

C 程序设计

河海大学出版社

杨锦堂 编

高等学校教材

C 程 序 设 计

杨锦堂 编

河海大学出版社

(苏) 新登字第 013 号

内 容 简 介

本书是为高等院校非计算机专业学生编写的《C 程序设计》(或称《C 语言程序设计》)课程教材。主要内容有 C 语言的基本概念、语法规则、C 程序的设计方法和实用 C 程序介绍等。

本书以目前我国比较普及的 IBM-PC 系列微型机及其兼容机为背景,介绍 C 语言及实用 C 程序设计,并假设读者已学过一门计算机高级语言课程(BASIC 或 FORTRAN 等)。在内容的安排上,减少一般程序设计的基本算法,着重 C 语言的规则和特点,强调结合实际应用,但又不失本课程内容的系统性。

全书共分十二章,每章都附有习题。书后的附录中,除包含有常用的 ANSI 标准库函数以外,尚有很实用的 DOS 接口和屏幕图形函数,并且指出 Microsoft C 和 Turbo C 有关库函数的区别。

本书篇幅不大,内容叙述简明扼要,通俗易懂,除作为高校教材外,还适合用作有关微型计算机应用培训班教材和作为工程技术人员的自学参考书。

责任编辑 卢黎明

高等学校教材

C 程序设计

杨锦堂 编

出版发行:河海大学出版社
(南京西康路一号 邮政编码 210024)

经 销:江苏省新华书店

印 刷:河海大学印刷厂

787×1092 毫米 16 开本 印张 16.25 字数 365 千字

1993 年 11 月第 1 版 1993 年 11 月第 1 次印刷

印数 1—4000 册

ISBN 7-5630-0637-0/TP. 18

定价 8.60 元

(河海版图书若有印刷装订错误,可向承印厂调换)

前 言

C语言以其功能丰富齐全、表示方法简洁、使用方便灵活、程序设计自由度大、应用面广、可移植性好等一系列优点，不仅受到计算机专业工作者、软件工程技术人员的青睐，被用来编写系统软件；而且越来越多地为广大计算机应用技术人员所喜爱，被用来编写林林总总的应用软件。

C语言被称为最靠近低级语言(汇编语言)的高级语言。它具有高级语言的特点——容易学习、容易被人接受；又保留低级语言的优势——直接高效地调度计算机的硬件资源。近年来，各公司推出的优良的C程序开发环境，给广大科技工作者提供极为有效的微型机应用工具。在应用计算机的各行各业中，C语言被迅速推广使用。

不少高等学校，不仅计算机专业开设C语言课程，而且各非计算机专业也纷纷开设C语言课程。本书主要为非计算机专业读者编写。

在各行各业从事微型计算机应用的科技人员中，为数相对较少的计算机专业人员是中坚力量、骨干队伍，而非计算机专业的技术人员总是大多数，他们是普及推广微型计算机技术的基本力量。本书也为他们而写，即可作为计算机继续教育、各种培训班及自学的教材。

目前，多数学习C语言课程的读者，通常已学习了至少一门计算机程序设计语言(多为BASIC或FORTRAN)。因此，在其他高级语言介绍过的基本算法设计等内容，在本书中则从简，而把重点放在介绍C语言的本身以及能充分发挥C语言优势的实用程序设计。不过这种安排将不会影响本书的系统性和完整性。

计算机程序设计语言的高低之分已为大家所接受。接近人的自然语言的程序设计语言被称为高级的语言，接近机器语言(归根结蒂是'0'和'1'两个数码的组合)的程序设计语言被称为低级的语言。有趣的是，与此相反，在C语言程序设计中，那些与计算机硬件结合较紧密的内容(如存储方式、动态存储管理、与操作系统的接口、与汇编语言的接口、嵌入式汇编器等)，因为需要较多的硬件知识，学习掌握相对比较困难，在不少资料和著作中被列入'高级C程序设计'。编者已注意到，在本书有限的篇幅中，尽可能介绍这些很实用的程序设计内容和程序实例。

