（附磁盘）

# C语言高级编程教程

## ——技巧与捷径

郑秋宝　编著

学苑出版社

# C#语言
# 高级编程教程

北京希望电脑公司计算机程序设计语言系列丛书

# C语言高级编程教程

## ——技巧与捷径

郑秋宝 编著
杜运炜 校
刘文姗

学苑出版社

1993年·北京

(京)新登字 151 号

## 内 容 摘 要

本书结合作者多年从事 C 语言教学和 C 软件设计的经验，以 Turbo C 2.0 为对象介绍了一种高效学习 C 语言的方法——"错误分析法"。该方法在作者的教学实践过程中受到了广大同学的欢迎和有关部门的肯定，收效不错。在本书的每一章中均有一节专门介绍和讨论学习本章内容时容易犯的错误及其解决方法，相信您会从中获益。另外，本书特别注重程序设计中模块化思想的介绍，并适时介绍了一些软件工程知识。本书提供的所有例程均上机调试通过，读者可以直接使用。

本书既可以作为研究生，大学本、专科学生的《C 语言》教材，也可以用作广大工程技术人员以及所有渴望迅速而正确掌握 C 语言的计算机爱好者的技术参考书和上机操作指导书。

欲购本书，请直接与北京 8721 信箱联系，邮政编码 100080，电话 2562329。

北京希望电脑公司计算机程序设计语言系列丛书

**C 语言高级编程教程**

**——技巧与捷径**

---

**作　　者:** 郑秋宝
**审　　校:** 杜运炜　刘文姗
**责任编辑:** 徐建军
**出版发行:** 学苑出版社　　**邮政编码: 100032**
**社　　址:** 北京市西城区成方街 33 号
**印　　刷:** 北京市朝阳区小红门印刷厂印刷
**开　　本:** 787×1092 毫米 1/16
**印　　张:** 37　　**字数:** 868 千字
**印　　数:** 1—5000 册
**版　　次:** 1993 年 12 月北京第 1 版第 1 次
ISBN 7-5077-0807-1/TP.18
**本册定价:** 39.00 元(含盘)

---

# 前　言

自 70 年代以来，C 语言作为一种结构化的通用程序设计语言得到了广泛的普及，尤其是 UNIX 系统的发展和普及，使作为 UNIX 系统主力语言的 C 语言应用面越来越大，几乎涉及当今科技、生产、日常生活、办公、商业等各个方面。

随着计算机及其应用技术在我国的日益普及和高速发展，在众多的程序设计语言中，C 语言以它丰富的数据类型，方便灵活的运算功能，模块化的数据结构和流程控制，高效的语句效率，良好的人机界面和强大的接口操作能力等特点，已成为众多程序员的优选语种。许多工程技术人员更是将它称为"具有高级语言和汇编语言两者优点的中级语言"。或者称之为"工程设计语言"；而系统程序员则因它在 UNIX 系统开发中的功绩，称之为"系统开发语言"……。虽然这些说法不太准确、全面，但的确反映了 C 语言具有一些其它语言无法比拟的优点。

为满足计算机爱好者，工程技术人员，大学本、专科学生以及所有渴望迅速而正确掌握 C 语言人士的需要，本书结合作者多年从事 C 语言教学和 C 软件设计的经验，介绍了一种高效学习 C 语言的方法 —— "错误分析法"。该方法在作者的教学实践过程中受到了广大同学的欢迎和有关部门的肯定，收效不错。在本书的每一章中均有一节专门介绍和讨论学习本章内容时容易犯的错误及其解决方法，相信您会从中获益。另外，本书特别注重程序设计中模块化思想的介绍，并适时介绍一些软件工程知识。

