Turbo C 使用大全

（1.5 —— 2.0） 第一册

徐金梧 刘冶钢
张思宇 仉伟敏
编译
北京科海培训中心

Turbo C 使用大全(V1.5~V2.0)

（第一册）

徐金梧　刘冶钢
张思宇　仉伟敏　编译

北京科海培训中心

编 译 者 序

奉献给读者的这套“Turbo C使用大全”是我们根据Herbert Schildt最新推出的新书“Turbo C—The complete Reference”和Borland公司最新的Turbo C 2.0版的“用户指南”和“参考手册”编译的。

Borland公司的Turbo C以它功能齐备的集成开发环境和无与伦比的编译速度在各种版本的C语言中占有十分重要的市场，是目前国内外在微机上运行的最为流行的C语言版本。

这套“Turbo C使用大全”系统全面地介绍了Turbo C集成开发环境、各种库函数的功能、各种语句的使用方法；使得C语言的各种指令、命令和库函数融为一体；并附有大量的应用实例，使读者在使用C语言时起到事半功倍的效果。在内容编排上层次清晰、通俗易懂、深入浅出，不仅适用于初次涉及C语言的编程者，而且对使用Turbo C语言编写大型，复杂的软件的程序员也是一本必备的工具书。

全书由六部分组成，共三十六章。

第一部分：介绍C语言的基本概念，基本语句以及一些基本函数。

第二部分：叙述Turbo C的集成开发环境。

第三部分：详细地解释了Turbo C1.5版的全部库函数，且对每个函数给出了使用的范例。

第四部分：介绍Turbo C的应用实例，包括怎样编写专家系统、数据库以及自己编写新的语言的内在奥秘。

第五部分：叙述了Turbo C进行软件开发的一些问题，包括与其它语言的接口、编写大型软件时应注意的问题。

第六部分：介绍了Turbo C最新版本2.0的增补部分。

在本书编译过程中得到了科海培训中心华根娣和夏非彼两位老师的悉心帮助和指引，谨向她们表示衷心感谢！

编 译 者

一九八九年十二月

目　录

第一部份　C语言

本书的第一部分将对Turbo C编程语言作详尽的介绍。第一章是C语言概述，有经验的编程者可以跳过这部份直接阅读第二章。第二章叙述Turbo C固有的数据类型、变量、运算符和表达式。第三章介绍程序控制语句。第四章介绍C语言的结构模块——函数。第五章描述数组。第六章讨论指针。第七章包括结构、联合、用户定义类型和枚举等内容；第八章介绍输入/输出（I/O）函数；第九章说明编译指令和代码生成。第十章讨论Turbo C的存储模式。第十一章，即第一部份的最后一章，介绍Turbo C的屏幕与图形功能函数。

第一章　C语言概述

本章概要地介绍Turbo C语言，包括它的起源，用途以及C语言编程的基本原理。

§1.1　C语言的起源

C语言由Dennis Ritchie设计，并首次在一台使用UNIX操作系统的DEC PDP-11的计算机上实现。C语言是由一种早期的编程语言BCPL发展演变而来的，BCPL语言目前仍在使用，主要是在欧洲。Martin Richards改进了BCPL语言，从而促进了Ken Thompson所设计的B语言的发展，最后导致了70年代C语言的问世。

多年来，在第五版本的UNIX操作系统下写成的C语言一直是事实上的标准版本。Brian Kernighan和Dennis Ritchie在C程序语言（The C Programming Language，Prentice-Hall，1978）一书中对此作了详尽的描述。随着微型计算机的日益普及，大量的C语言工具得以问世。在源程序水平上，这些工具程序几乎可以说是奇迹般地高度兼容。然而，由于没有统一的标准，这些工具之间也有不一致的地方。为了改变这种情况，ANSI于1983年初夏专门成立了一个委员会，为C语言制定了ANSI标准。Turbo C完全是按照ANSI的C语言标准实施的。它是一种快速、高效的编译程序。Turbo C不仅提供一个集成开发环境，同时还按传统方式提供了一个命令行编译程序版本，以满足不同用户的需要。

§1.2　C是中级语言

C语言常被称为中级计算机语言。这并不意味着C语言功能差、难以使用，或者不如象BASIC、Pascal这些高级语言那样完善，也不是说C语言类似于汇编语言，或涉及到使用汇编语言时所出现的有关问题。C语言定义为中级语言，只是意味着它把高级语言的基本结构与低级语言的实用性两者结合起来。表1-1说明了C语言及其它语言的属性。

