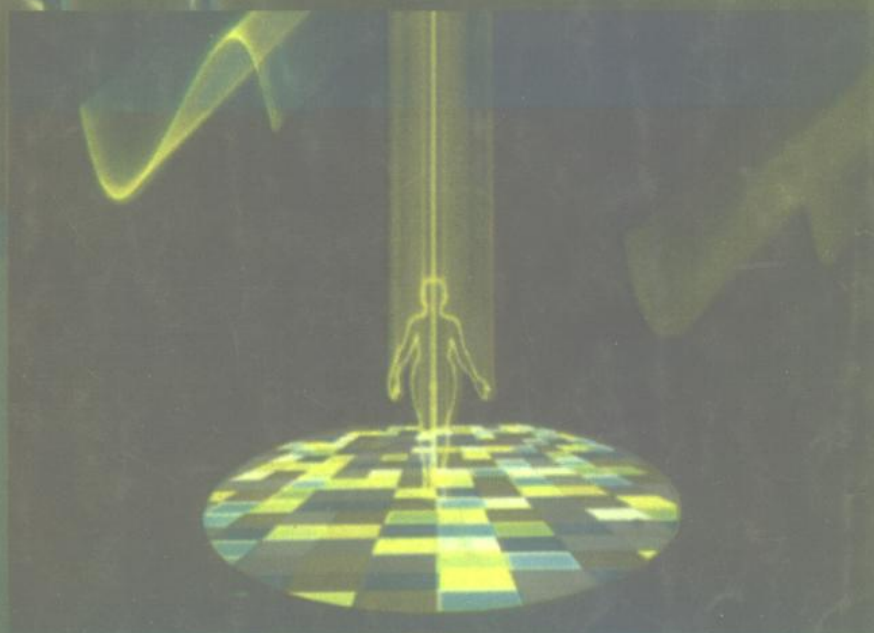


C语言快速入门

王亚军 李超琼 编著



中国致公出版社

12
J·Y1

C 语言快速入门

王亚军 李超琼 编著



中国致公出版社

336335

图书在版编目(CIP)数据

C语言快速入门 王亚军,李超琼著,北京:中国致公出版社,1996.2

ISBN7-80096-014-5

I. T… I. ①王…②李… II. C语言. Turbo-程序语言基本知识 N. TP321C

中国版本图书馆CIP数据核字(95)第06311号

C语言快速入门

王亚军 李超琼 编著

* * *

中国致公出版社出版发行

北京市西城区太平桥大街4号(邮编:100034)

新华书店经销 河北省景县印刷厂印刷

开本:787×1092 1/16 印张:17.125 字数:270千字

1996年2月第1版 1996年2月第1次印刷

印数:1~10010册

ISBN 7-80096-014-5/TP·12

定价:18.00元

目 录

第一章 C语言的历史与发展	1
1.1 C语言的历史	1
1.2 C语言与其他程序设计语言的比较	1
1.3 编译程序的使用	4
1.4 PC上C程序的内部结构(选择性的内存模式)	5
1.5 结论	8
第二章 Turbo C语言的集成环境	9
2.1 Turbo C的安装	9
2.2 Turbo C的执行	10
2.3 主菜单	11
2.4 编辑窗口	14
2.5 信息窗口	14
2.6 Turbo C的功能键	15
2.7 TCINST程序	16
第三章 Turbo C的编辑器	20
3.1 编辑指令	20
3.2 调用编辑器并且输入文字	20
3.3 字符、字和行的删除	23
3.4 整块文字的移动、拷贝和删除	23
3.5 移动光标	24
3.6 查找以及查找替换	25
3.7 存储及调用文件	26
3.8 自动缩排	27
3.9 块输入和输出	28
3.10 其他若干指令	28
3.11 编辑的综述	29
3.12 Turbo C文件的命名	30
第四章 变量、常量、运算符和表达式	31
4.1 标识符	31
4.2 数据类型	31
4.3 常量	33
4.4 变量	34
4.5 运算符	35
4.6 表达式	40

