可以用这两种方式发子菜单命令。

（1）通过键入菜单项名称字符串中的高亮度字符。如主菜单 File 项的字符 F 为高亮度字符，可以看到它颜色特殊，在主菜单状态下按这个字符键，系统将弹出 File 子菜单。

（2）在菜单状态下用光标移动键操作菜单中的亮条，当亮条位于某菜单项时按回车键，系统即执行这个菜单项所对应的命令。例如在 File 子菜单中将亮条移动到 Save 位置并按回车键，系统将把当前编辑的文件存盘。

有些菜单项旁注有热键名，通过有关功能键同样可以执行命令。此外，在菜单状态下按一次 Esc 键可以使系统退回原来的状态，在子菜单状态下按一次 Esc 键将退回到上一层菜单。

编程环境的菜单命令很多，在主菜单的每个菜单项下（除了 Edit 项之外）都有一级或几级子菜单。这里只对其中最常用的一些命令做简单介绍。

- File 子菜单包含新文件编辑、编辑文件装入和保存、目录显示与转换、临时转入 DOS 系统、退出 Turbo C 环境等命令。
- Edit 命令使系统转到编辑状态。
- Run 子菜单命令包括：运行当前程序（子菜单项 Run，热键为 Ctrl+F9，若源程序修改过，系统将重新编译之后再运行），User Screen 项（热键 Alt+F5）用于查看程序运行的输出屏幕情况（按 Esc 键退回）。本菜单还包括几个用于以调试方式运行程序的命令项，其使用在后面介绍。
- Compile 子菜单用于要求系统对源程序进行编译加工。包括：编译当前文件、生成目标代码文件（Compile To OBJ 项，生成目标文件、.OBJ 文件）、连接目标模块、生成可执行程序（Link EXE File 项，生成可执行的文件）、连续完成程序编译和连接（Make EXE File 项）等。Get Info 项用于查看当前文件的有关信息。
- Project 子菜单用于使用项目管理功能开发较大的程序。有关问题的讨论见 0.1.5 节。
- Options 子菜单用于设置和修改与系统工作方式有关的各种参数，它有许多子菜单，分为若干层。初学者对许多选择的意义不清楚，最好不要随便改动系统的原有设置。如果无意中改变了某些设置，只要不执行 Save Options 命令项，系统退出后再启动工作方式就不会改变。这里只建议改变 Environment 子菜单中的 Tab 项，将原设置值 8 改为 4，这样在编辑程序时看着舒服些，也避免了几次退格就使程序正文超出屏幕范围。修改设置后按 Esc 键退回，并执行 Save Options 命令保存。0.1.6 节介绍有关情况。
- Debug 和 Break/Watch 菜单用于程序调试，这方面问题见 0.1.4 节。

可以看到 Turbo C 环境的常用命令都可以通过功能热键的方式使用。熟悉一些常用的命令热键可以提高工作效率。

3. 消息窗口状态

在编辑状态下用 F6 功能键能使系统转入消息窗口状态。在消息窗口状态下，提示行列出的各功能键仍可以使用。

前面讲过，系统用消息窗口显示程序加工中发现错误的有关信息，在出现错误时系统将自动进入消息窗口状态，窗口里显示出一些错误信息行。用光标移动键可以将消息窗口中的亮条移动到任一个消息行，与此同时，系统将自动对该消息在编辑窗口的源程序中定位，把编辑窗口中的亮条和光标移到产生这个消息的位置，即编译程序发现程序错误的地方。

程序开发过程中利用消息自动定位机制查找程序错误的方法将在下面介绍。

0.1.3 编程的一般工作过程

在 Turbo C 集成开发环境中编程，一般的工作过程如下：

（1）启动集成环境。一般用：

TC　　被编辑文件自动设为 NONAME.C（可以另行设置默认文件）。

TC 文件名　　以指定文件作为被编辑文件。

如果指定文件存在，系统自动装入它；否则就按名字建立新文件。若指定文件时未提供扩展名，则自动加.C 扩展名。

（2）用开发环境的编辑器输入源程序代码或修改已有的代码。

（3）在源程序编辑完成后，用功能键 Alt+F9（或通过菜单）发命令，要求系统对当前源

程序进行编译。这也要求系统帮助检查源程序中的语法错误。

（4）如果编译中发现问题，错误和警告信息将显示在消息窗口。每行表示一条错误（或警告）信息，包括发现问题的位置和有关说明。这时系统自动进入消息窗口状态，亮条位于一个消息行上，编辑窗口里用亮条和光标指出发现问题的确切位置。

此时应当仔细阅读消息行的文字内容，观察系统所指程序位置及其上下文，分析出现问题的原因。用 F6 键切换到编辑窗口改正程序中的错误。然后再用 F6 键切换回消息窗口，把亮条移到下一个消息行，定位和改正下一个错误。

检查程序错误时应注意几点：

- 有时程序中的一个错误会引发一系列错误信息，工作中不应被这种情况所迷惑，改正了一些错误后应及时对源程序重新进行编译。
- 如果修改错误时增删了行，或是一个行里有多个错误，更正前面错误时增删了字符，就可能导致系统对错误定位不准，此时应该重新编译。
- 系统给出的警告信息一般都说明程序中有问题，因为系统发现了可疑情况。对于警告信息同样要逐个仔细分析。除非明确认定不是问题，否则绝不能简单地认为不是错误而不予理睬。实际上，很多警告都是因为程序中确实有严重的隐含错误。