作为一种中级语言，C语言允许对位、字节和地址的操作，这三者是计算机最基本工作

表1-1　　C在各种语言中的位置

高级语言	Ada
	Modula-2
	Pascal
	COBOL
	FORTRAN
	BASIC
中级语言	C
	FORTH
	宏汇编语言
低级语言	汇编语言

单元。C语言代码具有很好的可移植性。可移植性是指将一种计算机上的软件移植到另一种计算机的可能性。例如，如果在Apple Ⅱ + 上编写的程序能够容易地移植至IBM PC上，则该程序是可移植的。

所有的高级语言都支持数据类型的概念。数据类型定义了一类数值的集合，它们可以作为一个变量来存贮，程序通过一系列操作符实现对变量的各种操作。常用的数据类型有整型、字符型和实型。虽然C语言有五种固有的数据类型，但它不象Pascal和Ada语言那样有很强的类型区分。实际上，C语言允许几乎所有的数据类型转换，例如，在大部分表达式中，字符型和整型量都可以自由地混合使用。习惯上，C语言程序在运行时不做诸如数组边界检查，变量类型兼容性检查等错误检查。作这些检查是编程者的责任。

C语言具有允许直接对位、字节、字和指针进行操作的特点。所以它适用于作系统程序的基本语言，编制系统程序时，经常使用上述操作。C语言的另一个重要特征是它只有32个关键字（27个来自Kernighan和Ritchie的标准，另5个由ANSI标准委员会增补），它们构成了C语言的全部指令。（Turbo C多包含了11个关键字，用以实现各种增强和扩展功能）。作为一个对比，IBM PC上的BASIC语言的关键字，竟多达159个！

§1.3　C是结构语言

虽然严格地说“模块结构语言”这一术语并不适合于C语言。但是C语言通常还是被称为结构语言，因为其结构类似于ALGOL，Pascal和Modula-2。（从技术上说，模块结构语言允许在子程序或函数中再定义子程序或函数。这样，按照作用域规则，“全程”和“局部”的概念就扩展了，这一规则决定了一个变量或子程序的“可见性”。由于C语言不允许在函数中建立函数，所以它不是真正的模块结构语言。）

结构语言的一个显著特点是代码及数据的分隔化。分隔化是指一种语言的分段隔离能力，即程序的各个部分除了必要的信息交流外彼此互不影响，相互隔离。实现分隔化的一个方法是使用若干子程序，在子程序中分别定义各自的局部（暂时）变量。由于使用局部变量，编程者能保证其子程序运行时不会对程序的其它部分产生副作用。由于C语言具有结构语言的这些特点，程序之间很容易实现程序段的共享。如果你编制了一个分隔化的函数，仅仅需要知道它实现什么功能，而无须知道其功能是如何实现的。请注意：过多地使用全程变量（在整个程序中都可以使用的变量）可能会由于不必要的副作用，造成将错误漫延到整个程序。（每一位使用过BASIC的编程者都明白这一点！）

结构语言使编程者有多种选择的可能性，它直接支持若干种循环结构，例如while，do-while和for。在结构语言中，goto语句常被禁止使用或者建议尽量避免使用，它不是象在BASIC和FORTRAN语言中那样被作为一种常用的程序控制语句形式。结构语言允许编程者使用缩进书写形式编程，而不必象传统的FORTRAN语言那样有严格的书写区域限制。

下面是几种结构语言和非结构语言的例子。

结构语言	非结构语言
Pascal	FORTRAN
Ada	BASIC
C	COBOL
Modula-2	

结构语言趋于现代编程风格，而非结构语言较为陈旧。今天人们普遍认为结构语言层次清晰，比非结构语言更易于使用和维护。

C语言的主要结构成分是函数—C语言独立的子程序。在C语言中，函数是基本的结构模块，所有的程序活动内容都包含在其中。函数可以在程序中被定义完成独立的任务，独立地编译成目标代码，这样可以实现程序的模块化。函数建立后，你可以借助于它在各种情况下正确运行，而不必担心它对程序的其它部分产生不良影响。可以建立独立的函数这一点在大型软件工程中是非常关键的，因为只有这样才能避免不同编程者的程序由于偶然因素而互相干扰。