4.7	程序基本结构	42
第五章	程序控制语句	44
5.1	if 语句	44
5.2	switch 语句	45
5.3	for 语句	48
5.4	while 循环	50
5.5	do/while 循环	51
5.6	break 和 continue 语句	52
5.7	区段语句	52
5.8	return 语句	52
5.9	goto 语句和语句标记	52
第六章	数组和字符串	55
6.1	数组	55
6.2	字符串	69
第七章	指针	75
7.1	指针是地址	75
7.2	指针变量	75
7.3	指针运算符	76
7.4	指针表达式	77
7.5	动态配置函数	79
7.6	指针和数组	80
7.7	指向指针的指针	84
7.8	设定指针的初始值	85
7.9	指向函数的指针	86
7.10	指针并不是整数	88
7.11	指针的相关问题	89
第八章	函数	91
8.1	前言	91
8.2	函数的一般格式	91
8.3	return 语句	91
8.4	函数的范例规则	94
8.5	函数的参数	100
8.6	argc 和 argv	105
8.7	返回非整数数值的函数	107
8.8	函数的标准类型	110
8.9	递归调用	111
8.10	编写函数时的关键	112
第九章	输入和输出	114
9.1	前言	114

9.2	两个预处理器的编译指示	114
9.3	stdio.h 头文件	116
9.4	数据流和文件	116
9.5	数据流	117
9.6	文件	117
9.7	观点和实际	118
9.8	控制台的输入输出	118
9.9	控制台的格式化输入输出	120
9.10	缓冲输入输出系统	124
9.11	非缓冲输入输出系统	138
9.12	选择一种方式	144
第十章	高层次的数据结构	145
10.1	前言	145
10.2	存取修饰符	145
10.3	存储修饰符	147
10.4	设定语句中的类型转换	152
10.5	函数类型修饰符	153
10.6	函数的指针	154
第十一章	用户定义的数据类型	157
11.1	前言	157
11.2	结构	157
11.3	结构数组	159
11.4	把结构传递给函数	167
11.5	指向结构的指针	169
11.6	结构中的数组以及结构	173
11.7	位字段	173
11.8	union(联合)	175
11.9	枚举类型	179
11.10	使用 sizeof 来达到程序转移	181
11.11	保留字 typedef	182
第十二章	高层次的运算符	184
12.1	前言	184
12.2	位运算符	184
12.3	运算符?	191
12.4	C 语言的简写格式	192
12.5	逗号运算符	192
12.6	小括号和中括号	193
12.7	运算次序	193
第十三章	Turbo C 的预处理器和编译选择	194

13.1	前言	194
13.2	Turbo C 的预处理器	194
13.3	编译指示 #define	194
13.4	编译指示 #error	196
13.5	编译指示 #include	196
13.6	条件编译指示	197
13.7	编译指示 #undef	200
13.8	编译指示 #line	200
13.9	编译指示 #pragma	201
13.10	预先定义的宏指令名字	202
13.11	编译选择功能和连接功能	203
13.12	Compiler 选择功能	203
13.13	连接选择功能	206
13.14	Turbo C 集成环境的选择功能	207
13.15	Args	208
13.16	Retrieve option 和 Store option	208
第十四章	Turbo C 常用到的库函数	210
14.1	前言	210
14.2	字符串和字符函数	210
14.3	算术函数	219
14.4	操作系统相关的函数	225
14.5	动态配置的函数	231
14.6	其它函数	235
第十五章	重要课题补充	245
15.1	前言	245
15.2	多文件程序的编译	245
15.3	命令行版本的 Turbo C	248
15.4	Make 用法的简介	251
15.5	一些常见的程序错误	253
附录 A	C 的错误信息	260
附录 B	ASCII 码表	262
附录 C	ANSI C 关键字	267

第一章 C语言的历史与发展

C 程序语言是系统程序设计的一般性用途语言,除了应用在系统软件的开发,如操作系统、编译程序、编辑程序外,也被应用在其他方面,如数据库软件、电子表格程序、工程应用和字处理软件,是目前流行的 UNIX 操作系统的主要语言,而且可在许多机器上使用如 Apple 电脑, Digital Equipment 公司的便携计算机, IBM 及 Cray 的大型计算机。