本书第一章至第九章为C语言程序设计的基本内容，第十章、第十一章和第十二章分别介绍C与DOS接口、图形处理与菜单技术以及C与汇编语言的接口。在教学实践中，各非计算机专业可根据需要及学时安排选用。对于没有学过'微型计算机原理'课程的专业学生，后面三章宜简略些。

C 语言内容丰富、自由度大,系统库函数浩瀚,要彻底掌握、运用自如,需要一个实践过程。本书是为初学者编写的,编者力图简明扼要地介绍C语言的基本知识,并尽可能引入一些实用的程序设计例,使初学C 程序设计的读者也能体会到C语言的优点,从而兴趣盎然,乐此不疲。

本书以目前我国比较普及的IBM-PC系列微型机(及其兼容机)上使用的C语言为背景(Microsoft 6.0版本为主,附录函数库说明中指出Turbo C与Microsoft C的区别)。书中引用的程序例都在PC机上调试过。不同C 系统、不同版本之间会有一些不大的差别,相信读者在实践中可以逐步适应。

本书由史金松教授审阅,他对书稿提出很多宝贵意见,编者谨表深切谢意。

编者水平有限,书中错误在所难免,恳请读者批评指正。

目 录

第一章 C 语言概述

§ 1-1 C 语言及其特点	1
1.1.1 C 语言简史	1
1.1.2 C 语言的特点	1
§ 1-2 C 程序	2
§ 1-3 C 程序上机操作	5
1.3.1 在PC-DOS操作系统下运行C 程序的步骤	5
1.3.2 在集成环境下运行C 程序	5
思考题与习题	6

第二章 数据类型、运算符与表达式

§ 2-1 概述	8
2.1.1 数据类型	8
2.1.2 常量与变量	8
2.1.3 变量的定义	9
§ 2-2 整型数据与实型数据	9
2.2.1 整型常量	9
2.2.2 整型变量	9
2.2.3 实型常量	11
2.2.4 实型变量	11
§ 2-3 字符型数据	12
2.3.1 字符常量	12
2.3.2 字符变量	13
2.3.3 字符串常量	14
2.3.4 字符数组	14
§ 2-4 类型的混合运算	15
§ 2-5 数组	17
2.5.1 数组的定义	17
2.5.2 数组的初始化	18
2.5.3 数组元素的引用	19
§ 2-6 基本运算符、表达式	21
2.6.1 运算符分类	21
2.6.2 表达式	22

2.6.3	算术运算符、算术表达式	22
2.6.4	赋值运算符	23
2.6.5	关系运算符	25
2.6.6	逻辑运算符	25
2.6.7	逗号运算符	26
习题		27
第三章 C 程序设计初步		
§ 3-1	C 语句	29
§ 3-2	函数及函数库	31
3.2.1	函数的定义	31
3.2.2	函数参数及参数传递	32
3.2.3	函数的返回值	33
§ 3-3	库函数	35
3.3.1	库函数简介	36
3.3.2	嵌入文件	40
§ 3-4	数据输出函数	41
3.4.1	字符输出函数	42
3.4.2	格式输出函数	43
§ 3-5	数据输入函数	47
3.5.1	字符输入函数	47
3.5.2	格式输入函数	48
习题		51
第四章 流程控制		
§ 4-1	if 语句	53
4.1.1	if语句的三种形式	53
4.1.2	if语句的嵌套	56
4.1.3	条件运算符	56
§ 4-2	for 语句	57
§ 4-3	while 语句和do-while 语句	59
4.3.1	while 语句	60
4.3.2	do-while 语句	60
§ 4-4	break 语句和continue语句	61
4.4.1	break语句	61
4.4.2	continue语句	62
§ 4-5	switch 语句	63
§ 4-6	goto 语句	64
§ 4-7	程序例	65
习题		67

