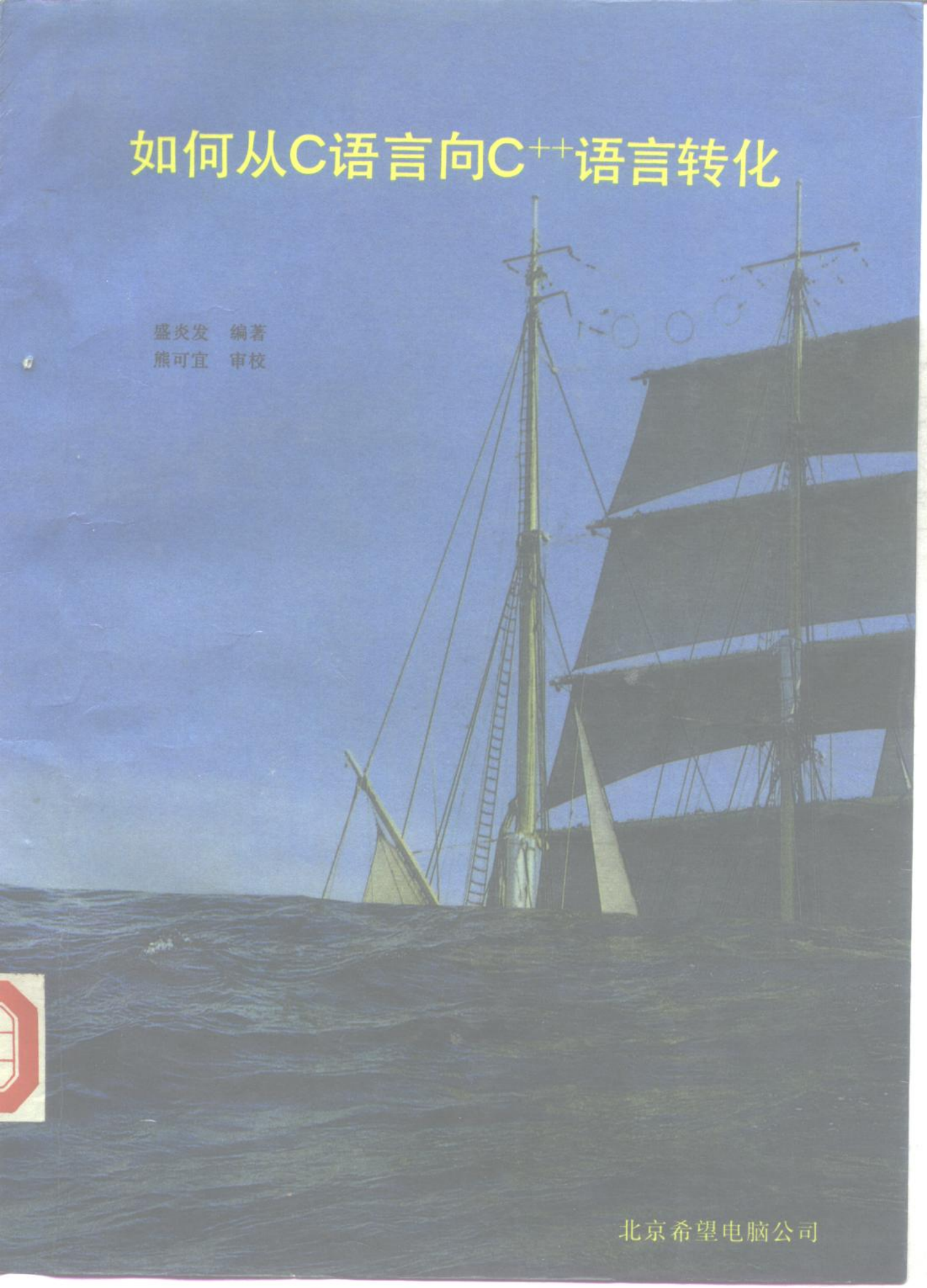
如何从C语言向C++语言转化

盛炎发　编著
熊可宜　审校



北京希望电脑公司

如何从 C 语言向 C++语言转化

盛炎发　　编著
熊可宜　　审校

北京希望电脑公司

内 容 简 介

本书介绍了C语言和C++的基础知识及其相互联系，使读者循环渐进地学习C语言进而掌握C++这种功能强大的程序设计语言，本书既可以作为初学C语和C++语言的入门教材，又可作为有一定经验的程序员的参考手册.