1.1 C语言的历史

C 程序设计语言在 1972 年由 Dennis Ritchie 在 AT&T 贝尔实验室开发。它的基础可溯及 Martin Richards 在英国剑桥开发的 BCPL 程序设计语言(Basic Combined Programming Language)。1970 年,贝尔实验室的计算机科学家们在 Digital Equipment 的 PDP-11 机器上。开发出一种修改 BCPL 后的程序设计语言,称为 B 语言,它就是早期使用的 UNIX 系统。B 语言是缺少数据类型的语言,它只有一种数据类型——机器字。这种机器的每一个字都能以“指针”变量存取。这种单一数据类型观点对此机器并不适合,因为 PDP-11(如 IBM-PC)不但可以用字节寻址,而且具有算术及浮点数运算的功能。B 语言缺乏字节寻址的能力,也缺乏区别整数及浮点数的能力,因此它不能同时提供整数运算及浮点数运算的功能。然而这些缺点促成了 B 语言转变为 C 语言,B 语言与 C 语言的主要不同在于:C 语言具有数据结构。在 C 语言中,每一个数据的定义或说明都详述了一个数据类型的内容,它定义了这种数据的类型需要多少内存,及如何解释这种数据类型的值。数据类型包含字节寻址的字符、整数和浮点数。C 语言保留了指针变量的概念,并提供了一些更高级的数据结构,如数组和结构。下一节将以其他流行的程序设计语言来说明 C 语言的特色。

1.2 C语言与其他程序设计语言的比较

许多语言都可以在 PC 上使用,它们包含汇编语言、BASIC、FORTRAN、COBOL、PL/1、PASCAL 及 C 语言,为了将 C 语言与这些语言进行比较,首先将探讨开发这些语言的目的。

汇编语言

汇编语言是最接近微处理器指令的语言。虽然程序员可以通过汇编语言而使用机器的所有功能,但它在表达算法(问题解法)时,并不是一个简单自然的语言。例如,表达式 $X=Y+Z$,以汇编语言编写的程序如下:

MOV AX,Y	;将 Y 装入寄存器 AX
MOV BX,Z	;将 Z 装入寄存器 BX
ADD AX,BX	;将 BX 寄存器的值加到寄存器 AX(Y+Z 值存入 Y)
MOV X,AX	;将结果存入 X

程序员称这种语言为“低级语言”，因为他必须以最基本的机器语言来编写程序。

汇编语言已成为开发系统软件的主要语言，如操作系统(DOS是以汇编语言编写)、编译程序以及连接编辑程序，因为开发这种软件时常需要复杂的数据操作(需要内存直接寻址)，或其他与机器类型有关的功能。由于每一种微机的汇编语言是独特且相异的(8088的汇编语言和Apple电脑的汇编语言不同)，所以用汇编语言写的系统软件便无法在不同的计算机上执行。每一个计算机厂商都使用不同的汇编语言来开发他们自己的操作系统，这就是为什么没有一个标准操作系统的原因。

以应用为主的计算机用户，如学生、科学家、工程师和企业家，很快就了解到他们需要更简易自然的计算机语言来表达他们的算法，也因此引发了“高级语言”的开发。使用高级语言必须有编译程序(或解释程序)。它把高级语言的语句转换成微处理器的机器语言。接下来要讨论最常用的高级语言。

FORTRAN

FORTRAN语言的用户认为计算机是数字处理的机器，它可以简单自然地表达数学的算法。FORTRAN语言开发于1950年，其间经过多次改善与扩充，现已广泛地为科学家、工程师们使用。然而，它并不适用于开发系统软件。

由于FORTRAN是为了要把计算机当作数字处理机使用而设计的，因此，它既没有详细描述数据结构的功能，也缺乏内存直接寻址的能力，它不具备一些汇编语言的运算功能(如位运算)，也不具备数据结构(除数组外)，更没有某些现代程序设计技巧的功能，如递归程序设计。用FORTRAN写大型程序时，可以把大问题分成许多小部分(叫子程序或函数)，先独立地设计这些程序，然后再把它们合在一起，成为最后的产品。

COBOL

COBOL的用户认为计算机是一部商业的数据处理器，它被设计为像英文的语言，具有很强的输入/输出处理功能，现在已经成为商业界和政府机关处理数据的主要语言。它有很强的档案处理能力，具有广泛多变的数据格式化功能。它也像FORTRAN一样，缺乏许多现代程序设计的功能，如递归程序设计和内存寻址，因此，不适用于系统程序设计。COBOL的编译程序通常是很大的程序，因此不适用于个人计算机。COBOL也具有独立编写和编译子程序的功能，可用于开发大型程序。