在C语言中，另一个实现程序结构化和分隔化的方法是使用复合语句。复合语句是被作为一个语句对待的，且具有逻辑联系的程序语句组合。在C语言中，只要将一系列语句置于一对花括号中间就构成了复合语句。在下面的例子中，如果x小于10，则执行if之后的花括号中两个语句。花括号内的这两个语句就构成了一个复合语句。它是一个逻辑单元，所以不可能只执行其中某一语句而不执行其它语句。在C语言中，任何一个语句既可以是一个简单语句，也可以是一个复合语句。复合语句不仅使得许多运算过程清晰、简洁和高效，而且还有助于表达程序的基本思想。

```
if (x<10) {
printf("too low, try again");
reset_counter(-1);
}
```

§1.44 C是编程者的语言

有人大概会对“C是编程者的语言”这一说法提出疑问：“难道还有不属于编程者的编程语言吗？”答案是完全肯定的。考虑一下两种非专业编程者的语言，COBOL和BASIC就知道，COBOL就是为使非专业编程者能阅读，甚至或许能读懂而设计的。而BASIC基本上是为非专业编程者设计用来编制计算机程序以解决一些相对简单的问题。

相比之下，C语言几乎可以说是独树一帜，它是由一些富有实际经验的专业编程者设计并在实际使用上不断完善和考验。最终的结果是C语言所能提供的恰好是编程者所要的：很少限制，很少缺陷，模块结构，彼此独立的函数和一些十分紧凑的关键字。使用C语言能达到接近汇编语言的效率，又兼有ALGOL或Modula-2语言的结构形式，这的确令人惊奇。

毫无疑问，C语言是最受第一流的专业编程者所欢迎的语言。

C语言常可用来代替汇编语言编程，这是它广为编程者所接受的主要原因之一。汇编语言使用助记符来表示计算机可直接执行的二进制代码。汇编语言的每个操作都对应于计算机的一项单一的任务。虽然汇编语言在编制程序时具有最大的灵活性、最高效率，但它却有着在编制和调试程序时较难使用的坏名声。此外，由于汇编语言是非结构语句，用它编制的程往往象意大利通心面条那样，错综复杂地缠结在一起，含有大量的跳转、子程序调用以及变址。由于这种结构的缺陷，使得汇编语言程序难以读懂、增强和维护。也许更为重要的是，汇编语言无法实现具有不同的中央处理器（CPU）的计算机之间的程序移植。

C语言在最初是被用作编制系统程序。系统程序是构成计算机操作系统或其实用支持工具的一大类程序的一部分。例如，下列程序通常被称为系统程序：

- 操作系统
- 解释程序
- 编辑程序
- 汇编程序
- 编译程序
- 数据库管理程序

随着C语言日益受到欢迎，许多编程者开始将所有的编程任务都用它来完成，因为它便于移植，有较高的效率。由于现在几乎所有的计算机都有C语言编译程序，在一种机器上编写和编译的程序只需作很小的改动甚至不用改动就可在另一种机器上运行。这种可移植性不仅节省时间而且节省费用。此外，C编译程序所产生的目标代码往往比大部分BASIC编译程序更为紧凑和快速。

C语言广泛用于各种类型的编程任务，其最主要原因也许是编程者喜欢用它！它具有汇编语言的速度，FORTH语言的可扩展性，但又很少有Pascal或Modula-2语言那样的严格限制。每一个C语言编程者都可以根据需要建立和维护属于自己的函数库，这一函数库可以用于许多不同的程序中。由于C语言允许，确切地说是提倡使用分块编译，因此使编程者易于管理大型软件工程，大大地减少了重复工作。

§1.5 编译程序与解释程序

编译程序与解释程序这两个术语是指实现程序运行的途径。从理论上说，任何编程语言既可以编译也可以解释，但很多语言通常只通过编译或只通过解释来运行。例如，BASIC语言常通过解释来运行，而C语言常通过编译实现运行。（然而最近流行一种C语言的解释程序，主要用作调试工具。）程序运行的方式并不取决于编写程序的语言。解释程序和编译程序只不过是一些较为复杂的运行程序，可以用于处理编程者的源程序。