第五章 编译预处理

§ 5-1 宏定义	69
5.1.1 不带参数的宏定义	69
5.1.2 带参数的宏定义	71
5.1.3 带参数的宏与函数	72
§ 5-2 文件包含处理	74
§ 5-3 条件编译	78
习题	80

第六章 函数调用

§ 6-1 函数的调用	81
6.1.1 函数调用形式	81
6.1.2 被调函数的说明	82
§ 6-2 嵌套调用与递归调用	83
6.2.1 函数的嵌套调用	83
6.2.2 函数的递归调用	86
§ 6-3 数组作函数参数	89
§ 6-4 内部函数与外部函数	93
6.4.1 内部函数	93
6.4.2 外部函数	93
§ 6-5 局部变量与全局变量	95
6.5.1 局部变量	95
6.5.2 全局变量	96
§ 6-6 变量的存储类别	97
6.6.1 局部变量的存储方式	98
6.6.2 全局变量的存储方式	100
6.6.3 变量存储类别小结	101
习题	102

第七章 指针

§ 7-1 概述	103
§ 7-2 变量的指针	104
7.2.1 指针变量的定义	105
7.2.2 指针变量的引用	105
7.2.3 指针变量作函数参数	106
§ 7-3 数组的指针	108
7.3.1 指向数组元素的指针变量	109
7.3.2 指向数组的指针的使用	110
7.3.3 指向多维数组的指针	111
§ 7-4 字符串指针	116

7.4.1 指向字符串的指针变量	116
7.4.2 字符指针变量与字符数组	118
§ 7-5 函数的指针和指针函数	119
7.5.1 函数的指针	119
7.5.2 指针函数	120
§ 7-6 指针数组和指向指针的指针	122
7.6.1 指针数组	122
7.6.2 main函数的参数	123
7.6.3 指向指针的指针	125
§ 7-7 指针小结	128
习题	129

第八章 结构数据

§ 8-1 结构类型变量的定义	131
§ 8-2 结构类型变量的引用	134
8.2.1 结构类型变量的初始化	134
8.2.2 结构类型变量的引用	134
§ 8-3 结构数组	137
§ 8-4 结构指针	138
8.4.1 指向结构变量的指针	138
8.4.2 指向结构数组的指针	140
8.4.3 用结构指针作函数参数	140
§ 8-5 动态链表	141
§ 8-6 联合	147
8.6.1 联合的概念	147
8.6.2 联合变量的引用	149
§ 8-7 枚举类型和用typedef定义类型	152
8.7.1 枚举类型	152
8.7.2 用typedef定义类型	153
习题	154

第九章 文件处理

§ 9-1 文件概述	155
§ 9-2 文件类型指针	156
§ 9-3 流式文件函数	157
9.3.1 文件的打开和关闭	157
9.3.2 文件的读写	159
9.3.3 文件的定位	164
习题	165

第十章 C 与PC-DOS接口

§ 10-1	8086 处理机	167
10.1.1	8086 CPU	167
10.1.2	地址计算	168
§ 10-2	存储模式	169
10.2.1	标准存储模式	169
10.2.2	混合存储模式	170
§ 10-3	8086中断与PC-DOS	173
10.3.1	8086中断	173
10.3.2	MS-DOS的中断	173
10.3.3	C 访问BIOS系统资源	174
§ 10-4	C 访问DOS 功能调用	178
10.4.1	DOS 功能调用	178
10.4.2	DOS 接口例程	178
10.4.3	中断函数	184
§ 10-5	位运算符及其应用	186
10.5.1	字段	186
10.5.2	位运算符	189
习题		192
第十一章 图形处理与菜单技术		
§ 11-1	图形处理	194
11.1.1	视屏模式	194
11.1.2	低级图形函数	195
11.1.3	动画设计	199
11.1.4	直观图形	202
§ 11-2	菜单技术	207
11.2.1	简易菜单	207
11.2.2	文本窗口与边框	209
11.2.3	光条技术	210
11.2.4	弹出式菜单	212
11.2.5	"DOS Shell"处理	217
习题		218
第十二章 C与汇编语言的接口		
§ 12-1	C 调用汇编语言	220
12.1.1	调用约定	220
12.1.2	被调用的汇编语言程序	221
12.1.3	C 程序接口	222
12.1.4	建立汇编语言框架	222
12.1.5	C 调用汇编语言程序举例	224