BASIC

BASIC的用户认为计算机是可以程序化的计算机。BASIC被设计为FORTRAN的简易版，是用来教导初学者学习程序设计用的，是今日个人用计算机最常用的语言。

大部的BASIC都是解释程序而不是编译程序。解释程序可以直接执行一个语言，它不像编译程序，必须把高级语言转换成机器语言之后才能执行。BASIC是编写小程序及交互式程序的好语言，这是由于解释程序所提供的环境能够使它很容易地输入及测试程序的缘故。对一些较严谨的程序员而言，解释程序也有其缺点，因为解释程序其执行的速度一般都比编译的程序慢。

BASIC不像FORTRAN或COBOL，它不具备真正的子程序，因此，BASIC的程序较为冗

长且很难管理。另外它也缺乏内存直接寻址、位运算及其他系统软件所需要的运算。除了数组及字符串外,它也缺乏表示数据结构的能力。

PL/1

PL/1 是 IBM 为取代 FORTRAN 及 COBOL 二者而设计的语言,它有 FORTRAN 数字处理的能力,也有 COBOL 数据档案处理的能力。此外,还包含许多其他的功能,如位运算、内存寻址的能力,并有递归程序设计的功能。然而,PL/1 并没有像其设计者的愿望而取代 FORTRAN 及 COBOL,这个原因可能是因为 PL/1 的功能太多了,因而导致很难学会此语言,并且由于 PL/1 编译程序本身就很大,要使用此语言就需要一个大机器,因此,在个人计算机上使用 PL/1 将是个很大的缺点。

PASCAL

PASCAL 是从教授现代程序设计的观念而开发出来的。PASCAL 的用户认为计算机是具有高度结构化的机器,它能够表示许多种不同类型的数据,而且可用来解决各种不同的应用问题。它具有现代程序设计的功能,并且也是结构化程序设计的锐利语言。像其他高级语言一样(BASIC 除外),也具有个别开发子程序的功能,这是开发大型程序必须有的功能。另外,输入及输出语句也是此语言的一部分。

PASCAL 是强数据类型的语言,这表示程序的每一个变量都需指定为某一特定的数据类型,而且此语言严格地要求给每一个变量设定适当的数据类型值。因此,PASCAL 的用户认为计算机是一个必须遵守一些高度结构化的一般用途的机器。

目前所使用的 PASCAL 有许多不同的版本,所以在某一版本的编译程序中有的功能并不一定存在另一版本的编译程序中。尽管工业界已经有人使用 PASCAL,但其主要应用范围仍在教育界。

C

C 语言是为了折衷低级的汇编语言及上面所讨论的高级语言而设计的。C 语言用户认为计算机是介于汇编语言的用户与 PASCAL 用户观点间的机器,它提供了大部分微处理器所具有的运算(如内存寻址、位运算),同时允许程序员把问题以传统高级语言的构造,如重复、选择及判定来表达他或他的算法及数据,并提供汇编语言所缺乏而在高级语言才具有的数据结构。此外,它也提供了个别开发子程序的功能,以便开发大型的结构化程序(不像 BASIC 或某些 PASCAL 的版本)。

和 PL/1 不一样的是:C 语言是一个小型的语言,其最初开发的设计者只需要 12K 字节内存的小计算机就可执行 C 编译程序,而 C 编译程序的大小使得它成为个人计算机最理想的语言。为了维持 C 语言的便携,C 语言的设计者并未加入很多大部分处理器没有的运算,例如:C 并没有内部字符串处理的运算,甚至没有内部输入/输出语句,这些功能会因机器或应用而异,因此,把它们放在 C 语言之外,而做成可以被 C 语言调用的子程序。

C 语言设计者的另一目标是开发一个可移植性的语言,并可用于开发系统软件。如果用一种语言编写的程序可以很容易地从一种计算机搬到另一种计算机上,则称其为可移植性的语言。以汇编语言写的系统软件并不具有可移植性,因为汇编语言是因机器而异(尤其是不同的