本书共分三部分：第一部分为基础教程，主要介绍 C 语言的基本运行环境，基本数据类型及其运算，程序流程的控制，数组和基本上机操作技巧等内容；第二部分为高级教程，进一步介绍 C 语言的工作环境，结构，指针，动态内存管理，文件，图形的操作及其使用技巧等 C 语言的高深内容；第三部分为附录，主要介绍 Turbo C 2.0 的一些辅助性知识，如运算符优先级，库函数简要用法，实用工具等可能帮助您对 C 语言运用自如，事半功倍的内容。

本书以 Turbo C 2.0 为主要介绍对象。其实，其它 C 语言与 Turbo C 大同小异，更何况掌握了 Turbo C 的读者仅需了解两者之间的差别，在非常短的时间内，甚至不费什么时间便可熟练掌握。需要说明的是，本书提供的所有例程均上机调试通过，可以直接使用。

由于时间仓促，作者对 C 语言的了解有限，错误和不当之处在所难免，望广大读者批评指正。

在本书的编写过程中，得到了许多同志的帮助，为此向他们表示感谢。需特别指出的是：北京希望电脑公司资料部的秦人华经理在本书的印刷和出版过程中自始至终给予了极大支持和鼓励。此外，北京科技大学的刘文姗小姐和杜运炜先生负责了部分校对工作，张冬梅小姐肩负了本书的录入和排版任务，在此一并表示感谢!

作　者

1993 年 12 月于北京科技大学

# 目　　录

## 第一部分　基础教程

# 第二部分　高级教程

# 第一部分　基础教程

## 第一章　基础知识

在介绍 C 语言的语法结构和具体的程序设计方法之前，本章先从计算机的一些基础知识入手，简要介绍一些 C 语言的概况和它的内部构成方式，使读者对 C 语言有一个粗略的印象，为后续各章的学习打下基础。

### §1.1　微型计算机系统简介

人类在改造大自然的过程中，发明了许许多多的计算工具，如草绳计数，算盘，机械计算机，计算尺，手摇计算机，计算器，电子计算机等。这些计算工具为现代科技的发展建立了卓越功绩。在这些工具中，尤以电子计算机的功效最大，它已深入到人类生活的各个领域，成为现代科技发展的一个标志。未来的世界将是计算机的世界。

自从 1946 年第一台电子计算机"ENIAC"在美国诞生以来，电子计算机的发展已到了日新月异的空前地步，几乎每 5～8 年微处理器（CPU）的运算速度就提高十倍，而体积和成本却降低 10 倍以上。与早期的电子管、晶体管和中小规模集成电路计算机相比，由超大规模集成电路为核心元件的新一代计算机具有计算速度快，精度高，功耗低，功能强，存贮量大等特点。据报导，日本等发达国家正研制第五代智能型计算机，着力加强计算机的并行处理能力，并已经开发出了速度高达每秒进行几百亿次运算的高速计算机。因此，可以说，计算机技术的发展水平已成为检阅一个国家科技水平的指标之一。

#### 一、微型计算机系统的构成

微型计算机系统由硬件和软件两大部分所组成。硬件为组成计算机系统的一切设备或装置，又可分为主机和外设两组成部分。主机包括运算器、控制器和内部存储器（简称"内存"或"主存"）；外设通常指外存储器，输入输出设备。软件为管理计算机所有硬件资源，协调各部分正常工作的程序总称。根据其作用和应用面不同，我们可将软件进一步分成系统软件和应用软件两大类。操作系统、各种语言的处理程序、机器维护程序等通用性软件称为系统软件，其中以操作系统最为重要，是整个计算机系统的"灵魂"；那些为特定目的而编制的软件，如用户自己编制的专用程序包，就属于应用软件。微型计算机系统的一般组成可用图 1.1 来描述。

如果将一个微型计算机系统比做一个"人"的话，那么硬件就是它的"躯体"，软件便是"知识"和"血液"；主机、内存、运算器和控制器可分别看成"大脑"、"脑细胞"和"中枢神经"；显示器、外存储器、打印机、键盘和鼠标、扫描仪则可想象成"眼睛"、"记录纸"、"笔"和"五官"。

