计算机及软件技术丛书

计算机常用语言混合编程

陈启秀　黄彻为　编著
舒正忠　审校

南京大学出版社

计算机及软件技术丛书

计算机常用语言混合编程

陈启秀　黄彻为　编著
舒忠正　　审校

南京大学出版社
1994·南京

(苏)新登字第 011 号

内 容 简 介

本书介绍了采用多种计算机语言混合编程的基本概念和基本方法。详细介绍了 IBM PC 机系列上最常用的 BASIC、C、FORTRAN、Pascal、汇编以及 dBASE Ⅱ (FOXBASE)语言的接口，归纳了上述语言间相互调用的方法，并以详尽实例说明了整数、实数、字符串、数组、结构和公共块等类型参数在各种语言间的传递。书中配有大量实例，所有实例均上机通过。

本书适合于广大计算机工程技术人员和计算机应用人员作参考书，也可作为高等院校本科生或研究生教材。

计算机及软件技术丛书

计算机常用语言混合编程

陈启秀 董彻为 编著

舒忠正 审校

南京大学出版社出版

(南京大学校内 邮政编码:210008)

南京新利电脑照排中心照排

丹阳练湖印刷厂印刷 江苏省新华书店发行

*

开本:787×1092 1/16 印张: 21.875 字数: 543 千

1994 年 2 月第 1 版 1994 年 2 月第 1 次印刷

印数:1—5000

ISBN 7-305-02432-5/TP · 84

定价: 15.00 元

《计算机及软件技术丛书》编委会

前　言

自第一种计算机高级语言 FORTRAN 诞生以来，至今已有数千种语言问世，人们不禁要问哪种语言最好呢？能否设计一种“最好”的语言取代其它语言呢？曾经有人在这方面做过不懈的努力，然而至今也未能实现这种愿望。人们更清楚地认识到每种计算机语言都有一定的适用范围，各有千秋。将诸多语言相互沟通，彼此调用，取长补短更为明智，这就是本书要介绍的、近年来越来越受到青睐的混合语言编程技术。

混合语言编程是指采用两种或两种以上语言协同编程，相互调用，传递参数，共享数据，完成同一个任务的编程技术。采用混合编程就可以兼用各语言特点和功能之长；充分利用机器资源；继承优秀软件成果；利用现有模块和库程序在短期内开发出新软件，以缩短开发周期，降低成本；高级语言可以借助于汇编驱动和使用各种类型的硬件设备。

本书重点介绍了混合编程的基本概念、原理和方法，以及与混合编程有关的语法规则，详细讨论了在 IBM PC 机系列最常用的 BASIC、C、FORTRAN、Pascal、汇编以及 dBASE Ⅲ (FOXBASE)语言之间，整数、实数、字符串、数组、结构和公共块等类型参数如何传送。如读者对要混合编程的各语言已有基本的了解，对 IBM PC 机系列比较熟悉，知道如何用这些语言编写、编译和连接程序，阅读本书后可以很快掌握混合编程技术。书中附有大量实例，所有实例均在机上调试通过。

本书的内容按一定层次分块组织，查阅非常方便。第一章为混合语言编程技术的综合性介绍；第二章与第七章分别介绍高级语言调用汇编和汇编调用高级语言技术；第三章至第六章分别介绍 BASIC、C、FORTRAN 和 Pascal 相互调用技术；第八章为 dBASE Ⅲ (FOXBASE)与 BASIC、C、FORTRAN、Pascal 和汇编的混合编程技术。

阅读本书时，如涉及 dBASE Ⅲ (FOXBASE)的混合编程，可直接阅读第八章，否则可先选读第一章的第一节至第六节的有关内容，了解混合编程的基本概念、基本约定和接口技术，以及编译连接的注意事项；然后找到调用语言所属的章，阅读其中第一节，这节(除第八章外)介绍该语言调用其它语言的基本技术和方法；最后依参数传递类型，选读第一章后半部分与该类型数据存贮方式有关小节，以及该章中如何传递这类参数的小节，这些小节依所传递的数据类型按整数、实数、字符串、数组、结构、公共块的次序安排，读者可以很方便地找到它们。