硬件制造商)。然而,以高级语言写的程序就具有可移植性,因为不论是哪一种机器或操作系统,它们都一样可以用。在过去,应用软件(用高级语言编写)比系统软件(用汇编语言编写)更具有可移植性,因为应用程序只要在新机器上重新编译就可以用了。C语言的目标之一是提供一个具有开发系统软件功能的编译程序,因此许多较高级的功能被设计成语言之外的子程序(例如输入/输出及数据结构操作),这些功能可以被写成更能适应特殊应用或操作环境的需求,而不必挤在C编译程序或任何C语言编写的软件内。它有普遍常用的子程序集合而成的程序库,事实上,大部分的编译程序制造者提供了一个输入/输出和其他子程序的程序库叫做标准程序库。最后,C语言的另一特性叫做条件编译,它允许程序员分离出与机器类型有关的程序,控制在不同的环境下编译不同的内容,这更加增强了以C语言编写的软件的可移植性。

基于流程控制与数据结构的观点,C语言的用户认为它是比汇编语言更高级的语言,而且不像其他的高级语言,偏向于某一特殊类型的应用。C语言的程序员都被假设为有丰富程序设计经验的人,所以C编译程序并不像PASCAL那么严格地限定其数据类型,这个结果使C语言非常灵活,但也使它容易产生错误。

1.3 编译程序的使用

如果你曾经用过编译程序,就可以直接进行1.4节;如果你只是BASIC的用户,本节将会告诉你一些使用编译程序的知识背景。

如上面所讨论的编译程序是一个系统程序,它将某一高级语言转换为机器语言——汇编语言或机器语言。BASIC的解释程序并不需要这一步骤,因为它直接执行BASIC的每一语句。小程序或急于完成的非商业性产品的程序,用解释程序很适合;但要开发执行很多次的大型程序时,用编译程序较好,这是因为由解释程序执行一个程序的速度要比编译完成的程序直接在机器上执行慢得多。

使用编译程序比使用解释程序需要一些额外的步骤,这些步骤如图1.1所示。

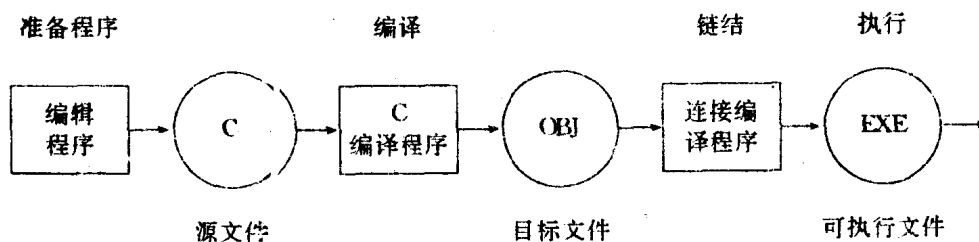


图1.1 编译程序的使用

以文字编辑程序或文字处理程序所编写的程序叫源文件,它是编译程序的输入文件,这个文件经过编译程序编译后,会另外产生一个文件叫做目标文件。其次,此目标文件可能要与其他目标文件(可能是不同的:编译程序所编译)经过输入程序或连接编译程序而连接在一起,以产生可执行文件,然后,此文件就可以直接在计算机上执行了。

1.4 PC 上 C 程序的内部结构(选择性的内存模式)

本节要探讨 C 编译程序运用微处理器管理内存的方法。

一个可执行的 C 程序由程序代码空间、堆栈空间、静态数据空间及动态数据空间四个部分组成。程序代码空间存储机器指令；堆栈空间是给暂存的数据及子程序连接用的；静态数据空间给程序的变量存储数据；而动态数据空间为程序转换时所需要的额外数据的空间，它可在程序执行时再分配位置。

早期的 C 编译程序如图 1.2 的方式分配内存：在程序执行的开始就设定 CS 寄存器的值为程序代码空间的起点，并且设定 ES、SS 及 DS 寄存器的值为堆栈空间及动态、静态联合数据空间的起点。