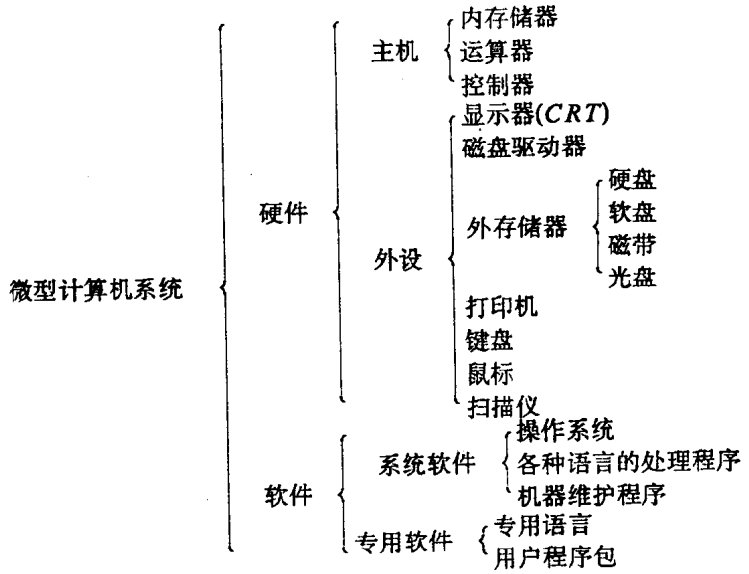


**图 1.1 微型计算机系统的一般组成**

目前国内流行的 IBM-PC 系列微型计算机系统及其兼容机如图 1.2 所示。

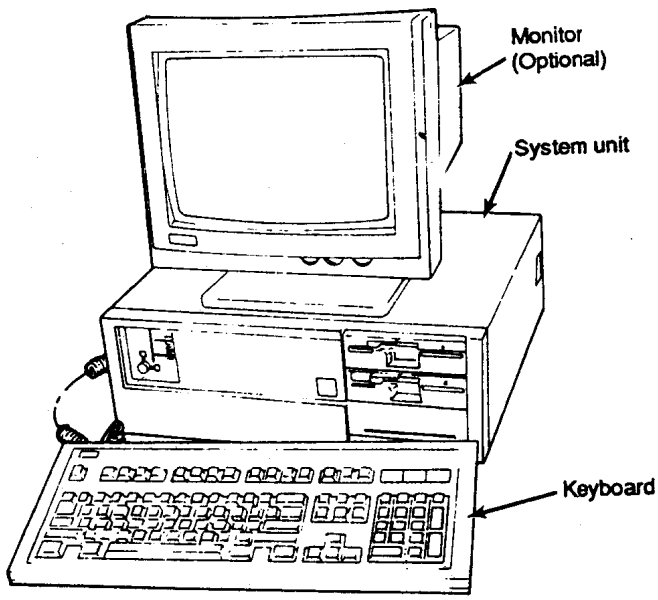


**图 1.2 IBM-PC 兼容机**

## 二、微型计算机的性能指标

目前国内市场上的微型计算机型号、机种较多，但以 IBM-PC 兼容机最为流行。如果要挑选一台微型机，怎样确认其性能的优劣呢？一般来说，衡量微机质量的指标有如下几个：

1.主机时钟频率（简称"主频"）：原则是越高越好，不过目前 286、386 档次的微机主频大都为 25MHz 或 33MHz（一句忠告：不要相信主机面板上的数字显示值！）

2.CPU 的字长，即 CPU 能直接操作的数据位数。通常所说的 8 位机，准 16 位机，16 位机，32 位机反映的就是这个指标，当然它越大越好。

3.显示器的分辨率，即显示器能分辨的点数（像素数目）和能同时显示的颜色数目。其分辨率的高低将决定用户日后使用时屏幕图象的色彩和逼真情况。通常有 MONO（单色显示器），CGA（彩色显示器）EGA（增强型彩色显示器）和 VGA（视频图象显示器）等型号。目前国内市场多出售 VGA 和 TVGA（Trident VGA），分辨率可达 640×480×256 色，甚至高达 1024×768×256 色。