本书由陈启秀和黄彻为共同执笔。韩亚萍和李来喜同学在毕业设计期间，参加了程序的调试工作。全书由舒忠正副教授进行审阅。本书成书过程中得到了南京航空航天大学领导的关心，得到了南京航空航天大学科研处的大力支持，得到刘迪吉、罗维忠、屈锐、张丽维、林钧海、沈孟涛、奚抗生、丁秋林、李定建、张丹等同志的支持和帮助，在此表示深切地感谢。

由于编写时间仓促以及编者水平有限，书中错误和不妥之处在所难免，敬请读者不吝指正。

编　者

1993 年 7 月于南京

目　录

绪　言

自从50年代中期第一种计算机高级语言FORTRAN诞生以来，许多高级语言纷纷问世，至今世界上已有数千种了。哪一种最好呢？能不能设计一种最好的来取代其它计算机语言呢？60年代有人试图设计一种功能最强的最完善的语言——PL/1语言，未达预期效果，此后计算机语言又向专业化方向发展，各语言都有自己的适用范围，都有优点，同时也有其不足。

例如，FORTRAN标准化程度高，通用性好，速度快，数值计算的能力非常强，特别适合于科学和工程计算。FORTRAN 77还增加了字符数据的处理功能，但FORTRAN不具备结构类型的数据，从而限制了它在事务处理中的应用。Pascal具有丰富的数据类型，简明灵活的通用语句，清晰明了的模块结构，以及编译紧凑方便，书写格式自由和运行效率高等优点，但Pascal的字符处理功能不如FORTRAN强，变量类型和过程参数类型无缺省值，不同类型数据混合运算也不够方便。C是面向结构的程序设计语言，通用性高，可以直接读入并处理字符串、数字和地址，提供了处理二进制位和字节的手段，可以完成通常由硬件实现的算术和逻辑运算，可以用"结构"定义出任意复杂程度的数据结构，并有汇编的某些长处，生成代码质量高，可移植性好，但运算优先级太多，不便记忆，类型检验太弱，转换比较随便，错误检验能力较差（例如不能检查出在调用函数时变量的个数是否恰当），而且字符处理也不如FORTRAN方便。BASIC是简洁清晰指令较少的语言，提供了友善的程序设计环境，具有较强的字符串处理和文件处理功能，但缺乏结构化的控制结构，速度较其它高级语言要慢，限制了它在大多数事务处理中的应用。汇编语言占用存储空间少，执行速度快，可以方便地直接访问许多硬件设备，可以准确地知道程序执行时间，在许多地方必不可少，然而编写难度大、周期长。dBASE Ⅲ因操作方便、使用灵活、通俗易懂、具有较强的数据处理能力，而得到广泛应用，然而计算能力较弱，对于较复杂的数字计算（如三角函数计算、矩阵运算等）和大量的数据管理，处理速度慢，算法不灵活，难于适应；并且缺乏图形功能，报表输出功能亦不够理想，而这些恰恰是高级语言的长处。

倘若把各种语言结合起来，互通有无，取长补短，融为一体，就能及时解决单独一种语言不能解决的某些问题，并兼用各种语言的特点和功能之长，于是混合语言程序设计技术诞生了。

混合语言程序设计是指采用两种以上的语言组合编程，相互调用，进行参数传递，共享数据，共同完成一个任务，形成一个软件整体的编程技术。采用混合语言编程有利于充分利用机器资源和继承已有优秀软件成果，可以利用各领域目前所使用的程序模块和目标库程序模块，在短时间内组成新的应用程序，大大缩短了开发时间，降低了成本，提高了软件效益，混合语言程序设计诞生以后，受到了广大编程者的青睐，得到越来越广泛的应用，并形成了程序设计的一个新的研究方向。

从数据传递的角度来看，混合语言编程有如下几种：

1. 调用型。即一种语言调用另一种语言编写的例程。大多数语言混合编程采用这种方法，

其特点是例程间参数传递数据量很小。

2. 公共数据共享型。两种不同语言编写的例程间可以不存在调用关系，但一种语言中定义了另一种语言例程也可以访问的数据块，形成共享数据区。其特点是共享数据量可以比较大，例如共享公共块、引用外部变量就是这种类型。

3. 文件格式转化型。不同语言编写的例程间可以不存在调用关系，各自使用自己的文件对数据区访问，不同的语言其文件格式是不同的，一种语言需要另一种"协助"时，把自己的文件格式转化成另一语言的文件格式，再交给对方处理，对方也采用相同的方法，例如 dBASE Ⅲ (FOXBASE)和其它高级语言混合编程常采用这种方法，参数传递量很大。