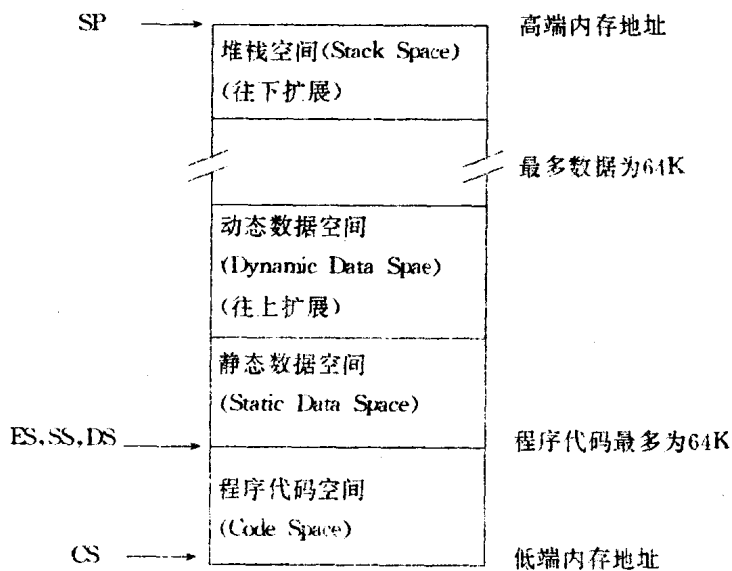


图1.2 小型内存模式

这种小型内存模式是用固定分段寄存器的方法，致使所编写的程序被限制，其程序代码空间最多为 64K，数据与堆栈的空间最多为 64K 的内存容量。用这种模式所提供的内存模式其执行速度较快，因为它不需要处理分段寄存器。然而，它没有让程序员运用 IBM PC 的所有能力。

新的编译程序提供了扩展内存范围的内存模式。大部分编译程序所提供的内存模式列表如下：

表 1.1 选择性的内存模式

模式	最大程序代码空间	最大数据空间
小型	64K	64K
大程序	1M	64K

(续表)

模式	最大程序代码空间	最大数据空间
大数据	64K	1M
大型	1M	1M

表 1.1 的数据空间为 C 内存模式的动态数据部分,而所有模式的静态数据都限制在最多 64K 的内存空间。同样,堆栈空间也限制在最多 64K 的内存中。

图 1.2 说明小型内存模式,图 1.3 为大程序内存模式,大程序模式的动态数据空间仍限制为 64K,由管理 CS 寄存器将程序分段扩展到 1M 字节的内存。

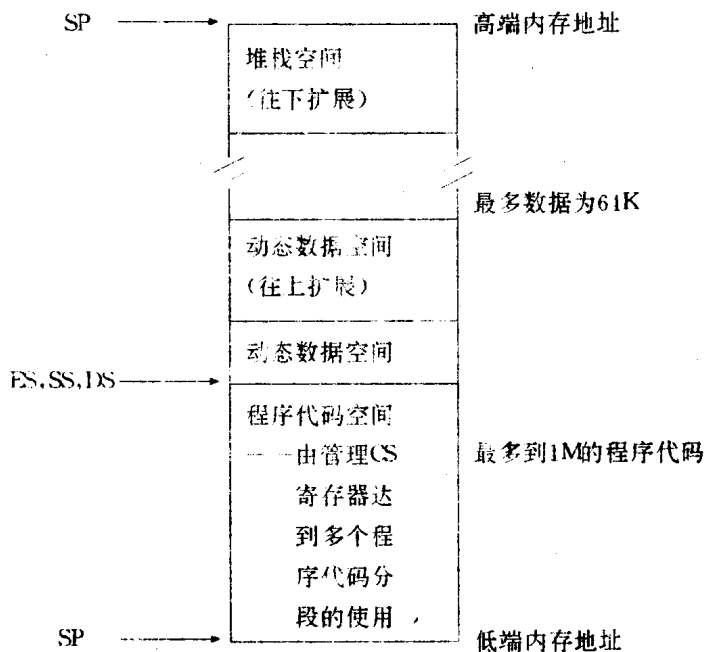


图1.3 大程序内存模式

图 1.4 说明大型数据内存模式,这种模式由管理数据分段寄存器(DS)提供大型的动态数据空间,但仍限制程序代码的空间。

图 1.5 说明最大型的内存模式,这种模式允许程序代码空间及动态数据空间扩展到最大的可用内存空间。

不同的内存模式在编译 C 程序时指定所选择的内存模式。事实上,编译程序根据不同的内存模式产生不同的机器码,例如选择大程序模式与选择小型模式所产生的连接子程序并不相同(FAR 对应于 NEAR——在以后再深入讨论)。选择大数据模式时,编译程序需要产生长达 4 个字节的地址(分段地址及差距值)而不是小型模式的 2 个字节的地址。对大部分的编译程序而言,都隐含着以下的事实:分别编译的 C 子程序连接成一个程序时,这些子程序必须是同一种内存模式,因为编译程序一起提供 I/O 子程序及其他函数的程序库,且对于不同的内存模式编译程序也会提供不同的程序库,每次编译都是针对某一特定的内存模式编译的。某些编译程序的编写者在设计上可能会与上面的讨论有些不同。值得注意的是:Microsoft 的新版 C