4.内存容量：一般为 1MB，即 1024KB（1KB＝1024 字节）。经扩展可达 2MB，4MB，8MB，16MB，甚至更多。

5.硬盘容量：一般有 10MB，33MB，102MB，200MB 等几种典型硬盘。硬盘大小影响着磁盘信息的输入输出速度，因此需视实际情况，即主机频率和日常工作软件及磁盘数据量等决定选取适当大小的硬盘。

6.磁盘存取速度：尤其是硬盘存取速度至关重要，因为它是决定计算机输入输出的主要因素（一般应达每秒几十 K 字节以上）。

7.其它外设指标：如点阵打印机的点阵数（16 或 24），激光打印机、扫描仪的分辨率（通常有 300dpi —— 每英寸多少线，400dpi，1200dpi 等），鼠标器的按键数目及接触形式。

8.软硬件的兼容性：如果计算机的兼容性不好，则意味着其中的部件和元素无法与同类机进行互换或交流，许多工作必须从头单独做起。如软件设计时就无法保证足够基础软件。

除了上述因素之外，还应根据使用环境和工作性质决定是选用商业用机，还是高性能、高可靠性的工业用机或军用机；是否考虑机箱的插件槽数目、输入输出（I／O）端口数目以及电源负载能力等。总之，一条原则，要物尽其能，避免“大马拉小车”，造成浪费，要讲究性能价格比。

## §1.2 程序设计知识

### 一、计算机语言

人与人之间进行交流要用某种共同理解的语言，人与计算机进行交流当然也得有“语言”。程序员或操作人员通过按某种语言规范编制的程序才能控制计算机的工作，完成指定的任务。因此程序员必须事先掌握与计算机打交道的“计算机语言”。就象人类之间，交换的语言有汉语、英语、日语等多种文字或声音形式外，还有手语等其它形式一样，计算机语言也有许多种，各自在不同的应用场合使用。DBASE、FOX BASE 等语言用于数据库的管理，Z80、8086 等汇编语言多用于与硬件、接口打交道的场合，Fortran、Basic、Pascal 等常用于数据处理，C、PL／M 等“中级语言”则常用于系统软件、工程软件的设计……。

根据各语言的可读性、面向的对象以及与机器硬件的独立性等，可以将计算机语言分成：低级、中级、和高级三类（如图 1.3 所示）。

图 1.3 中各类语言的发展历程大致为：先有机器语言，后有汇编语言，再有高级语言和中级语言。机器语言是各类语言在计算机上储存的最终形式，也是一有计算机便有的最

早一种语言。它是由二进制编码所组成的，各命令和地址均用对应的二进制码表示。可见机器语言程序的编写非常繁琐，阅读困难，对程序员要求高。汇编语言正是为了克服机器语言的上述不足而诞生的，它用符号代替机器语言中的各条命令、地址，书写、阅读及记忆均更简便，但汇编语言程序必须把其中的符号还原成对应的二进制码，即经过编译("翻译")才能运行，并且仍要求程序员对机器系统的内部情况比较了解。典型的汇编语言有Z80，8086等。为了使程序脱离具体的机器硬件，将一般程序员从对机器内部结构的深入了解中解放出来，并使程序的编写接近日常的数学表达形式，计算机专家们便设计出了一系列的高级语言，如BASIC,FORTRAN,COBOL,PASCAL,DBASE等。当然，高级语言程序也必须经编译，转换成机器语言才可以运行。一切事情总是一分为二的，高级语言虽然保证了程序的可读性和直观性，编程工作也比汇编语言简便得多，程序的适应面较宽，但它使程序员对硬件的控制能力大为减弱，为工程软件和系统软件的设计带来许多困难，为此又诞生了一类既保留高级语言使用简便、可读性好等特点，又具备汇编语言程序强大硬件控制能力的"中级语言"，如C语言，PL／M语言等。

