

计算机C/C++语言系列丛书

Instant-C

新型的高速

C 语言使用指南

秦笃烈 童隆正 周晓玉 编

希 望 审校



学 苑 出 版 社

计算机C/C++语言系列丛书

Instant-C

新型的高速C语言使用指南

秦笃烈 童隆正 周晓玉 编

希 望 审校

学苑出版社

1993

(京)新登字151号

内 容 提 要

Instant-C是一种新型的C语言。它吸取了Lisp, Smalltalk以及FORTH语言的许多优点。它的独创之处在于使用最新的增量编译方法,在对源程序编辑的同时自动进行编译,而对修改的程序仅编译修改部分,既省去了传统的编译命令,同时也使编译速度提高50-500倍。它的环境分为直接模式“解释程序”,编辑程序以及用户程序工作环境三部分,各有许多独特之处。它可用作系统软件和应用软件开发工具,开发效率提高一倍以上。它可在保护模式下直接访问16兆内存。它具有和Microsoft C等C语言的兼容软开关。它能处理每秒1千次以上中断。系统分别包括一百余条命令、一百多个函数和宏。无论对专业程序设计人员还是对一般程序设计人员,Instant-C都是优良的工具。它在系统程序设计、应用程序设计以及C语言教学方面都具有重要价值。

需要本书的用户,请与北京8721信箱资料部联系,邮编:100060,电话:2562329。

计算机C/C++语言系列丛书

Instant-C新型的高速C语言使用指南

著:秦笃烈 童隆正 周晓玉

校:希 望

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:沙河印刷厂

开 本:787×1092 1/16

印 张:18.625 字数:416千字

印 数:1—5000册

版 次:1993年12月第1版第1次

ISBN7-5077-0875-6/TP·24

本册定价:13.00元

学苑版图书印、装错误可随时退换

编 者 序

Instant-C是由美国Rational System公司开发的。它开发的Instant-C/16M能直接访问DOS 16兆内存并因此闻名于计算机界。读者细读本书以后，自然可以知道Instant-C的优越性和特色。真正的评价留给读者和用户。

我们特别要感谢北京希望电脑公司秦人华副总裁。由于她以传播计算机高科技为己任，使软件及图书的出版工作蒸蒸日上、硕果累累，这也是我们和北京希望电脑公司长期愉快合作的基础。

秦笃烈 1993.12. 于静思斋

目 录

第一章 概述	(1)
1.1 Instant-C的组成 部分	(1)
1.2 三个独特的环境	(2)
1.3 直接执行模式	(3)
1.4 对表达式求值	(3)
1.5 自动化编译	(4)
1.6 编辑程序	(4)
1.7 磁盘文件和内存文件	(5)
1.8 运行程序	(6)
1.9 对代码进行调式	(6)
1.10 建立可执行文件	(7)
1.11 小结	(7)
第二章 准备工作	(8)
2.1 系 统 要 求	(8)
2.2 Instant-C安装 方法	(8)
2.3 运行测试程序	(9)
2.4 Instant-C发行包中包含的文件	(10)
第三章 Instant-C使用方法	(14)
3.1 启 动Instant-C	(14)
3.2 执行表达式	(15)
3.3 发布Instant-C命令	(18)
3.4 设计程序示范	(20)
3.5 退 出Instant-C	(24)
3.6 建立独立程序	(24)
3.7 命令回顾	(29)
3.8 Instant-C命令行变元	(30)
3.9 对Instant-C命令进行编辑	(31)
3.10 使程序中断	(31)
3.11 源窗口使用方法	(33)
3.12 显示处理	(33)
第四章 编辑程序使用方法	(35)
4.1 学习基本功能	(35)
4.2 启动编辑程序	(40)
4.3 改变模式	(41)