需要本书的用户，请直接与北京8721信箱联系，电话：2562329，邮政编码：100080.

如何从C语言向C++语言转化

战东发　　编著
熊可宜　　审校

北京希望电脑公司　　施园印刷厂印刷
开本：787×1092　1/16　印张：19.81　字数：456千字
1993年4月第一版　　1993年4月第一次印刷
印数：1—5000册
京准印字：3566—91566　定价：15.00 元

目　录

此为试读,需要完整PDF请访问: www.ertongbook.com

第一部分　C 语言介绍

第一章　C语言简介

本章介绍:

C语言的起源
C语言是一种中级语言
C语言是一种结构化语言
C语言是一种程序设计员使用的语言
编译程序和解释程序
C语言的格式
库函数及联接
分段式的编译
编译一个C语言程序
C语言的内存配置

本章的目的是要明确表达出C语言的基本概念，包括它的起源和用法，适合于初学者学习，若读者已有这方面的知识，可直接跳过这一章。

1.1　C语言的起源

C语言最先是由Dennis Ritchie在DEC公司的PDP-11的机器上设计发展出来的，它的操作系统正是目前最流行的UNIX的早期版本，而C语言发展至今主要是由当时的一种语言BCPL演变而来的，这种语言目前在欧洲仍然被广泛使用。BCPL是由Martin Richards所设计，它同时也对B语言产生重大的影响。而C语言就是在七十年代从B语言演变得来的。

经过许多年的发展，C语言已形成了一套满足UNIX第五版操作系统要求的标准。详细内容参见Brain Kernigham & Dennis Ritchie所著的<<The C Programming Language>>一书，该书被认为是C的经典。随着微电脑的普及与流行，C语言已相当广泛被使用，用来作为软件设计的工具。这些软件具有相当好的程序代码，可在不同的机器上使用，但却没有一定的标准格式。为了解决这个问题，ANSI（美国国家标准局）在1983年制定了一套永久的标准。该标准也兼容了以往的格式。所以大部分的C编译程序都是以ANSI为标准的，本书也同样是以ANSIC作为蓝本，换句话说，它可适用于任何版本的C编译程序。

1.2　C是哪一级的语言?

C通常被称为一种中级语言，但这并不表示C语言的功能不强，很难用，或是在语言的设计思想上比BASIC，PASCAL等高级语言差。同时也不表示C语言和汇编语言相

此为试读,需要完整PDF请访问: www.ertongbook.com

类似．C语言之所以被称为中级语言，主要是由于C语言具备了低级语言的按地址存取功能和高级语言的函数调用功能．见表1-1:

表1-1　C语言在各种语言中的地位

高级语言	Ada
	Pascal
	COBOL
	FORTRAN
	Prolog
	BASIC
中级语言	C
	Macro　Assembler
低级语言	Assembly

C语言是中级语言，它可直接对计算机的基本数据单位BIT，BYTE和　ADDRESS等进行运算处理．C语言程序具有可移植性．所谓可移植性就是指同一软件可在不同的机器上运行．例如，一个软件是在APPLE MAC上开发的，并且可以在IBM　PC上运行，那么这个软件就具有可移植性．

所有的高级语言都具有各种复杂的数据类型．一般的数据类型可分为整数，字符和浮点数．C语言具有五种数据类型，并允许大部分数据类型间的转换．例如，你可以在一个表达式中混合使用各种不同的数据类型．