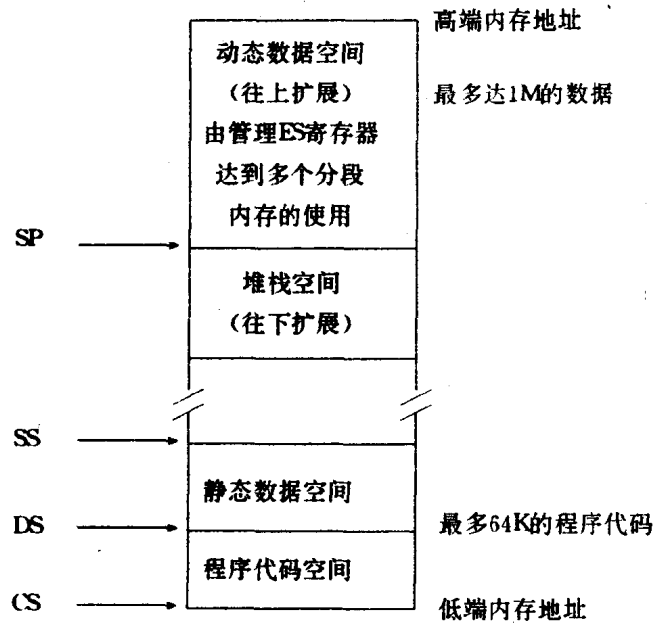


图1.4 大数据内存模式

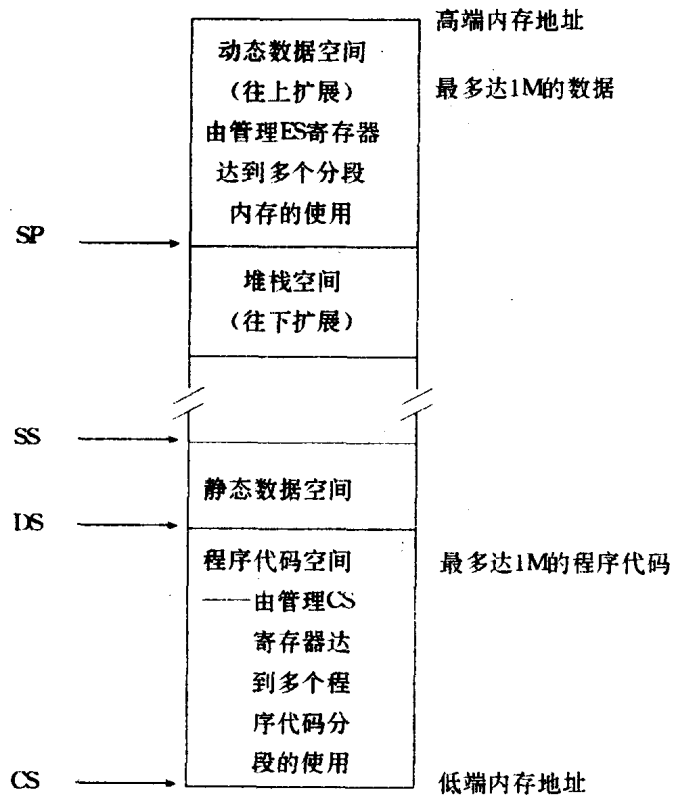


图1.5 大型内存模式

编译程序已经允许将使用不同内存模式的模块混合在一起。

内存模式的选择应根据其应用来选择适合需求的最小内存模式,因为较大的内存模式会

产生较多的程序代码,因而增加程序执行的负载。

1.5 结论

这一章我们讨论 C 语言的简史及其相对于其他流行语言的地位。我们知道 C 语言是为了结合汇编语言及大部分有特定应用范畴的高级语言的优点,因此,C 语言像其他应用软件一样,成为开发可移植性系统软件的强有力的语言。本书我们只介绍目前最流行的 Turbo C 语言,其特征和用法与其它 C 语言大体相似,读者可触类旁通。