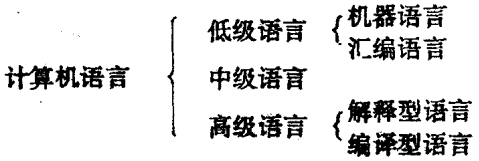


图1.3 计算机语言的分类

## 二、程序的运行

除机器语言之外，任何计算机语言的程序都必须事先经编译和连接转化为机器码才可以直接运行。需要提醒的是，各种语言源程序（非机器码）本身并不能直接运行，不是可执行文件，只有通过编译、连接工具将它们生成对应的exe或com文件后才能运行。exe和com文件才是可执行文件。可执行文件的运行方式为在当前DOS提示符状态下，直接键入文件名（不含扩展名）便可。如果某用户程序为C语言程序，文件名为f1.c，那么f1.c本身并不能直接运行，只有在C语言的工作环境下或经编译（生成目标文件f1.obj）和连接（将f1.obj与程序中调用的各库函数连接起来）生成可执行文件f1.exe或其它名字的exe、com文件才能运行。即：在DOS状态下键入

C＞f1 ＜CR＞①　（假设f1.exe在C盘上当前目录下）

便可（BASIC FORTRAN程序的执行过程与此类似）。

根据各种语言程序的编译和运行过程不同，可将它进一步分为解释型语言和编译型语言两种。解释型语言一般有一个工作环境，如普通BASIC就有BASIC工作环境；程序编写完毕，甚至程序编写过程中便可以逐条语句地边编译、边运行；这种工作方式对程序的调试较为有利，但程序的执行速度较慢，且不能处理大型程序。编译型语言，如Quicky Basic,Fortran,C,Pascal等，均采用先编译（检查语句的语法错误，并翻译成目标

① ＜CR＞—— 表示按回车键，下同。

代码，建立 * .obj 文件)，然后连接生成可执行文件 * .exe，最后再运行所生成的可执行文件的工作方式。这种工作方式具有执行速度快的优点，但也存在调试不够方便的缺点；幸运的是，目前许多编译型语言也象解释型语言一样配备了一个编辑和调试工具，使调试工作大为简便，Turbo C 便是其中一例。

## 三、程序的编写步骤

对于一个初级程序员和一个简单的问题来说，若要用计算机去解决，那么具体步骤大致为：

1.确定待解决问题的计算或处理方法。

2.将实际问题转变为一个数学问题，用数学模型或表达式描述。

3.编制程序框图，确定程序结构。

4.选择计算机语言和它的工作模式。

5.编制程序。

6.上机编辑、调试（包括编译和连接），消除语法错误。

7.如果结果正确则生成最终的用户程序，否则返回 1 从头重来，找出逻辑或设计错误所在。

上述步骤对于一个大型软件的设计来说就嫌不足，必须按软件工程的方法来设计。首先也必须确定待解决问题的计算或处理方法，将实际问题转换为数学问题，然后由系统分析员根据该问题的任务情况确定项目的实施计划，规定各阶段的任务和费用情况，决定各参与单位和参与人员的职责；接着提出项目的需求分析报告，确定项目的软、硬件环境，各题目的设计要求、数据结构；随后进行项目的概要设计，对项目的大致结构和数据传递即数据流进行设计，并完成项目中各模块任务及其关系的确立；在概要设计的基础上，对各模块的数据输入输出和流程进行详细设计，据此程序员便可立即编码，而不必了解项目的背景。在程序编码完成后，由程序员上机完成各模块的调试。为确保软件的可靠性和正确性，程序调试后还要先制订一个测试计划，并按测试计划利用测试工具进行测试，使语句覆盖率和分支覆盖率达到指定的数目，并保证各模块的功能正确，通过功能测试，最终通过系统测试，完成测试报告。至此，可以说软件设计的工作基本完成，但还有一个文档的整理工作待完成，它包括项目的使用说明书和项目开发总结等。总之，为了项目软件的可靠性，可移植性，可维护性和可扩充性，各阶段的文档工作必须齐全，并严格按规范进行。需特别指出的是，初学者在编程时必须注重所编程序的格式，保证可读性，并加入尽可能多的注释。读者若想了解有关软件工程方面的知识可参阅专门的书籍，如参考文献〔6〕。