重复（3）、（4）两个步骤，直到排除了系统发现的所有错误。

（5）源程序能正常完成编译，没有错误时可以进入下一步：用 F9 功能键（或菜单命令方式）发出连接命令，要求系统建立可执行程序。在连接中发现新错误也需要仔细检查和修改程序。连接时发现的错误一般是由于函数名或外部变量名字写错，或者一些函数、外部变量没有定义引起的。系统不能对连接错误给予自动定位，只能提供有关的名字信息等。对于这类问题，可以借助编辑器的字符串查找命令进行定位。

（6）连接正常完成后，可执行程序已经生成完毕。这时可以启动程序，进行运行和调试。用 Ctrl+F9 或菜单 Run 命令发出运行命令，在程序运行中和运行后检查其表现是否正常。用 Alt+F5 功能键或菜单可以切换到用户屏幕，检查程序输出。如果程序运行中发生运行错误，或者输出结果不正确，则需要进一步检查源程序代码，找出并排除错误，直到得到正确的程序为止。下面要介绍的集成开发环境调试功能可以帮助查找在程序运行中出现的错误。

为能高效地编写出正确的程序，除了认真地分析要解决的问题、熟悉程序语言（C 语言）、熟悉编程环境功能之外，还有一些应当注意的事项，主要是：

- 应重视程序设计方法。分析问题时，应考虑如何将复杂问题分解成简单问题的组合，如何进一步分解。在此基础上考虑如何用语言机制实现计算过程。要特别注意具有独立逻辑意义的片段，将它们孤立出来定义为函数。将程序适当分解为一组互相调用的函数是处理复杂问题的基本方法。一个函数不宜过长，有人曾建议一个函数的定义不应超过一页。一组较小的、本身有逻辑独立意义的模块的组合比一个大而复杂的、没有良好结构的长程序更容易把握，更容易写正确，出现了问题也容易发现和改正。
- 注意源程序代码的形式，采用易读的、一致性的换行和退格形式，使程序的书写形式能明显反映出程序的结构。良好的书写形式有利于人理解程序，不但能帮助减少编辑错误，也有利于在调试中发现错误，有利于对程序的进一步修改和重复使用。良好的编程习惯也能使工作更有效率。从写简单程序开始就应当注意养成好的编程习惯。

- 在程序里必要的地方加一些注释有助于程序的理解和修改。
- 注意利用系统的帮助功能（F1 和 Ctrl+F1 功能）。要学会阅读理解系统的提示和帮助信息。出错时要仔细阅读系统提供的错误信息，弄清楚意思，不要盲目、草率地处理。

0.1.4 程序调试

调试是程序开发过程中一个必不可少的阶段。程序初步完成后要经过调试，实验性地运行，设法确认程序没问题，或者找出程序中潜藏的错误。调试的基本出发点是设法发现程序中的错误，基本方法是选择一些数据实例，令程序用这些数据运行，考查运行过程和有关结果。如果在实例运行中发现错误，则应设法确定出错原因并予以排除。

经过反复调试运行和纠正错误，就可能使程序中的错误变得更少。对于小程序，到某个时候有可能认为程序中已经没有错误了。对于大程序，实际软件系统，人们通常不敢说这种话，只说：这个系统已经经过仔细调试，发现的错误大部分已经解决，现存的错误还有……大软件实在太复杂，以至于可能无法完全把握它们，这是软件世界中的一个客观情况。计算机领域的人们都知道一句名言：调试只能说明程序中有错误，不能说明其中没有错误。程序调试有其内在的局限性，但即便如此，调试仍然是软件开发过程中必要的一环，需要认真对待。

如果在程序调试中发现了错误，就要设法确定问题原因，确定程序中出错的位置。寻找出错位置的基本方法是对程序本身做进一步的分析，另一个常用方法是检查程序的执行流程和运行中各变量的变化情况。为此可以在源程序的适当位置插入输出语句，输出有关变量的值检查。这个方法的缺点是比较麻烦，而且对程序的运行过程无法控制。

今天的开发环境都为程序调试和排错提供了强有力的支持。Turbo C 集成开发环境的功能很典型，其他系统的功能都与它类似。Turbo C 提供的与调试有关的功能包括两个方面：一是能按照源程序中的语句一步步执行，或者要求它执行到某语句时停在那里；另一功能是能够在运行中直接观察指定变量（或表达式）的值。

（1）单步执行及控制执行，用于指定程序的执行方式。命令有三个，可以通过功能键发命令，也可以通过菜单选择：

F7　　单步执行。令程序执行一个语句，遇到函数调用时执行进入函数体。

F8　　与 F7 类似，但执行不进入函数，把函数调用简单地看做一个语句。

F4　　使程序连续执行，直到达到当前光标所在的语句，停在那里。

（2）断点（BreakPoint）设置。断点是源程序中标出的位置，程序执行达到断点时将自动停住。可以同时设置多个断点。

Ctrl+F8　　断点设置与取消。在当时的光标位置设一个断点。如果光标当时放在已设置的断点上，则取消这个断点。

在程序中设了断点以后，如果再要求程序连续运行（通过菜单命令或者功能键 Ctrl+F9），程序将执行到所遇到的第一个断点并停在那里。此后再命令程序继续，它将执行到下一个断点位置。断点和单步执行等可以混合使用。

（3）设置监视（Watch）表达式。在调试运行时，屏幕下部的消息窗口自动变为监视窗口，用于显示被观察表达式（可以是简单变量）的值。这种表达式称为监视表达式。在程序的执行终止时，例如在单步执行中，或在连续执行遇到了断点时，在监视窗口中将显示被监视表达式当时的值。与监视表达式有关的操作包括：