解释程序一次只读一行源程序，并且执行该行源程序所指定的操作。而编译程序则读入整个源程序并将其转换为目标代码。目标代码是按照计算机能够直接运行的形式而实现的对源程序的一种翻译。目标代码也称为二进制码和机器码。一旦被编译后，源程序某一行在程序的运行中不再有具体含义。

当使用解释程序时，每次运行用户程序时都必须有解释程序存在。例如，在BASIC中，你必须先运行BASIC解释程序，然后才能装入你自己的程序，每运行一次都要键入RUN，

然后由BASIC解释程序逐行检查你编写的程序是否正确，若正确则运行。这一缓慢过程每当你的程序运行时就重复一次。与此不同的是，编译程序将你的整个源程序转换为目标代码，然后由计算机直接运行。由于编译程序只需将你的源程序翻译一次，你只需要直接运行你的程序，通常只要键入文件名即可。所以，编译只花了一遍的时间，而解释的方法在每次运行时都要花费重复的工作时间。

编译过程本身占用了一些额外的时间，但当你运行程序时这一点很容易被补偿。经过编译的程序比在解释环境下运行的程序快得多。唯一的例外是你的程序非常短，不到50行，而且没有循环语句。

在本书及C语言编译手册中，有两个术语经常出现，一个是编译状态，它是指在编译过程中所出现的事件；另一个是运行状态，它是指在程序实际运行过程中所出现的事件。这些术语在谈及错误时常用到，例如："编译时的错误"和"运行时的错误"。

§1.6 C语言程序的形式

表1-2列出了C语言的43个关键字，这些关键字再加上语法规则，就构成了C编程语言。

所有的C语言关键字都是小写的。在C语言中，大写和小写是有区别的：else是关键字，而ELSE则不是。关键字不能用作其它任何用途，例如它不能用作变量名，也不能用作函数名。

表1-2　　Turbo C的关键字

由ANSI标准推荐的32个关键字

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Turbo C扩展的关键字

asm	_cs	_ds	_es
_ss	cdecl	far	huge
interrupt	near	pascal	

所有的C语言程序都包含一个或多个函数。唯一不可缺少的函数是main( )，它是程序开始运行时第一个被调用的函数。在好的C语言程序中，main( )函数提纲性地列出程序所要做的事情，而这一提纲由一系列函数调用所组成。虽然从技术上说main一词并不是C语言的组成部分，但还是应当把它当作关键词来对待。不要将main用作变量名，因为这样有可能使编译程序产生混淆。

C语言程序的一般形式如图1-1所示，其中f1( )至fN( )代表用户定义的函数。

```
全程变量说明
main ( )
{
 局部变量
 程序段
}
f1 ( )
{
 局部变量
 程序段
}
f2 ( )
{
 局部变量
 程序段
}
;
fn ( )
{
 局部变量
 程序段
}
```

图1-1 C语言程序的一般形式

§1.7 库函数和连接

从技术上说，完全可以编制一个完全由编程者所写的实用的C语言程序，然而实际上很少这样做，因为在实际所定义的C语言中，并没有提供输入/输出（I/O）操作方法。所以，大部分程序都包含对Turbo C标准库中各种函数的调用。

Turbo C提供一个标准库，其中包含许多完成大部分常规功能所需的库函数。当程序中调用一个未曾定义的函数时，Turbo C“记住”这一函数名，然后由连续程序将该程序与偏准库中该函数的目标代码连接起来。这一过程叫做连接。

在标准库中，标准函数是以可浮动地址的形式保存的。这就意味着，函数的机器码指令的内存地址并没有绝对确定，而只是保存了一个偏移量。当你的程序与标准库函数连接时，这一偏移量被用来产生实际的内存地址（绝对地址）。有几种技术手册和参考书详细地解释了上述过程。然而对于使用Turbo C的编程者来说，没有必要对这一内存地址重定位过程有更深的了解。

§1.8 分块编译

大部分较短的C语言程序完全可以包含在一个源文件里。然而，随着程序变长，编译时间也就长了。而长时间的编译过程又容易使人烦躁难耐。所以，Turbo C允许把程序分割成许多块，分别装进不同的文件里，每个文件可以单独编译。每个文件都编译完以后，就可以将它们连接在一起，包括对库函数的连接，从而形成一个可以运行的目标代码文件。分块编译的优点在于修改一个源文件中的程序后，并不需要把整个程序的所有文件重新编译一