## 四、程序质量

一个程序的设计好坏，要有一个标准。“简捷、快速、可靠、正确”的原则，虽从一个侧面提出了对程序的要求，但要真正衡量一个程序的设计质量应考虑的因素更多、更全，一般有：

1.功能齐全，正确，保证可靠性。这也是一个程序首先必须满足的。

2.运算速度快，即时间特性好。当然，程序的运算速度除了与程序设计思想和编码质

量有关外，还与所配置的硬件，如主机（主频），数学协处理器等有较大关系。这里所说的时间特性指在机器配置固定的情况程序的速度高低。

3.空间特性，即占内存和外存情况，代码是否精练（代码效率），是否对硬件有特殊的要求。不能说一个对硬件要求多的程序是一个好程序，同样也不能说一个无硬件要求的冗长程序或执行速度缓慢的程序一定是一个好程序，这要看项目的要求和用户覆盖面而定。

4.可移植性，指程序对计算机机型的适应性，是否仅作少量工作或无需额外转换便可以应用到与开发时所用机型不同的另一机种上。这往往取决于所用语言的性质。

5.易维护性。一个软件要保证绝对正确是很难的，因此便存在软件的维护问题。一个模块化的软件，若设计或开发文档齐全，源程序可读性好（格式规范，层次清楚、注释充足），对维护工作来说是极为有利的，否则要维护一个无任何文档、源程序格式杂乱无章的软件犹如"腾云驾雾"，不如另起炉灶。

6.可扩展性：一个软件在开发时所确定的功能可能满足不了日后该项目的新需求。处理对象的推广、使用环境的变更均要求对已有软件进行功能的扩展。因此，一个好的软件除了满足上述各项指标要求外，还必须为日后的扩展和改进留下良好的接口和简易的扩充方法。

## §1.3 C语言简介

### 一、C语言的发展过程

C语言的产生与发展和UNIX系统的产生与普及密切相关。可以说，C语言是为了描述和实现UNIX操作系统而设计的，而C语言的广泛流行又得益于UNIX操作系统的日益普及。在C语言出现之前，操作系统的编写语言主要为汇编语言，鉴于汇编语言的独立性、可读性和可移植性差，而一般的高级语言又难以完成对硬件的访问和进行位操作这样一种局面，操作系统的开发迫切希望一种既具有汇编语言特性，又具备高级语言特性的新一代计算机语言的问世。C语言就是在这样的背景应运而生的。

1960年出现了一种面向问题的高级语言ALGOL 60，它具有与硬件无关的特点，不宜用来编写系统软件。1963年英国剑桥大学的一批计算机专家们在ALGOL 60的基础上，改进了它与硬件脱离的不足，推出了CPL (Combined Programming Language)语言，但它的规模较大，难以使用。1967年英国剑桥大学的Martin Richards对CPL进行了简化处理，形成了BCPL(Basic CPL)语言。三年后，即1970年，在世界著名的贝尔实验室内有位名叫Ken Thompson的计算机设计专家对BCPL语言进行了进一步的简化，开发了一种硬件访问能力较强并且功能简单的B语言，并且用B语言写出了第一个UNIX操作系统（在PDP-7上通过）。第二年又在PDP-11/20机上开发了新版本的B语言，进而写出了相应版本的UNIX操作系统。虽然如此，但B语言毕竟存在功能过于简单，数据无类型等缺点，影响了许多软件功能的实现。为解决这个问题，贝尔实验室的D.M.Ritchie于1972年至1973年间推出了第一个C语言版本，使利用高级语言开发操作系统的设想成为可能。1973年，K.Thompson和D.M.Ritchie联手用C语言对他们在