4.4	移动光标.....	(42)
4.5	输入文本.....	(43)
4.6	删除文本.....	(43)
4.7	代替文本.....	(44)
4.8	拷贝和移动文本.....	(44)
4.9	发布编辑程序命令.....	(45)
4.10	使用缓冲器.....	(46)
4.11	退出编辑程序.....	(50)
4.12	保存未完成的代码.....	(50)
4.13	使用独立的编辑程序.....	(52)
4.14	解决内存问题.....	(52)
第五章	对样本程序进行调试.....	(54)
5.1	对原来程序进行测试.....	(54)
5.2	定义漏缺的函数.....	(56)
5.3	纠正输出格式.....	(60)
5.4	获得正确值.....	(62)
5.5	小结.....	(65)
第六章	C语言	(67)
6.1	未实现的功能.....	(67)
6.2	与硬件有关的功能.....	(67)
第七章	装入目标代码.....	(69)
7.1	建立目标代码接口.....	(69)
7.2	建立针对特定编译程序的Instant-C版本.....	(70)
7.3	支持编译后的目标代码.....	(71)
7.4	使用汇编程序代码.....	(74)
7.5	支持商品化库.....	(77)
第八章	对Instant-C进行修改.....	(78)
8.1	为所用计算机配置Instant-C.....	(79)
8.2	修改系统选择变量.....	(84)
8.3	修改数据显示格式.....	(86)
8.4	使库更小些.....	(86)
8.5	对Instant-C作永久性修改	(88)
8.6	配置屏幕.....	(92)
8.7	配置键盘.....	(99)
8.8	配置独立的编辑程序.....	(100)
第九章	实时程序设计和中断处理.....	(100)
9.1	中断程序设计.....	(100)
9.2	Instant-C中断15.....	(102)
9.3	例1: 系统时钟	(102)

9.4	例2: 对非标准图形设备的支持	(102)
第十章	Instant-C/16M: 直接访问16兆内存	(104)
10.1	Instant-C家族的第二个成员	(104)
10.2	系统要求	(105)
10.3	处理不兼容的BIOS	(105)
10.4	好的内存控制方法	(107)
10.5	保护模式限制	(107)
10.6	Instant-C/16M的savemod命令	(110)
10.7	内装的#define	(110)
第十一章	Instant-C命令.....	(111)
11.1	back命令	(111)
11.2	cd命令	(112)
11.3	cls命令	(112)
11.4	copy命令	(113)
11.5	datasegment命令.....	(113)
11.6	datasize命令	(113)
11.7	#define命令	(114)
11.8	delete命令... ..	(114)
11.9	deletem命令	(115)
11.10	dir命令.....	(115)
11.11	ed命令	(115)
11.12	edconfigure命令.....	(116)
11.13	edd命令	(116)
11.14	edi命令.....	(117)
11.15	edm命令	(118)
11.16	erase命令.....	(119)
11.17	go命令	(119)
11.18	#include命令	(119)
11.19	infile命令.....	(120)
11.20	ipath命令	(120)
11.21	list命令.....	(121)
11.22	listfile命令	(121)
11.23	listmissing命令.....	(122)
11.24	listname命令.....	(122)
11.25	llist命令	(123)
11.26	load命令	(123)
11.27	loadlib命令.....	(124)
11.28	loadmap命令	(124)
11.29	loadobj命令	(125)

11.30	local命令	(125)
11.31	make命令	(126)
11.32	monitor命令	(127)
11.33	new命令	(127)
11.34	outfile命令	(128)
11.35	pc命令	(129)
11.36	pd命令	(129)
11.37	pf命令	(130)
11.38	ph命令	(130)
11.39	pld命令	(131)
11.40	plf命令	(132)
11.41	plo命令	(132)
11.42	plu命令	(133)
11.43	plx命令	(134)
11.44	po命令	(134)
11.45	ps命令	(135)
11.46	pu命令	(135)
11.47	px命令	(136)
11.48	qquit命令	(137)
11.49	quit命令	(137)
11.50	#rename命令	(137)
11.51	renamem命令	(138)
11.52	reset命令	(138)
11.53	return命令	(139)
11.54	run命令	(139)
11.55	save命令	(140)
11.56	savemod命令	(142)
11.57	screen命令	(143)
11.58	segments命令	(144)
11.59	shell命令	(145)
11.60	show命令	(146)
11.61	step命令	(146)
11.62	step exec命令	(146)
11.63	step in命令	(147)
11.64	step out命令	(147)
11.65	step return命令	(148)
11.66	system命令	(148)
11.67	time命令	(148)
11.68	trace命令	(149)