4. 直接访问其它语言文件格式型。一种语言直接访问另一种语言的特殊格式文件。例如 BASIC 以及其它高级语言直接访问 dBASE Ⅲ (FOXBASE)的. DBF 文件。

5. 宿主型。混合语言通常是采用各自编译形成目标码，再用连接程序连接成可执行文件。然而宿主型却将一种语言(宿语言)程序先编译成另一种语言(主语言)程序，经主语言的编译器编译连接后再执行，如 ORACLE 和 C 就是如此。

本书介绍了混合语言程序设计的原理和各种方法的实现，并以 Microsoft BASIC、C、FORTRAN、Pascal、宏汇编以及 dBASE Ⅲ (FOXBASE)为例，介绍了如何在各模块之间建立接口，如何传递各种类型的参数，所有例子均在上述语言中上机通过。本书第一章介绍了混合语言程序设计的基础，以后各章分别介绍一种语言调用其它语言的基本要求和具体实例，各章节可根据混合编程所用的语言选读。

第一章　混合语言程序设计基础

混合语言程序设计最常用的方法是不同语言的多个模块(模块今后如不作特殊说明是指可以独立编译的程序文件)在同一环境下的相互调用。关键问题之一是建立语言之间的接口，接口主要表现在语言的命名约定、调用约定和参数传递约定，调用者通常在接口语句中进行说明这些约定，而被调用者通常在子例程定义语句和有关变量(形参)说明中进行说明，以便在两种语言间提供有效的通讯，不同语言的各模块编程完成后要用各自的编译程序分别编译，然后把目标文件连接起来。注意，在编程时调用和被调用例程要遵守一致的命名约定、调用约定和参数传递约定；编译时要选用适当的存储模式；连接时按正确的顺序连接。

本章的前 6 节是混合语言程序设计的基础，如只进行整数、实数等简单类型参数传递，只需看这 6 节即可；后 5 节是字符串、数组、结构和公共块等特殊类型参数传递的基础，编程时如涉及这些类型参数传递，可选读有关小节。

1.1　混合语言编程的一个例子

混合语言程序设计常涉及不同语言间例程(本书中的“例程”是指可由其它模块调用的函数、过程或子程序)相互调用，调用一般涉及到多个模块。相同语言编写的程序用同一个编译程序对所有源程序模块编译，然后对目标文件进行连接，混合语言使用不同的编译程序，然后再连接目标文件。图 1－1 给出了 BASIC 主程序调用 C 函数的例子。

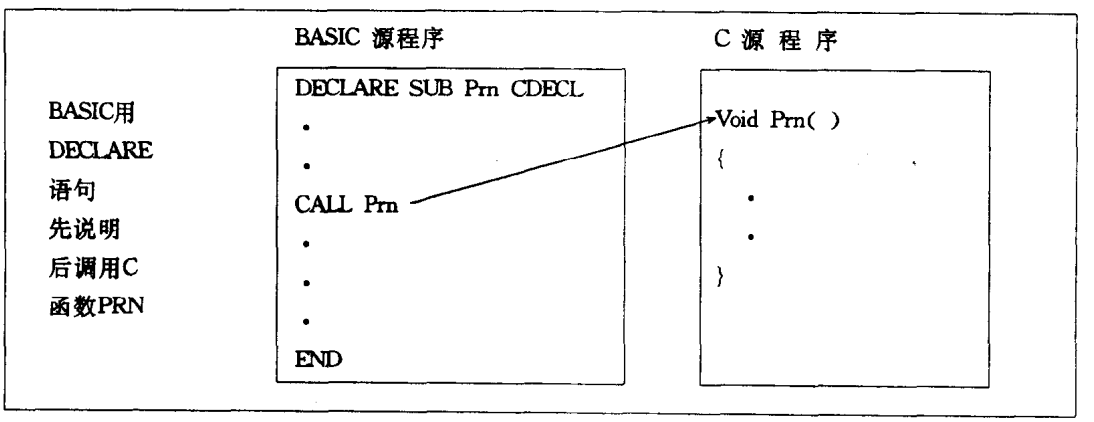

图 1－1　BASIC 主程序调用 C 函数的例子

BASIC 主程序调用 C 函数的语句是 CALL Prn(类似于调用 BASIC 子程序),但 BASIC 调用 C 函数与两个 BASIC 模块间的调用有两点不同:

1) 子程序 Prn 是使用 C 语言编写的空函数(没有返回值的函数),而不是 BASIC 子程序。

2) BASIC 要调用 C 函数 prn( ),必须在调用前用 DECLARE 语句对函数 prn() 进行外部调用说明,并使用关键字 CDECL 以建立与 C 一致的约定。DECLARE 语句(在 1.3 节详细讨论)是 BASIC 提供的支持混合语言的一种"接口"语句,每种语言都有自己的接口语句。

各种语言的函数、过程和子程序的名称不同,语法也有差异,其本质区别就在于有些类型的例程有返回值,而有些则没有。表 1.1 给出各语言间例程的对应关系。

表 1.1 各语言间例程的等价关系

语 言	有返回值	无返回值
BASIC	函数	子程序
	过程	
C	函数	void(空)函数
FORTRAN	函数	子程序
Pascal	函数	过程
宏汇编程序	过程	过程

例如,若 BASIC 模块要调用 FORTRAN 例程以获取函数返回值,则应调用函数 FUNCTION,而不能调用子程序 SUBROUTINE,否则虽可以调用,但会丢失返回值。

另外 BASIC 的 DEFINE 函数和 GOSUB 子例程不能被其它语言调用。

1.2 接口的三个基本约定

混合语言接口有三个基本约定,即命名约定、调用约定和参数传递约定,这三个约定是混合语言程序设计的核心,调用和被调用例程都必须遵守一致的约定。

1.2.1 命名约定

Microsoft 编译程序把机器码放入目标文件的同时,把所有需要共同访问的例程名和(外部)变量名以 ASCII 码的形式放入目标文件。

一、什么是命名约定

"命名约定"就是指编译程序在把名字放入目标文件之前,改变名字的一种约定。此约定是为多种语言对名字访问而设置的,它体现在对目标名字的长度限制和大小写等规定。

进行混合语言调用时,应采用一致的命名约定。连接程序将目标模块中所调用的例程名和外部变量名与被调用模块中定义的例程名和被引用的变量名相匹配,若成功则进行连接,否则报告未解决的外部引用。为了察看到名字如何存放,可借助于 DEBUG 调试程序。DEBUG 在

调试目标文件时，可从装载该文件后的内存空间 0—400H 内获得以 ASCII 码形式存放的外部例程名和外部变量名，清楚地看到它们的具体存放形式。

二、命名约定的具体内容

1. BASIC、FORTRAN、Pascal 和宏汇编的命名约定

BASIC、FORTRAN、Pascal 和宏汇编的命名约定大致相同，均将名字字母转换为大写，BASIC 还将名字所带的类型说明字符(%、&、!、# 和 $ 分别为 2 字节整数、4 字节整数、4 字节实数、8 字节实数和字符串的类型说明符)删除。但这些语言识别并放入目标文件的名字字符长度有所不同，FORTRAN 识别名字的前 6 个字符，Pascal 是前 8 个，BASIC 是前 40 个，而宏汇编为前 31 个。如名字长度超过规定的字符，多余的字符不放入目标文件。

2. C 的命名约定

C 的命名约定与前者不同，它不将名字字母转换为大写，并把下划线"_"插入到名字之前，C 识别名字的前 8 个字符。

例如源程序的名字为 FirstNumber，FORTRAN 放入目标文件的是 FIRSTN，Pascal 为 FIRSTNUM，BASIC 为 FIRSTNUMBER，宏汇编为 FIRSTNUMBER，而 C 放入目标文件的却是_FirstNum。

三、采用一致的命名约定

从以上叙述可知各语言间存在命名约定差异，在进行混合语言编程时，如何解决这个差异呢？

混合语言的关键字如 BASIC 的 DECLARE 语句中的 CDECL，FORTRAN 的 INTERFACE 语句中的[C]，Pascal 的 EXTERN 语句中的[C]，C 的 EXTERN 语句中的[Pascal]或[Fortran] 会对命名约定进行调整，使不同的语言采用一致的命名约定。