1969 年用汇编语言写成的 UNIX 操作系统进行了改写，结果 90%以上的代码为 C 语言代码，而且发现其代码效率仅比汇编语言低 10～20%。

在随后的十年内，许多计算机软件公司，如 Microsoft,Borland 等，相继推出了自己的 C 语言版本，不断完善和丰富着 C 语言的功能。目前国内外流行较广的 C 语言版本有 Turbo C 2.0，Microsoft C 5.0,Borland C++ 3.0,Turbo C++等。综上所述，可总结出 C 语言的发展历程如下表所示。

| 语言名称 | 设 计 者 | 开发时间 |
|---|---|---|
| ALGOL | P.M.Woodward | 1960 |
| CPL | C.Strachey 等 | 1963 |
| BCPL | M.Richards | 1967 |
| B | K.Thompson | 1970 |
| C | D.M.Ritchie | 1972 |
| $C^{++}$ | R.Maseitti | 1983 |

## 二、C 语言的特点

C 语言与其它高级语言相比，具有如下优点:

1.能处理字符，数值和地址的内容。表达式&x 表示变量 x 的地址，*p 表示地址单元 p 中的内容。注意 C 语言是大小写敏感的语言，将同一字母的大小写看成不同的两个字符，并且通常以小写字母作标识符（C 语言内部定义的宏及外部变量及常量一般用由大写字母组成标识符来表示)。

2.能进行位操作，如按位与（&)、按位或（|)、移位(<<，>>）等。例如，a=1，b=7 时，a&b 则为 00000001B（二进制）或 1。

```
    00000001B
&   00000001B
-------------
    00000001B
```

3.语言简捷。如 a=a*3 可写成 a*=3（当标识符 a 较长或较复杂时，其优越性更为明显)。复合语句（语句组）的标识符为一双大括号——{ }（而 Pascal 语言为 BEGIN 和 END)。

4.数据类型比较丰富，而且自定义类型简便，灵活。除了有常规的整形(int)，浮点型(float 或 double)，字符型(char)外，还有结构(struct), 指针(*)，枚举(enum)，链表，位段等。既使是同一数据类型还可以根据数据在内存中的存放方式不同进一步细分为若干存储类型。例如，为了表示一个由姓名，年龄，性别，体重等多种类型数据组成的记录时，可以定义一个如下的结构（类型):

```
typedef   struct
          {
            char        *name;
```

```
    unsigned   year;
    char       sex;
    double     weight;
} record;
```

5.良好的模块化结构，使用灵活。一个 C 程序是由若干个函数所组成，每个函数是完成某一特定任务的模块；而一个大模块又可由若干个小模块所组成。只要函数的输入输出接口一致，便可灵活地修改其中的内容，并且可由多个程序员按函数接口要求并行编程，提高工作效率。

6.生成的目标代码效率高，通常只比汇编语言的代码效率低 10～20%。

7.书写格式自由。这里所说的"自由"是指每条语句的起始位置无任何特殊要求，不象 Fortran 等语言对每一位置均有严格规定。虽然如此，但这并不意味程序的编写不必注意格式；为了程序的可读性，必须对程序的格式进行规范，贯彻"层次清楚"的原则。如同一列上开始的语句属同一层次，复合语句中的语句组缩进到分隔符{ }之内，并且各功能块间空行并加入适当的注释。此外，在函数的开始处还应对函数的功能，输入输出参数，库文件使用情况等进行注释。

8.可移植性好。C 程序可简便地在 DOS，VAX，PDP，UNIX，AT&T 等机型上移植。

9.配置有结构化的语句。

a.有标准的控制语句代替随意的转移语句。

b.有表示层次结构的语句。

c.模块化的结构。

10.Turbo C 良好的用户界面。Turbo C 提供了一个集成工作环境，使程序编辑，编译和连接，调试工具等集为一体，并可通过下拉式菜单方便地设置工作环境，使程序员更加得心应手，工作效率大为提高。