C语言的特点就是它能直接处理BITS，BYTES，WORD和POINTER，所以它很适合用于系统软件．另外一个特点是C语言的关键字（KEYWORD）只有32个，其中27个来源于早期的K&R版本，加上另外的5个ANSI版本的关键字．C语言就是由这些关键字所组成．作为对比，在IBM　PC上的BASIC的关键字多达159个，相比之下，C语言就比较简洁．

1.3　C语言是结构化语言?

虽然严格来说C语言不是分段结构语言，但是我们通常都把它视为结构化语言，因为它有一些特点很象其它高级语言．严格来说，分段结构化语言允许子程序（PROCEDURE）或函数（FUNCTION）在另外一个子程序或函数中定义．而C语言不允许在一个函数中再建立一个函数，所以我们不把C语言称为是一种分段结构化语言．

结构化语言最大的特点是将程序代码和数据划分开来．但是如何实现呢？我们可在一个子程序中使用一些局部变量（LOCAL　VARIABLE），而这些变量并不影响其它部分的程序，所以你可以容易地共享程序代码．如果你在使用这项功能时，你只需要了解它在作什么，并不需要了解它实际如何工作．但要记住一点，使用过多的全程变量(GLOGAL　VARIABLE)，会引起程序产生不必要的错误．为什么?

结构化语言加强了程序的流程功能，它提供了许多的循环语句（LOOP），例如

while, do while 和 for. 而在结构化的要求中，goto 语句是被禁止或尽量避免使用的.

以往的电脑语言不是结构语言，随着语言结构化的发展趋势，今日的电脑程序设计员更偏好使用结构化语言，原因无它———易于设计和易于维护. C 语言的主要结构成份是独立的子程序，而且它们可以在一个程序中进行分别定义和编程，以使程序具有模块化(Modular) 特性，因而你可以建立一个函数或子程序，使得它的执行不会影响其它部分的程序.

在 C 语言中你可以将两个大括号之间的语句作为复合语句进行编程，例如:

```
if( X < 50 )
 {
    printf("too low, try again\n");
    scanf("%d",&x);
  }
```

在大括号之间的两行语句，只有在 X 大于 50 时它们才会被执行到，在逻辑上它们是在一起的，不能单独地执行. 重要的一点是，C 语言的表达式可以为单独的一条语句或是由数条语句组合成的一个复合语句.

1.4 C 语言的使用者

令人惊讶的是，并不是所有的电脑语言都是为程序设计员设计的. 举例来说，COBOL 语言并不完全适合于程序设计员，因为它是给那些非电脑程序员们使用的，他们能从中了解这个程序在干什么. 而 BASIC 语言则是提供给那些业余爱好者，使他们能用电脑来处理相当简单的问题.

相对而言，C 语言能提供给那些电脑程序员们所需要的———限制少、分段结构化、独立的函数和简洁的语句. 当你使用 C 语言时，你就知道它的功能强大，可以轻易地与汇编程序代码结合在一起使用，这就是为什么 C 语言受到专业程序设计员的垂青. 如今你也可以用 C 语言来达到汇编语言的功能，而事实上汇编语言正是利用一些符号来替代一些电脑可直接执行的二进制机器码的. 有经验的人都知道，如果要对一个汇编程序进行扩充或改错，是一件很不容易的事，有时还不如重写一遍更方便. 除此之外，因为它不是结构化语言，所以在程序中必有许多的 jump,call,和 index，这就造成了程序可读性差，不易维护，更重要的一点是它的不可移植性，因为在不同的中央处理单元 (CPU) 间，汇编语言的可执行性也同时降低.

目前 C 语言愈来愈流行和愈来愈受欢迎，许多程序设计员就因为它的可移植性和高效率而改采用它，即它可先在甲机器上设计，然后再移至乙机器上使用，只需进行少量的修改甚至完全不必修改，这样即省钱又省时. 此外它的目标代码比用 BASIC 编译程序编译后产生的目标代码快许多.

除此之外，程序员们都喜欢用它来开发系统程序和商业化软件，因为它的执行速度和汇编语言相差无几，也不象 Pascal，BASIC 有众多限制. 另外你也可以自己编写一个程序库供日后其它程序使用，分段编译，因而易于开发较大的系统，可节省许多重复的工作.