命名字时，须遵循以下两个规则：

(1) 如果使用 FORTRAN 编程，所有名字长度最好不超过 6 个字符，否则要使用关键字 ALIAS。

(2) 连接时不要使用/NOIGNORE 任选项(它使连接程序区别名字中字母大小写，把 PRN 和 prn 认为是不同的名字)。

图 1—2 给出了一个完整的混合语言开发例子的示意图，BASIC 用 DECLARE 语句说明要调用外部子程序，关键字 CDECL 指示 BASIC 编译采用 C 的命名约定。

在图 1—2 的例子中，BASIC 用 DECLEAR 语句说明了要调用外部子程序(无返回值) Prn，因为在 DECLEAR 语句中用 CDECL 关键字指示 BASIC 编译程序使用"C"的命名约定，所以 BASIC 编译程序把名字放入目标文件时，将所有字母转换为小写，并在 Prn 之前插入了引导的下划线，而真正的 C 的命名约定并不转化名字字母的大小写，编写 C 语言程序时习惯采用小写。

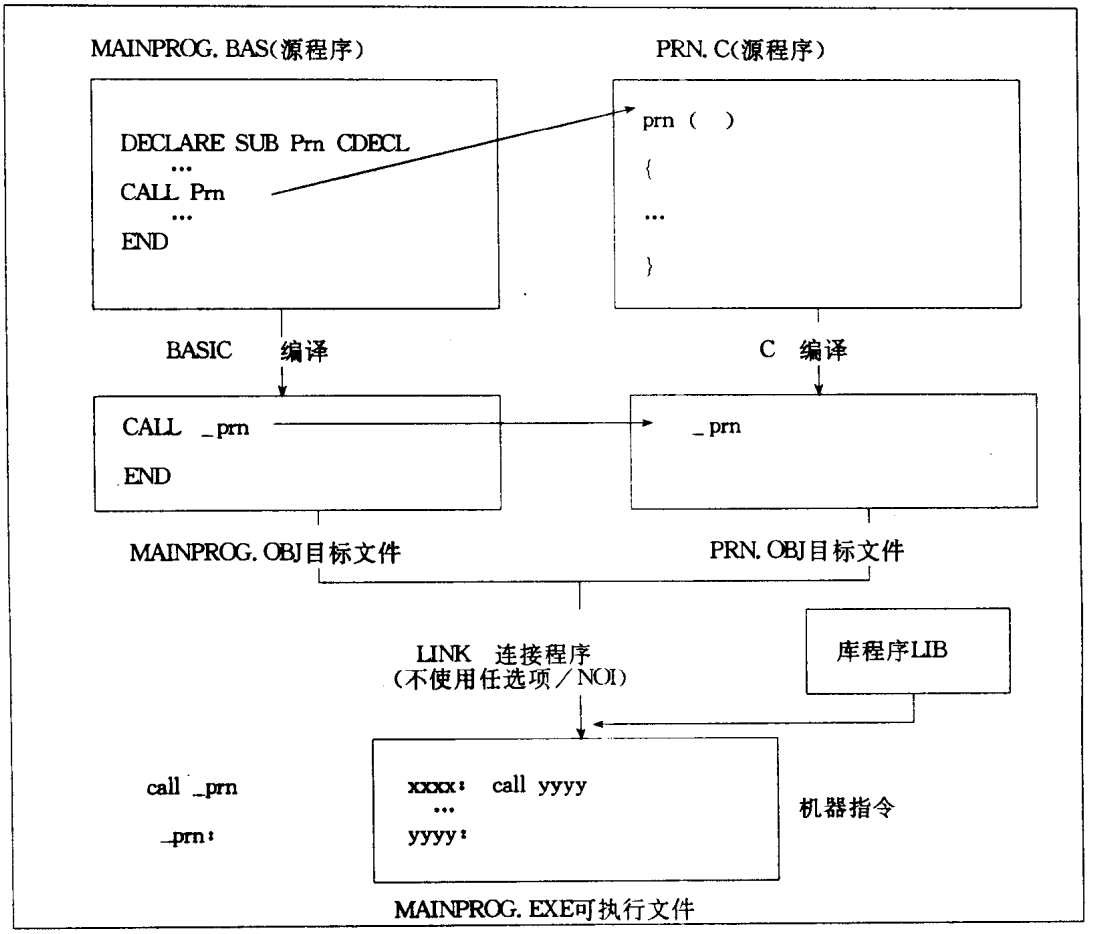

图 1－2 BASIC 调用 C 混合编程流程图

1.2.2 调用约定

一、什么是调用约定

调用和被调用例程间的参数传送一般通过堆栈进行。“调用约定”有两方面的内容：

(1) 调用者以什么次序把参数压入堆栈和被调用者以什么次序从堆栈中取参数。

(2) 从被调用例程返回时，是由被调用例程还是由调用例程释放堆栈中参数区。

调用约定的选择会影响编译程序为调用和被调用例程生成的机器指令。调用的和被调用的例程须遵守一致的调用约定。

调用约定在两方面影响高级语言混合程序设计：

(1) 确定调用例程以什么次序“发送”参数到被调用例程，调用者通常在自己的接口语句中说明采用什么样的调用约定。

(2) 确定被调用例程以什么次序“接收”传来的参数，被调用例程通常在子例程的定义语句中说明采用什么样的调用约定，但 BASIC 总使用自己的调用约定接收参数。

二、调用约定的具体内容

Microsoft BASIC、FORTRAN 和 Pascal 的调用约定相同，都是调用者依参数出现在源程序中从左到右的次序把参数压入堆栈，被调用者以相应的次序从堆栈中取参数。如 BASIC 在执行调用语句 CALL TEST(A,B)时，按 BASIC 的调用约定，A 先进栈，B 后进栈，并由被调用例程在返回调用程序前恢复堆栈(被调用例程用 ret n 指令释放 BASIC 传来的 n 个字节参数，恢复 BASIC 例程堆栈再返回)。

C 的调用约定以参数出现在源程序中从右到左的次序把参数压入堆栈，被调用者以相应的次序从堆栈中取参数。如 C 调用函数 TEST(A,B)时，先把 B 压进堆栈再压入 A，并约定被调用例程用不带参数的 ret 指令返回 C，返回 C 后由 C 立刻释放它传给被调用例程的参数区，恢复堆栈。

混合编程时，调用例程在接口语句中说明自己的调用约定；而被调用例程在"定义"语句中说明，如不作说明，则采用事先约定好的调用约定，调用者和被调用者所采用的调用约定必须一致。C、FORTRAN 和 Pascal 都可以改变自己的调用约定(这包括作为调用者按什么次序发送和作为被调用者按什么次序接收，以及由调用者还是被调用者释放参数区两个方面)，BASIC 作为调用者可以改变调用约定，作为被调用者却只能按自己的调用约定。通常采用被调用例程的约定较简单。调用约定在如何从被调用例程返回方面影响实际产生的目标码，这在高级语言相互调用时程序员是看不到的，但高级语言和汇编相互调用时必须搞清楚。

BASIC、FORTRAN 或 Pascal 的调用约定产生较少的目标码，但是 C 的调用约定能以可变数目的参数进行调用，因为第一个参数总是最后一个进栈，放在堆栈的顶部，不管传送了多少参数，它相对于帧指针(帧指针的定义参见 2.1.2 节)的地址偏移是相同的，第一个参数相对帧指针的地址偏移只和编译所选的存储模式有关而和参数个数无关。

三、采用一致的调用约定

从以上叙述可得知各语言间存在调用约定差异，在进行混合语言编程时，如何解决这个问题呢？

混合语言的关键字如 BASIC 的 DECLARE 语句中的 CDECL，FORTRAN 的 INTERFACE 语句和子例程定义语句中的[C]，Pascal 的 EXTERN 语句和子例程定义语句中的[C]，C 的 EXTERN 语句和函数定义语句中的[Pascal]或[Fortran]会对调用约定进行调整，使不同的语言采用相同的调用约定。

1.2.3 参数传递约定

参数传送分为按值(又称值参传送)和按引用传送(又称按地址传送或变参传送)两大类，在 Microsoft 中又把按引用传送分为按近引用(近地址)和按远引用(远地址)。这样传送参数就有三种方法：

1. 按近引用传送

传送参数的近地址(地址为二个字节的段内偏移量)。该方法使被调用例程直接存取参数自身，被调用例程对参数的改变将反映到调用例程中。

2. 按远引用传送

传送参数的远地址(地址为四个字节，二个字节为段地址，另外二个字节为段内偏移量)。该方法除了传送较长的地址、速度较慢之外，类似按近引用传送。

3. 按值传送