11.69	type命令	(149)
11.70	unmonitor命令	(149)
11.71	untrace命令	(150)
11.72	use命令	(150)
11.73	view命令	(151)
11.74	where命令	(151)
第十二章	Instant-C函数库	(152)
12.1	库源文件	(152)
12.2	库标题文件	(155)
12.3	库函数摘要	(156)
12.4	函数分述	(159)
第十三章	对Instant-C使用中若干问题的回答	(239)
附录A	错误报文和解释	(244)
A.1	程序错误	(244)
A.2	内部错误	(277)
A.3	启动错误	(280)
A.4	独立应用错误	(280)
附录B	命令摘要	
B.1	Instant-C命令	
B.2	功能键	
B.3	编辑程序功能键和命令	

第一章 概 述

Instant-C为C语言提供了一个完整的程序设计环境。它的设计受到几个计算机科学研究系统（包括MACLISP, INTERLISP以及SMALLTALK）和FORTH微型计算机系统的精妙之处的激励和滋养。假如你对这些系统已经熟悉，那么Instant-C的程序设计环境对你就不会是陌生的。

本章要介绍Instant-C和传统程序设计工具相对应的组成部分，然后对如何在Instant-C中进行工作作扼要说明。不过要记住，这些仅仅是概貌性说明。本章所讨论的所有内容将在第3，4和5章更详细地进行探讨。

1.1 Instant-C的组成部分

Instant-C的程序设计环境由若干部分构成，它们相当于传统程序设计环境中的工具。对这些工具你可能比较熟悉。Instant-C有一个全屏幕编辑程序和漂亮的打印程序以便输入源代码。Instant-C也包含有标准的C函数库、编译程序和连接程序，可用它们建立可执行程序。为了有效地查错，它配有原语言调试程序和系统核查程序。

用户可以访问C“解释程序”、编辑程序和函数库。另有一些东西是不可见的，你会从它们在“后台”做的工作获益。下面是Instant-C的组成部分的介绍：

I. C编译程序

编译程序把C源语言翻译成可执行的机器指令。当将新代码存盘后离开编辑程序或从磁盘装入已存在代码时它将自动地不可见地加以引导。只有当你把一个大型函数存盘使编译程序做完它的工作稍有延迟时才会注意到它的存在。

II. C解释程序

Instant-C的直接模式的工作方式很象解释程序，而且它对Instant-C有全面的控制能力。可以用它装入、运行、调试和贮存程序。

III. 全屏幕编辑程序

编辑程序的作用是建立和修改C语言程序。它读取你发布的命令字符（通常是一次击键）并对源代码进行操作。当把编辑的代码存盘并返回直接模式时将自动对它进行编译。

IV. 连接程序

连接程序把各个源文件或目标文件中的函数组合成一个程序。也用它建立独立的可执行文件。连接程序是作为不可见的编译过程的一部分自动引导的不可见工具。

V. 源格式化程序

源格式化程序“漂亮的印出”C源代码使它具有标准的而易读的层次。它也是作为编辑过程的一部分自动引导的不可见工具。

VI. C函数库

函数库提供常用的操作，如读写磁盘文件。它包括在K & R中描述标准函数，UNIX兼容的函数，低级硬件函数以及DOS接口函数中描述的标准函数。这个库是以源码形式提供，以便于你对特定的函数作出修改。

也可以建立一种使用Lattice C或Microsoft库而不是Instant-C标准库的Instant-C版本。以后就可以把适用于Lattice或Microsoft编译程序的商品化库加进去使用。

Ⅶ. 源语言调试程序

调试程序帮助你调试和跟踪C程序。它使你能对程序和数据进行检查和修改。这个工具和直接模式结合在一起。它的可见性仅为几条命令而已。

Ⅷ. 跨模块语法检查程序

跨模块语法检查程序对于由多个源文件构成的C程序的一致性进行检查。它是作为不可见的编译过程的一部分自动而不可见地加以引导。用户看到的结果类似于对通常的语法错误所产生的错误报文。