1.5 编译程序和解释程序的区别:

编译程序（COMPILER）和解释程序（INTERPRETER）都是使程序转变成计算机可执行的机器代码。从理论上说，任何程序语言都可以被编译或被解释，然而通常只采用其中一种，C 语言是采用编译式的。

解释程序是一次一行的读取源程序，然后执行这一行的指令。而编译程序却是一次读取整个程序，然后将其转化成目标代码（OBJECT CODE），也就是说编译程序把源程序转化成电脑可执行的二进制码或机器码。一旦程序被编译过后，对程序的执行过程来说源程序已经没有任何意义了。

每当你要执行你的程序，若你采用的是解释程序时，你就得每次调用解释程序来执行，例如：在使用 IBM PC 的 BASIC 时，你得先执行一次解释程序，将其调入内存，再装载程序部分，最后再键入 RUN 去执行。这样一来，执行的速度自然比不上经过编译程序编译过的目标文件，因为解释程序是每次一行地去检查语法，然后再执行，所以相对速度就减慢了许多。而编译程序的过程则不一样，它是把源程序转化为电脑可直接执行的目标文件，以后只要键入文件名就可执行了，不必象解释方式那样每执行一次就调用一次，白白增加了许多额外的负担。

1.6 C 语言的格式:

表 1-2 列出了 C 语言的关键字，而这 32 个关键字就是组成 C 语言的成份。其中有 27 个是原来就有的，另外 5 个为 enum、const、signed、void 和 volatile，是 ANSI 委员会后来加上去的。

表 1-2 ANSIC 的关键字

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

除此之外，许多 C 语言的编译程序都加上了一些额外的关键字，主要是想尽量利用 80× 86 系列微处理器的内存组织结构，同时还提供了内部语言（INTER-LANGUAGE）和中断的功能（INTERRUPT）。也许你的编译程序还提供了其它更好的功能，那么就得看你如何充分利用它了。

注意！所有的 C 语言的关键字都是小写的，因为在 C 语言中大写和小写是不同的，例如 else 是一个关键字，而 ELSE 就不是了，这一点读者要小心！否则 C 语言会把它当成是一个变量。在 C 程序中最少要有一个或一个以上的函数，最基本的是 main() 函

数，而且要放在程序的前端。就技术上而言，main() 不是 C 语言的一部分，但我们就把它当成是吧!

1.7 库函数及联接

就技术的观点来说，你可以仅用这些语句来编写一个功能强大的程序，但是这并不实际，因为 C 语言并不能只用这些指令进行输入输出 (I／O)，所以大多数的程序都去调用 C 语言的标准库函数来达到目的。

所有 C 语言的编译程序都有一个标准的函数库，而它就是用来满足通常的应用需要的。也许你的函数库还包含了很多其它的函数，不包含在 ANSI C 标准程序库内，例如绘图的功能 (GRAPHIC)，尽管如此，它仍然被认为是 C 语言的函数库。

C 语言的函数库有的是被设计成放在一个大的文件中，有些则是放在许多小的文件里，但都是为了要提高实际使用的效率。本书中采用的是以单一文件类型的函数库为主。

目前所使用的 C 语言编译程序，几乎都是应用于一般目的的。所以当调用一个函数时，这个函数并不是程序的一部分，于是编译程序就会记住这个函数名称。等到后来的联接程序把它和目标码联接在一起时，自然会到函数库中找出名称相同的库函数，这个过程就叫联接 linging。许多 C 语言编译程序都有自己的联接程序，也有一些是由操作系统本身提供的，这就要看是使用何种编译程序与系统了，例如 MS-DOS 的 link·exe。

库函数都是可重定位的 (RELOCATABLE)，这也意味着机器码在内存的地址并不是绝对的 (ABSOLUTE)。所以当你的程序和标准库函数联接在一起后，它实际上只定出实际使用时的相对地址。