§ 12-2 内部汇编器	228
习题	230
附录一 ASCII表	231
附录二 C 运算符及其优先级与结合性	232
附录三 C 库函数	233
附录四 错误信息表	244

第一章 C 语言概述

§ 1-1 C 语言及其特点

C 语言是目前广泛流行的计算机程序设计语言。C 语言有许多突出的优点,它既能被用来编写系统软件,又可用于编写应用软件,是一种很有发展前途的高级语言。

1.1.1 C 语言简史

70年代初,美国贝尔实验室K. Thompson和Dennis M. Ritchie两人合作,在被称为B语言(Basic Command Programming Language)的基础上设计成功C语言。他们将UNIX操作系统(原用汇编语言写成)用C语言重新改写。此后,C语言作了多次改进,一直和UNIX操作系统联系在一起,至UNIX第6版公布,C语言的优点引起人们的普遍注意。

1978年,Brian W. Kerlinghan和Dennis M. Ritchie(人们把他俩称为K&R)合著的有里程碑意义的名著《The C Programming Language》一书问世,此书奠定了C语言的基础,它所介绍的C语言被称为标准C。1983年,美国国家标准化协会(ANSI)制定C语言新的标准,被称为ANSI C。1988年,K&R按照ANSI C标准,又修改了他们的《The C Programming Language》。

C语言开始只是附属于UNIX,在小型机PDP-11上实现。现在已被移植到各种大、中、小机,可在各机种和各操作系统上运行,在微型机上使用尤为普遍。

目前,在微型机上广泛使用的C语言编译系统有Microsoft公司的Microsoft C和Borland公司的Turbo C等。各种不同版本略有差别,但他们的基本部分都相同,符合ANSI标准。

1.1.2 C语言的特点

C语言与其他计算机语言比较,有明显的特点:

一、C语言允许访问物理地址,能进行位操作,能实现汇编语言的大部分功能,可直接对硬件进行操作。C语言比其他任何高级语言更具有汇编语言的特点,因而,被迅速应用到以前汇编语言所应用的领域。C语言又有高级语言的编程方便、可读性好的优点,因此,它又是一个实用的程序设计语言。

二、C语言表达简洁,结构紧凑,使用方便、灵活,C程序书写自由。例如:

定义f为一个返回整型值的函数

C语言

PASCAL语言

int f(); FUNCTION f() : INTEGER

定义a 为一个整型一维数组

C 语言 PASCAL 语言
int a[]; VAR a: ARRAY[...] OF INTEGER

无论是否学过PASCAL语言, 读者都可看到C 语言表达简洁得多

C 语言的关键字少, ANSI C 仅有的32个关键字:

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

其中const、volatile为ANSI C规定保留 此外, 有些系统尚有asm、fortran、far、near、interrupt、pascal等少数几个关键字。

学过高级语言的读者, 可以看到C 语言的关键字少, 容易记忆 通过学习本书内容, 读者一定能掌握绝大多数关键字的使用方法

三、C语言有丰富的运算符和表达式, 使用方便、灵活 例如, C 语言中将赋值符号 '=' 作为赋值运算符处理, 从而可组成赋值表达式(不是赋值语句) 使用赋值表达式, 使程序书写特别简洁、方便、灵活

如: if ((a=getch())==0x0d) _outtext('OK!');

其意是将函数getch() 的返回值赋给变量a, 如果是'Enter'键, 则输出'OK!'