Ⅸ. 运行态检查程序

运行态检查程序用于查找运行态错误，如被零除，数组界溢出，无效指针，调用未定义函数，以及指针算术运算溢出。它在代码执行过程中自动而不可见地加以引导。

1.2 三个独特的环境

Instant-C被用户看到的是两个不同的程序环境：Instant-C的直接模式“解释程序”和编辑程序。用户可能使用Instant-C开发从键盘读取并写入屏幕的交互式程序。用户自己的程序提供用于自身工作的第三个环境。这三个程序环境必要时可引伸出其它成分。

图1-1展示这三个Instant-C的基本环境：直接模式“解释程序”、编辑程序和用户程序如何结合及切换的模式图。

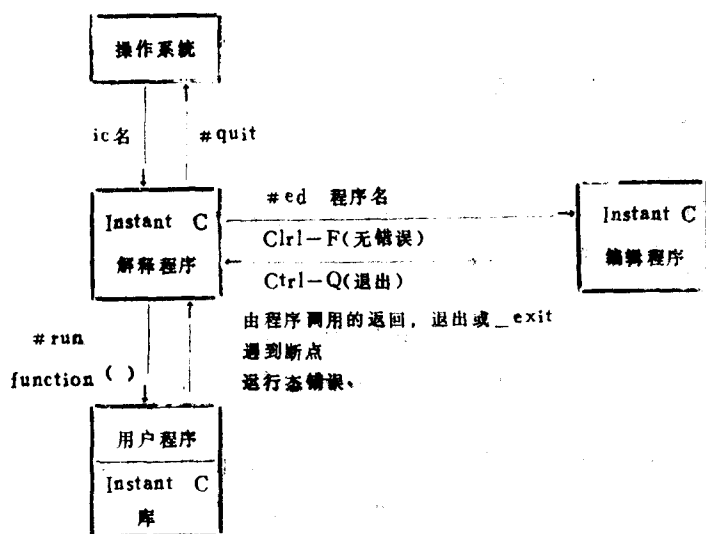


图 1-1 Instant-C三种环境的组成及切换模式图

该模式图表明，用户可以从操作系统进入Instant-C的直接模式，该模式的作用很象解释程序。这一模式提供对Instant-C的控制。用户可以从这个“解释程序”访问编辑程序对单个函数或文件进行编辑。在对代码编辑以后自动进行编译从而使用户返回到直接模式“解释程序”。也可选择不经编译而退出编辑程序并返回到直接模式“解释程序”。

除了访问编辑程序以外，可以从Instant-C的“解释程序”发布诸如图1-1模式图中的#run命令。在该模式图中，程序在运行中使用包括在Instant-C中的标准库。用户有权对这个库进行修改或者甚至使用自己选定的函数库。

至此，已经对Instant-C的工作方式有总的了解。本章其余部分介绍用Instant-C开发程序的方式。

1.3 直接执行模式

启动Instant-C时就是用直接模式进行工作。Instant-C给出磅提示符(#)告诉用户已准备接受来自用户的命令。在IBM PC及兼容机上的屏幕显示如图1-2所示。

```
Instant-C 4.0 188K unused      Level-0, Memory File, •
Instant-C--Version 4.0 ( medium model), September 15, 1988
Copyright 1983, 1985, 1987, 1988 by Rational Systems, Inc.
# _
```

图 1-2 Instant-C启动后的屏幕

当然，这是示意性说明。将屏幕缩短是为了节省空间。Instant-C使有些文字以高亮度或反相视频显示以表示强调。这一类在本手册中就不能加以模拟。但是，插图已很接近实际内容，足以使用户对屏幕所见有明晰印象。整个手册的屏幕插图与图1-2类似。

可利用Instant-C的命令启动编辑程序，执行和调试代码或完成其它各种操作。

在深入了解Instant-C的工作方式以前，首先应该学会如何“读”例子。在本手册中，使用特殊字体以区分键入内容和Instant-C显示内容。输入内容和光标显示如下所示：

Sampl input_

而计算机的输出如下所示：

Sample output from the computer

1.4 对表达式求值

Instant-C能够对任何C语言表达式进行求值并显示其值。例如打入

```
# 5 * ( 3 + 4 )
```

以后，Instant-C的响应是

```
35
```

```
# _
```

这种执行源代码的方法对于学习用C进行程序设计特别有用，因为能迅速看到表达式是否产生预期效果。

可以在直接模式调用任何函数。设计程序时可以用这种方法对函数分别进行测试。这是对isalpha库函数的一种调用：

```
# isalpha ( 'A' )
```