C 语言还提供了另一种特别的功能，当在编写一个程序时，必定会用到一些标准的库函数，此外你也可以写一个自己会常用到的函数，把它放到函数库里，以便日后直接调用，而不必重复输入这一个函数。

```
global declarations
return-type main(parameter list)
{
 statment sequence
}

return-type f1(parameter list)
{
 statment sequence
}

return-type f2(parameter list)
{
  statment sequence
}
```

```
.
.
.
.
return-type fn(parameter list)
{
 statment sequence
}
```

图 1-1　一般 C 语言程序的格式

1.8　分段式的编译

大多数较短的 C 语言程序都只有一个源文件，然而当一个程序变得越来越大时，编译所需的时间也越来越长了，设计者也渐渐没耐心等待电脑检查语法的正确与否．因此 C 语言提供了可将一个大程序分解为由几个小程序组合而成的功能，从而可以分别编译每个小程序，而不必每次去重复编译整个程序．其优点是当你想修改程序中的一小部分时，你不必去编译那些没有被修改的部分，这样就节省了不少时间，有关详细的内容会在本书的后面章节中介绍．

1.9　编写 C 语言程序的步骤:

将一个源程序变为可执行文件通常需以下三个步骤:

1.建立源程序

2.编译源程序

3.联接库函数

许多 C 语言的编译程序提供有集成环境，例如 TURBO　C 的集成环境．有些就要使用别的编辑程序来编辑你的程序，但是要注意一点，这个编辑程序产生的字符文件必需是标准的字符文件，否则你用的文字处理程序可能会产生一些可见或不可见的特殊控制符，使 C 语言的编译程序无法编译．详细内容可参见相应的用户手册．

1.10　C 语言的内存配置:

一个被编译过的 C 语言程序，将生成并使用内存的四个不同区域，而且各有各的功能．其中第一个区域用来存放程序代码的，第二个区域用来存放全程变量 (GLOBAL　VARIABLE) 的，剩下的两个一个用作为栈 (STACK)，另一个被用作为堆 (HEAP)，栈是用来存放调用子程序或函数后的返回地址 (RETURN ADDRESS)，函数的参数变量 (ARGMENT) 与局部变量 (LOCAL　VARIABLE)，同时还存放 CPU 的当前状态值 (STATE)．而堆其实是一块空白的内存区域，是提供给 C 语言进行 LINK　LIST，TREE 之类功能的使用的，这四个区域具体的内存定位则与 CPU 的种类有关．

第二章　C语言的表达式

本章介绍:

5种基本的数据类型
基本类型的转换
变量
局部变量
形式参数
全程变量
特殊的存储类型
静态变量
寄存器变量
常量
运算符
表达式

2.1　5种基本的数据类型:

在C语言中有五种自动存储类型: 字符 (CHARACTER), 整数 (INTEGER), 浮点数 (FLOATING-POINT), 双精度数 (DOUBLE FLOATING-POINT) 和无返回 (VOID) 等五种类型, 分别用 char, int, float, double 与 void 来表示. 这些数据类型的大小范围与CPU及编译程序的类型有直接的关系. 通常以一个字节 (BYTE) 作为存储的基本单位, 整数的长度是两个字节 (2 BYTES), 但不要认为其它的计算机也是如此, 而将程序转换过去使用. ANSI标准只规定了所采用的最小的值.

浮点数的正确格式就要按其设计出来的方式来决定, 而整数通常表示为一个字 (WORD) 长. 字符则通常用ASCII码来表示, 多少个字符就使用多少个字节.

浮点数和双精度数比较注重数字的精确度, 一般采用两个字节, 以提供较高的精确度. ANSI标准的浮点数值的范围较小, 约在1E-37和1E+37之间.

无返回类型是用在调用函数不返回任何值的场合, 有关详细说明将在以后的章节中提到.

表2-1给出了ANSI标准的最小范围和大概占的字节数. 把signed用在整数类型上是多余的, 但若你使用了也不会造成错误, 因为C语言会把错误的整型变量说明当成是有正负号的.

此为试读,需要完整PDF请访问: www.ertongbook.com