四、C 语言规定丰富的数据类型(有整型、实型、字符型、数组类型、指针类型、结构类型和联合类型等), 能用来实现各种复杂的数据结构

五、C 语言的可移植性好, 无需作很多修改就可移植到各种计算机和各种操作系统 当然, C 程序在不同环境之间进行移植时, 要注意各变量的类型和字长以及外部函数的程序之间的关系

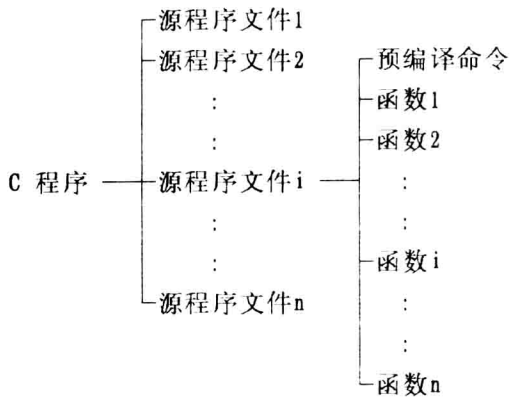
六、C 语言语法限制不太严格, 程序设计自由度大 而其它高级语言一般语法检查比较严格, 能检查出几乎所有的语法错误, 因而程序设计者只要按照语法规则编写程序, 就能得到正确结果 C 语言对变量类型、变量范围和存储空间的存取的制约少, 程序设计自由度大 C 编译放宽语法检查, 因而相对比较而言, 编写正确的C 程序可能会比用其它高级语言编写程序稍难一些

§ 1-2 C 程序

用C 语言编写的源程序, 简称为C 程序 C 程序由一个或多个函数组成 若由多个函数组成, 这些函数可以集中放在一个文件中, 也可以分散放在几个文件中 C 语言编写的源程序文件, 通常需要用C 为扩展名 如: 'EXS1.C', EXS1为源文件名, C 为扩展名

一个C 源程序文件包含一个或若干个函数以及预处理命令(预处理命令不是必须的)。

C 程序结构示意图如下:



一个C 程序至少包含一个函数(此函数名为main),也可以包含一个main函数和若干个其它函数。可见,函数为C 程序的最小单位

[例1-1] 有一个已编就的C 程序,只有一个文件(文件名为EXS1_1.C),文件中只包含一个函数(main函数) 现将文件EXS1_1.C输出显示如下:

```
main( )           /* 这是主函数 */
{
    int a, b, sum; /* 定义三个整型变量a、b、sum */
    a=55;          /* 以55赋值给变量a */
    b=33;          /* 以33赋值给变量b */
    sum=a+b;       /* a+b赋值给sum */
    printf("sum is %d\n", sum); /* 以十进制整数形式输出变量sum的值 */
}                  /* 函数结束 */
```

本程序的作用是求a、b两整数的和sum。其输出为:

```
sum is 88
```

程序开头main() 表示一个主函数,每一个C 程序都必须有一个主函数。本例中整个程序只有一个函数(就是这个main函数),大括号{ }内为函数体。本例函数体包含五条语句(每一条语句都应以分号结尾)。其中,第一条语句定义三个整型变量;第二、三、四条为赋值语句;第五条为输出函数调用语句,语句中用到的printf是C 程序中的输出函数(将在第三章中详细介绍)。输出函数printf的圆括号内是函数参数,其中双引号内的字符串按原样输出(即输出sum is) '%d'是输出格式字符串,此处表示以十进制整数形式输出变量sum的值 '\n'表示输出换行。printf函数的圆括号内,逗号右边sum是要输出的变量名。/**/表示程序语句的注释部分,C 编译对注释部分不予理睬,注释只是增加程序的可读性。C语言规定,注释部分可出现在程序的任何位置。

C 语言源程序书写是很自由的,我们可以将此源程序改写如下:

```
main( )
{ int a, b, sum; a=55; b=33; sum=a+b;
  printf("sum is %d\n", sum); }
```

程序一样正确。

也可将程序改写成:

```
main()
{ int a, b, sum; a=55; b=33; sum=a+b; printf
  ("sum is %d\n", sum); }
```