事情总是一分为二的，正是由于 C 语言的众多优点，才引出了下列缺点：

1.运算符多，优先级不易记住，个别运算符含义多。C 语言共有 43 个运算符，且有几个运算符具有双重意义，如乘法运算符又是表示指针的运算符，表示取地址单元中内容的操作，地址操作符&又是按位与的运算符，小括号（ ）既是函数的参数传递通道，又是表达式的优先运算符，还是数据类型的强制转换符等（详见附录五）。

2.有些约定与常规不同，如位与（&），位或（|）的优先级较低，使用时应注意。

3.多重复合语句中各层次的区分符大括号"{ }"不醒目。

4.类型转换灵活，容易出错。

如前所述，上述缺点并不都是 C 语言的不足，有些"缺点"对初学者来说的确是一个缺憾，但对熟练人员来说却是优势无比的高效率表达方式。初学者往往爱将 C 语言与他（她）过去学过并熟悉的语言相比较，并不很愿意改变原来的一些习惯，所以觉得与过去习惯不同的一些 C 语言描述和工作方式"不好用"，是一个不足。如果的确如此，那么随着 C 语言的深入学习和逐步熟练，那些"不适感"相信会很快不复存在，并会喜欢 C 语言的。总之，初学时的一些困难或问题也许就是将来感到 C 语言优势所在的基点。

## 三、C语言中的标识符

C语言中允许出现的标识符有如下六类：

1.英文字母（'a'～'z'，'A'～'Z'）和下划线（_）。

2.阿拉伯数字：0～9。

3.运算符：+，－，*，／等43个（详见附录五）。

4.关键字（详见附录三）：

a.数据类型标识符（共有10个）：char，double，enum，float，int，long，short，struct，union，unsigned。

b.数据类型定义符：typedef。

c.存储方式标识符（共有4个）：auto，extern，register，static。

d.语句标识符（14个）：asm，break，case，continue，default，define，do，else，for，goto，if，return，switch，while，sizeof。

e.函数及数据类型修饰符：cdecl，const，far，huge，near，pascal，void，volatile。

5.特殊字符：\n，\t，\b，\r，\f，\\，%%，#等。

6.用户定义的标识符（包括C语言各头文件中定义的库函数名和宏定义）。用户定义标识符必须符合下面三条规则：

a.以英文字母（大、小写均可）或下划线中任一字符作首字符。

b.第一个字符之后可以是任意的字母，下划线，阿拉伯数字（0～9）所组成的字符序列。

c.不能是C语言中定义的各类关键字和宏定义，不能含有运算符和特殊字符。

注意：C语言各头文件中定义的库函数名和宏定义可参阅附录四，在此不一一列出。另外，若标识符中含有关键字或当前程序所引入头文件中定义的函数名和其它符号时，C语言的编译器在编译源程序时会指出"重复定义（redefined）"的错误信息，用户不必担心。

下面举一个标识符的定义用例，帮助理解。

**例1.1** 判断下述标识符的正确与否。

| | | | | |
|---|---|---|---|---|
| #300 | A__of__man | length__of__string | | |
| A>b | b*A | S(5) | NULL | EOF |
| FILE | x\n | a·b | Len1 | len1 |

在上述各标识符中，A__of__man，length__of__string，Len1，len1是合法的标识符（注意：Len1和len1表示不同的标识符）；a.b也是合法的标识符，虽然'.'为结构元素引用符，但a.b正好是结构a中元素b的描述方式；NULL，EOF，FILE均为C语言中定义的宏定义，不能作为用户标识符（NULL表示空指针，EOF为文件结束标记，FILE为文件类型符）；其它各标识符均是非法标识符：#300中第一个字符不是英文字母或下划线，A>b，b*A，x\n中含有运算符，a(5)表示对函数a进行求值（输入参数为5）。