第二章 Turbo C 语言的集成环境

2.1 Turbo C 的安装

一般 Turbo C 的源盘有四片,可以安装到硬盘驱动器上,也可以用软盘的方式来操作,说明如下:

一、硬盘的安装

1. 快速法(适合初学者)

这是一种方便且快速的安装方式,先在你的硬盘驱动器上建立一个目录(建议用 TURBOC 或 TC 来作为目录的名称),准备用来存放 Turbo C 的程序,如果你只要安装集成环境版本的 Turbo C,只要将第一、三、四片的 Turbo C 磁盘拷贝到 TURBOC 或 TC 的目录中即可;如果你只要安装命令行版本的 Turbo C,你只要将第二、三、四片的 Turbo C 磁盘拷贝到 TURBOC 或 TC 的目录中即可。如果两种版本都想安装的话,便将四片磁盘全部拷贝到你的硬盘驱动器中。

2. 洁净法

用上述快速法来安装 Turbo C 对未来程序开发阶段来说,会造成文件管理上的困难(时间一长,不知道哪个是自己写的、哪个是系统的),所以洁净法是利用树形的文件结构,将文件有系统的编排,分别存放。

首先在硬盘驱动器中建立一个目录(建议你用 TURBO C 作为目录名称)再在 TURBO C 目录底下建立两个子目录 LIB 和 INCLUDE,然后将第三片 Turbo C 中所有 .H 的文件全部拷贝到 INCLUDE 这个目录中,将第四片 Turbo C 中所有文件以及第三片 Turbo C 中不是 .H 的文件全部拷贝到 LIB 这个子目录中,再将第一、二片的 Turbo C 拷贝到 TURBO C 目录中。

当使用洁净法来安装 Turbo C 之后,在每次操作 Turbo C 之前你必须进行下列操作来配合:

(1) 使用集成环境

在 Option/Environment 的选择中要注明 INCLUDE 文件及 LIB 文件在哪个目录中。

(2) 使用命令行的方式

必须在命令行中加一个 -I 的选择,注明哪一个目录。

二、软盘的安装

1. 只有一台驱动器时

当然在这样的情况下,执行 Turbo C 效果一定不佳,但不管怎么说,它还是可以使用的。

基本上只要准备第一片及第三片源盘即可,其中第三片源盘保留其中的 .H 文件及下列各文件便可正确地执行 Turbo C,为了安全起见,最好先将源盘备份一遍。

C model.OBJ

C model. LIB

EMU. LIB

FP87. LIB

MATHmodel. LIB

model 是代表采用什么形式的内存模式(Tiny, Small, Large……),这是因为 Intel 808x 系统 CPU 的结构采取分段结构法,对于初学者来说使用小型内存模式便能应付自如,所以我们建议用 s 来代替 model 这个变量。

在使用时先将第一片磁盘放入驱动器中执行完 TC 之后,将其取出,再放入第二片,然后便可开发你所需要的程序,当想要使用 Help 指令时,需将第一片磁盘放入驱动器中。

2. 有两台驱动器时

想使用集成环境方式来操作 Turbo C 时,与只有一台驱动器是类似的,只不过是两片磁盘分别放在两台驱动器中执行便可。

当想使用命令行的方式时,拿一片经过格式化后的磁盘将第二片源盘中 TCC. EXE 及 TLINK. EXE 拷贝进来,同时建立两个子目录 LIB 及 INCLUDE,将所有的. H 文件(在第三片源盘中)拷贝到 INCLUDE 的子目录中,另外将下列文件(在第三片源盘中)拷贝到 LIB 的子目录里:

C model. OBJ

C model. LIB

EMU. LIB

FP87. LIB

MATHmodel. LIB

model 可用 s 来取代。在使用命令行的方式时只需要一片便可执行,而另外一台驱动器可用来当作你的工作区来开发你自己所写的程序。

2.2 Turbo C 的执行

要进入 Turbo C 的集成开发环境,你只需键入 TC 之后再按下回车键即可。当 Turbo C 开始执行时,你可以看到如图 2.1 的菜单。此菜单称为主屏幕,它包括四个部分,从上到下分别是:

- 主菜单
- 编辑状态显示栏以及编辑窗口
- 编译信息显示窗口
- 快速功能键显示栏