程序仍然正确。可见, C 语言源程序中, 一行可写若干条语句, 或者一条语句可分成几行写, 编译都认可。

[例1-2]

```
main()
{
  int a, b, c;
  scanf("%d,%d", &a, &b); /* 调用输入函数scanf, 输入变量a、b的值 */
  c=sum(a, b);           /* 调用sum函数, 将得到的值赋给变量c */
  printf("sum=%d", c);    /* 输出变量c 的值 */
}

int sum(x, y)             /* 定义一个名为sum的函数, 函数的值为整型, 其中x、y是函数的
                           形式参数 */
int x, y;                 /* 定义形式参数的数据类型 */
{
  int z;                  /* 定义sum函数中用到的变量z的类型 */
  z=x+y;
  return(z);              /* 将z的值返回, 通过函数sum带回到主调函数的调用处 */
}
```

本例程序中包含两个函数: 主函数和被调用的函数sum。sum函数的作用是求变量x、y之和, 并将和赋值给变量z。return语句将z的值返回给主调函数main。

本例主函数中, scanf(输入函数)与printf函数一样, 是C语言函数库中的函数。调用scanf函数, 是为了从键盘输入a和b的值。关于scanf函数用法详见第三章。

程序第五行是调用sum函数语句, 在调用时将实际参数a和b的值分别传递给sum函数中的形式参数x和y。经过执行sum函数得到一个返回值, 并将这个值赋给变量c。当然, 读者可以看到本例中被调函数功能特别简单, 只是将两个整型变量相加, 完全可以在main函数中解决。之所以设计成两个函数, 其目的是希望通过本例使读者对C 程序的组成和形式有一个初步的了解。

有关函数调用、函数的形式参数和实际参数等, 在以后有关章节将会详细介绍。通过以上例子, 我们可以小结如下:

一、C程序由函数组成。一个C程序必须有一个主函数, 还可以包含若干个其它函数。函数是C 程序的基本单位。出现在源程序中的各个函数, 互相之间是并行的。

二、函数由说明部分和函数体两部分组成。C 程序中的函数相当于其它高级语言中的子程序。C的函数库非常丰富, 标准C库函数有一百多个, 有关软件公司提供的C版本, 库

函数就更多。如Microsoft C 6.0 提供近五百个库函数。用户根据自己的需要,调用C 函数库中的函数,也可以自己设计专用的函数(如上例中的sum函数)。

三、C 程序总是从main函数开始执行,这与main函数在源程序中的位置无关。

四、C 程序书写格式自由,没有行号,一行可以写几个语句,一个语句可分成几行写。然而,为了增加源程序的可读性,书写整齐显然是有好处的。

五、每个语句后面必须以分号结尾。

六、源程序中可以加注释,注释部分(/* */)可出现在程序的任何位置。注释是为了增加程序的可读性。一般推向市场的、有价值的程序都有必要的注释。

§ 1-3 C 程序的上机操作

C 源程序经过编译程序(Compiler 又称编译器)编译,生成目标程序,然后进行链接,得到可执行程序。现以IBM-PC机及其兼容机为例,简要介绍分别在PC DOS操作系统下的上机操作以及在集成环境下的上机操作的步骤。

1.3.1 在PC DOS操作系统下运行C 程序的步骤

一、编辑源程序

现在各公司提供给用户在PC机上用的编辑程序(Editor)很多,如DOS Edit、Q-Edit、PE(Personal Editor)等。读者可以任选一个对自己最熟悉的编辑程序,来编辑、修改源程序。检查确认无误后,存入文件系统(例如文件名为EXS.C)。

二、编译

早期的C 编译系统,在DOS下运行,C 源程序需经过四道编译(CC1, CC2, CC3, CC4),才能生成目标程序。如果编译过程中发现有语法错误,系数会输出“错误信息”,告诉用户什么错误。用户重新调用编辑程序修改源程序后,再进行编译,直到通过生成目标程序(例如EXS.OBJ)为止。