Instant-C的响应是

```
1
```

```
# _
```

然而，如果以大写方法输入该调用：

```
#ISALPHA('A')
```

这时会发生如下出错报文：

```
ERROR 218, I'm sorry, but I don't know the word "ISLPHA" ...
```

```
# _
```

因为C语言对大小写敏感，所以大写字母和小写字母是不同的字符。假如你一直是用BAS-IC编程或使用dBase II 这样的程序（它们视大小写为等同），你会容易犯这种错误。

1.5 自动化编译

对Instant-C来说，完全不需要发布一条命令对代码进行编译。每当从磁盘装入源文件或使用编辑程序进行修改或添加时，编译工作已在自动地进行。

当用户把所作编辑存盘并返回到直接模式时，Instant-C就对经过编辑的源代码进行编译。假如Instant-C发现错误，它会使你返回到编辑程序进行纠正。这一交互过程是Instant-C作为一种增量编译程序的核心：它仅仅是经过编辑的代码进行编译。对未编辑的代码则不必化时间进行重新编译。也就是说，用传统工具需要重复编译这部分时间就可以节省下来。对大文件来说就不必再等待编译即可启动编辑程序纠错。

在某一阶段中，代码以编译形式保存。必要时Instant-C从编译代码重新构成源代码，编辑或列出代码清单并在工作结束时将它存盘。在重建源代码以后，Instant-C可对它作“漂亮打印”。不必对代码作格式化处理而由Instant-C替你做这件工作。如果不喜欢默认格式，可以修改系统选项对它进行控制。

1.6 编辑程序

Instant-C的全屏幕编译程序能迅速而容易地输入和修改程序。Instant-C有几条命令可以启动编辑程序。最常用的一条命令是#ed，它能编辑特定部分代码：函数、申明或定义。

不难区别编辑程序和直接模式。在直接模式，Instant-C给出提示符#并对输入的每一条命令进行处理，而编辑程序使源代码所在的屏幕部分重新设定颜色并使光标定位在顶部。把光标移到需要修改的位置。图1-3是编辑程序显示例子。

```
Instant-ED Copyright (C) 1983-87, 1988 by Rational Systems, Inc.  
INSERT compare (in SORT.C)          245932 line   5 of   9, col 1
```

```
#define ARRAYSIZE 10  
int values[ARRAYSIZE]={8, 2, 9, 7, 1, 4, 0, 6, 3, 5};  
int compare(a, b)  
{  
    int *a, *b;  
    {  
        return(*a, - *b);  
    }  
}
```

图 1-3 编辑程序显示例子

Instant-C在你打入Ctrl-F将代码存盘并返回到直接模式以后才对代码进行处理。因此用户不必调用语法检查程序就能按任何方式自由地编辑代码。当Instant-C发现错误时,会返回到编辑程序并在显示器上部显示出错报文,这时光标处在发现错误的位置上。这样用户可以迅速地消除错误。传统的编程工具需要记下出错报文,启动编辑程序并找到出错的位置,当然这是很慢的。

Instant-C系统提供编辑程序,它可以在IBM PC及兼容机上运行。可以预先配置功能键、光标键以及其它特殊键来完成编辑功能,如移动光标和删除文本。然而,因为每个人对“完善的”编辑程序有各自不同的想法,Instant-C的编辑程序是可充分地进行重新配置的。如果你由于使用另外的程序而使手指习惯于使用某些键,则可以对Instant-C的键重新定义使之与自己的习惯相匹配。

1.7 磁盘文件和内存文件

一个典型的C程序由若干磁盘文件构成,包括源文件和标题文件。

源文件包含程序使用的函数定义以及可能的某些数据申明。它们包含程序的全部可执行代码。它们的文件名往往是以“.C”结尾。

标题文件包含数据申明以及由若干源文件共享的#define函数。每个标题文件由一个或多个带有#include语句的源文件引导。它们的文件名通常以“.H”结尾。

在Instant-C中,要使用#Load命令把源文件从磁盘装入内存。虽然标题文件在正常情况下由源文件中的#include语句自动地装入,它们也可以用上面的同样方法从磁盘装入。

当把文件从磁盘装入内存时,产生的编译后文件称为“内存文件”。它和磁盘文件有相同的名称。可以一次把许多磁盘文件装入内存。装入到一个内存文件的一组目标代码文件(由汇编程序或优化编译程序建立)的大小限于64k字节。然而,可以把目标代码装到多个内存文件中。单个的源文件可以大于64k字节。可以装入Instant-C的代码容量仅仅受计算机内存容量的限制。Instant-C使用中内存模型,允许最大容量为1M字节代码以及64k字节数据,它也支持远指针和大内存模型。

Instant-C总是把一个内存文件指定为“当前内存文件”。它的名称出现在Instant-C的状态行上。虽然可以对其它内存文件中的代码进行编辑,但通常是对当前内存文件进行工作,每当发出命令对函数或其它对象进行操作时,Instant-C首先在当前内存文件寻找该对象;如果找不到,再在它的外部符号表中查找。在直接模式中打入的任何新的申明或函数都加到当前内存文件的末尾。

当启动Instant-C时,当前内存文件称为未名内存文件或“*”内存文件。当开始一新文件时可使用它;将它存盘可赋以自己选定的文件名。

从磁盘装入的文件成为默认当前内存文件。例如,当装入名为GOODTIME.C的磁盘文件时,Instant-C对它的源代码进行编译并将它以文件名GOODTIME.C存入内存文件。GOODTIME.C成为当前内存文件。命令#use可使任何装入的内存文件成为当前内存文件。Instant-C的#new命令可用于建立新的内存文件。如果希望重新开始也可用它清除内存文件。

为了把所做工作存入磁盘,需要把源代码从内存文件转移到相应的磁盘文件。#save命令把内存文件写入磁盘。

你大概已经注意到Instant-C的命令以磅符(#)开始。因为每次打入磅符前缀并无实际意义, 对于多数命令它可以省略。只有当在代码中放入和命令相同的对象名时才必须加上它。例如, 假设你写了一个名叫quit的函数, 退出Instant-C时应打#quit——如只打quit, 就只能得到函数quit的地址。

1.8 运行程序

只要在直接模式打入函数调用就能个别地执行任何函数。如要运行整个程序, 可以发布一个对main函数的调用或发一个#run命令。不必对文件进行连接, 因为Instant-C直接对内存进行编译, 在编译过程中完成全部必要的连接工作。

即使不先装入所有文件就运行程序也无妨。只要不调用不存在的函数, 程序就能成功地运行。如果调用不存在的函数, 只会得到出错报文和提示符#。这时, 可以装入缺失的文件并从停顿处继续执行。

1.9 对代码进行调试

每当程序停止执行时, 会出现提示符#。为了对代码进行调试, 可利用设置断点、对函数跟踪或单步式执行每条语句的方法设计一些中断。也可以在任何时候打Break或Ctrl-C键中止执行。诸如被零除的运行态错误也可以中止执行, 在纠正错误后可以从中断处恢复执行。当程序不是由于调用exit或_exit而停止执行时, 在屏幕上都会出现一个“源窗口”。出错报文让你知道程序为什么停止执行。断点所在的一小段代码也会在该窗口中出现。可以将光标移到该窗口滚动代码或回到当前函数的调用程序。也可以从源窗口直接启动编辑程序而不必从命令行发布命令。

图1-4就是一个源窗口, 它显示对未定义函数调用的一段代码; 用户和Instant-C的对话仍然留在窗口之下。

```
Instant-C4.0 191K unused      Level 1. Memory File: •
• • 447:  undefined function"powerof2" • • in square,line 4
; int square(number)
; int number;
; {
;   printf("The square of %d is %d.\n",number,powerof2(number));
> }
# main ( );
Hello,world!
## run
Hello,world!
# reset
## use •
    • (Scratch)
## ed square
# square (3);
#
```

图 1-4 Instant-C的源窗口

1.10 建立可执行文件

对程序进行工作时，通常只把源代码保存在磁盘上；当离开Instant-C时编译后代码就丢弃。只有在程序进行彻底调试可交付使用时才需要建立可执行磁盘文件。Instant-C的#make命令建立可以从操作系统运行的可执行文件。

另一个选择是利用优化编译程序建立程序的最后工作版本。如果希望使用所带的库是Instant-C支持的编译程序建立程序的生产性版本，可以建立使用该库的特殊版本Instant-C以保证在开发和生产过程中库函数以同样方式工作。

因为Instant-C以源码形式贮存你的代码，只要愿意就可以从操作系统对源文件进行操作。例如，可以使用诸如type或print的操作系统命令看文件，也可以用通常的文本编辑程序对它进行编辑。

1.11 小结

下面是你从本章学到的某些主要概念的复习：

- Instant-C有两个主要的程序环境：直接模式“解释程序”和编辑程序。
- Instant-C利用它的直接执行模式可以对任何C表达式求值。它也执行许多命令。
- Instant-C把编译后代码放在“内存文件”中。只有需要在屏幕上显示它，对它进行编辑或将它存入磁盘时才重建源代码。
- Instant-C自动对你的代码进行编译、连接和格式化。
- 只有当代码调试工作结束以后才需要建立可执行磁盘文件。

现在，你已经具有了可用于学会Instant-C的概念性框架，我们可以在计算机上来安装软件。然后通过学习Instant-C的方方面面获得一些实际的上手经验。

第二章 准备工作

本章帮助你开始使用Instant-C。我们要介绍你能使用的计算系统和需要的文件，给出基本安装指令并讲解如何运行测试程序以保证正确地安装Instant-C。还要介绍如何得到使用Instant-C的帮助以及如何获得本软件的新版本。在本章结尾，可以找到Instant-C发行磁盘上文件的说明。

2.1 系统要求

可以在具有Intel 8086, 8088, 80186, 80286, 82386或兼容微处理器的个人计算机上运行Instant-C。这类计算机包括IBM PC, XT, AT和兼容机。

单用户系统应该至少有512K字节内存，450K以上字节可为程序所用。多用户和多任务操作系统需要多得多的内存以便向Instant-C提供450K内存。Instant-C在操作系统允许情况下可以最多使用1024K内存（可以把IBM PC或兼容机的物理内存扩充到IBM640K限度以上）。

Instant-C可以在PC-DOS或MS-DOS2.0以上版本下运行，也可以在并发PC-DOS下运行。在本手册中，DOS指的是PC-DOS和MS-DOS。

假如你的系统有Intel 8087或80287数值协处理器，Instant-C自动地利用它作浮点计算；否则，Instant-C用软件模拟数值协处理器。

用户只需要一个1.2MB或1.44MB软盘驱动器。因为Instant-C只使用磁盘读写你的C语言磁盘文件，启动Instant-C后即可去掉Instant-C磁盘。这样可用该驱动器访问你的C程序。也可以把Instant-C从软盘拷贝到硬盘后在硬盘上使用它。Instant-C并没有拷贝保护。

2.2 Instant-C安装方法

开始使用Instant-C以前，至少要对Instant-C软盘作两份拷贝。写下软盘标签上的产品号并放在手头（假如你打电话给我们的支持工作人员，会询问这一点）。[务必把你的登记卡寄来以便告知何时会有升级消息]。要把原装盘放在安全地方。Instant-C可以在IBM PC及兼容机上使用。需要访问的只是一个文件，IC.EXE。如果你要在一台非IBM PC兼容计算机上安装Instant-C，需要建立一个特别的IC.EXE，使得它适合计算机屏幕和键盘的配置。所用指令在第八章介绍。

如果你的计算机有硬盘，应把Instant-C拷贝其上使访问更快更容易。把Instant-C软盘上的文件拷贝到名为INSTANTC的目录下（也可以使用其它你喜欢的目录名）。可以从该目录下运行Instant-C。

如果希望在其它目录下运行Instant-C，文件IC.EXE要放在操作系统搜索路径上的目录中。如果你有存放程序的特殊目录，不要把该文件移到目录中。另一种效率稍差的做法是把Instant-C加到操作系统的命令搜索路径。你的操作系统手册提供有关目录和搜索路径的信息。