三、链接

调用系统的链接程序(Linker),键入LINK命令以及目标文件名和库文件名,将编译生成的目标程序与库函数链接起来,得到可执行文件(例如EXS.EXE)。

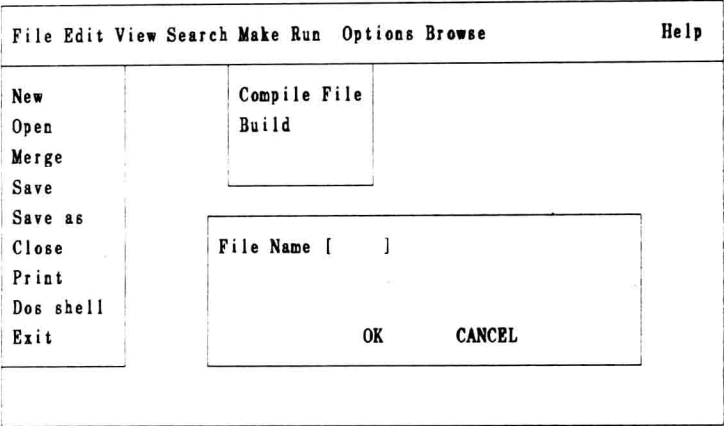
目前,各高版本的C 编译系统在DOS下运行,只需键入CL命令及有关文件名,系统自动地先编译后链接,有错误提供错误信息,无错误编译链接一次完成。

1.3.2 在集成环境下运行C 程序

Microsoft、Borland等公司新推出的C 编译系统都提供优良的集成环境,使用非常方便。如Microsoft C 5.0以上版本都提供快速编译器(Quick C)集成化编译调试环境。现以Microsoft C 6.0为例,简要介绍集成环境下编译器的使用方法。

Microsoft C 6.0提供程序员工作台(PWB)专用开发环境,集编辑、编译、链接、调

试于一体,使用非常方便。在DOS下键入命令PWB,屏幕上显示一个下拉式菜单:



屏幕上方显示有一排主菜单选择项: File、Edit、View、Search、Make、Run、Options、Browse、Help等。利用 'Alt' 键及'→←↑↓' 键(或鼠标器),移动光标,可选择各菜单项。

如选择'File',按回车键,屏幕上会显示下拉式的子菜单。在子菜单中,选择'New'表示等用户输入(编辑)新的源文件。选择'Open'表示打开一个已经存放在系统的源文件,此时,屏幕上会弹出一个对话框,要求用户输入文件名。如果输入文件名正确,系统就将源文件从盘中调出,供用户修改和编译。选择'Save'可将以编就的源文件存盘。如选择'Exit'则退出PWB,返回操作系统。

选择'Make',按回车键,屏幕上会显示另一个下拉式的子菜单。在子菜单中,若选择'Compile File'可进行当前源程序的编译。编译结果,若无错误则生成目标文件(.OBJ文件)。接着可选择'Make'下的另一子菜单项'Build',进行链接。链接结果,若无错误可建立执行文件(.EXE文件)。

如在编译或链接中发现错误,屏幕上会显示错误信息。错误信息会指出在编译中发现源程序的哪一行有什么性质的错误,或在链接中缺少应有的目标文件(.OBJ)或库文件(.LIB)等。用户可根据错误信息的提示,修改后重新编译或链接,直至得到正确结果。

可执行文件又称运行文件,可以在PWB环境下运行。此时,只需将光标移至'Run',按回车键,然后确认运行文件名,便可运行。在PWB环境下,还有调试等其它功能,读者需通过上机实践,逐步掌握。

当然,可执行文件(.EXE文件)可以在操作系统下运行。此时,只需键入可执行文件名,按回车键,即可运行。

思考题与习题

- 1-1 C语言与其它高级语言比较有哪些特点?与汇编语言相比,C语言有什么优点?
- 1-2 编写一个C程序,输出以下内容: