

如何在 C 语言程序中调用汇编语言

其其格

(内蒙古师范大学数学系, 内蒙古 呼和浩特 010022)

摘 要: 本篇文章论述了 C 语言程序中调用汇编语言的问题。在 C 程序中调用汇编语言有两种方法。一种是在 C 程序中直接嵌入汇编语言代码。另一种是调用汇编语言子程序的方法。

关键词: C 语言; 汇编语言; 混合编程

中图分类号: TP 313 **文献标识码:** A **文章编号:** 1007—6921 (2002) 09—0078—02

C 语言是具有高级语言和低级语言特点的中级语言。它可以完成许多由汇编语言完成的工作。但在有些对硬件和操作系统直接操作的场合, 在有些要求执行速度较快的场合, 仍要用到汇编语言。这可以由 C 程序直接调用汇编语言程序来实现。这种调用可以用两种方法进行。第一种是在 C 程序中直接嵌入汇编代码, 这是一种当汇编程序较短时可采用的方法。另一种是直接调用汇编语言子程序, 这是一种较广泛使用的方法。

1 C 语言程序中直接嵌入汇编语言代码

在 C 程序中嵌入的汇编语言指令就好像 C 程序的语句一样。当 C 语言程序中想直接控制硬件或加快运行速度, 但可用较短的汇编程序能实现时, 可以用这种方法。

1.1 实现的方法

在嵌入的汇编指令前, 必须用关键字 `asm` 说明, 其格式为:

`asm < 操作码 > 操作数 < ; 或换行 >`

其中操作码是有效的 8086 指令或汇编伪指令 `db`, `dw`, `dd` 及 `extern` 等。操作数可以是 C 语言中的常数、变量和标号, 也可以是操作码可以接收的数, 内嵌汇编指令用分号或换行作为结束标志。需要注意的是, 不能像 MASM 一样, 用分号作为一条注释的开始, 注释必须用 `* * *`。一条汇编指令不能跨两行, 但允许一个行中可有多条汇编指令, 中间用分号隔开。

1.2 内嵌汇编指令的 C 程序函数

实际应用中经常使用内嵌汇编指令的 C 程序函数。这种函数是 C 语言函数, 但函数体完全由汇编指令完成。当写成 C 程序函数形式时, 它可以在各种编译模式下均通过, 如同用 C 语句编写的函数一样。

1.3 举例

下面是两个变量交换的程序文件 `file.c`。交换由函数 `exchange` 来实现。

```
file c
main ()
{void exchange (int a, int b);
int x, y;
scanf ("% d,% d ", &x, &y);
printf ("x=% d,y=% d ", x, y);
exchange (x, y);
printf ("x=% d,y=% d ", x, y);}
void exchange (int a, int b)
{asm mov ax, a
asm mov bx, b
asm mov a, bx
asm mov b, ax }
```

1.4 内嵌汇编指令的 C 程序编译连接方法

内嵌汇编指令的 C 程序只能采用 TCC 命令行的编译连接方法, 用 TCC 命令行实现编译连接的格式是 `TCC - B - L: LIB 文件名 库文件名`

其中 `-L` 选项指定了连接所需的库文件路径, 库文件名指程序中要用到的库函数所在的库文件。Turbo C 标准库可以省略。 `-B` 选项说明所编译连接的程序中有汇编指令。所以内嵌汇编指令的 C 程序进行编译连接时, 必须有 `-B` 选项。在 C 程序中加上 `# program in line` 时可省略 `-B` 选项。所以, 要对上面的例子进行编译连接, 只要在 DOS 下 `TCC - B file.c` 即可。

2 直接调用汇编语言子程序

C 程序中调用的汇编指令较多时经常使用直接调用汇编语言子程序的格式。由于 C 编译系统要求约定的段序, 要求规定的段组结合, 故要编制能被 C

调用的汇编子程序,则要严格按照C的约定来设计程序的结构,否则不能被正确调用。

2.1 C 编译系统将不同类型的变量保存在不同的几个段内

其具体情况如下:

BSS 段——存放未初始化的静态变量

DATA 段——存放所有的全局变量和已初始化的静态变量

CONST 段——存放只读常数

STACK 段——存放自动变量和函数参数

TEXT 段——存放程序的执行代码

而且,经常将 DATA , CONST , BSS 和 STACK 段组合成一个DGROUP 段组。

2.2 C 编译系统对不同的段的段名、边界类型、结合类型及类型规定了统一的命名规则

段名	边界类型	结合类型	类型
TEXT	BYTE	PUBLIC	'CODE '
DATA	WORD	PUBLIC	'DATA '
BSS	WORD	PUBLIC	'BSS '
CONST	WORD	PUBLIC	'CONST '
STACK	PARA	STACK	'STACK '

2.3 可被 C 调用的一般汇编子程序结构

```
code SEGMENT BYTE PUBLIC 'CODE '
    ASSUME CS: code ,DS:< dseg>
    代码
code ENDS
dseg GROUP DATA , BSS
data SEGMENT WORD PUBLIC 'DATA '

data ENDS
BSS SEGMENT WORD PUBLIC 'BSS '

BSS ENDS
END
```

在该结构中 code , data , dseg 要根据存储模式,换成相应的名字,按 Turbo C 的规定,必须按如下约定替换:在微小紧凑模式下,code —— TEXT、data —— DATA、dseg ——DGROUP。在中大模式下 code —— 文件名 TEXT、data —— DATA、dseg ——DGROUP。在巨模式下 code —— 文件名 TEXT、data —— 文件名 DATA、dseg —— 文件名 DATA。

该结构中未定义CONST 段和 STACK 段,因

C 调用汇编子程序时,常量和局部变量及参数均放在 C 的 CONST 段和 STACK 段中,故在汇编子程序中无需再设置CONST 段和 STACK 段。若需要也可以设置。实际上汇编子程序可根据需要设置,仅有代码段也可以。

2.4 举例

下面是在小模式下的混合编程例子。该文件有 min. c 和 min num. asm 两个部分组成。功能为从 5 个数中找出最小的。min. c 是主程序,由 C 语言编写。min num. asm 是汇编子程序,完成寻找小数的

工作。由min. c 来调用它。程序代码如下:

```
min. c
int extern min_num (int count, int x1, int x2, int x3, int x4, int
x5);
main ()
{int i;
i= min_num (5,7,0,6,1,12 );
printf ("The minimum of five is %d \n", i);}
min num. asm
TEXT SEGMENT BYTE PUBLIC 'CODE '
PUBLIC min_num
min_num proc far
ASSUME CS: TEXT
push bp
mov bp, sp
mov ax,0
mov cx,[ bp+4]
cmp cx, ax
jle exit
mov ax,[ bp+6]
jmp ltest
comp: cmp ax,[ bp+6]
jle ltest
mov ax,[ bp+6]
ltest: add bp,2
loop comp
exit: pop bp
ret
min_num endp
TEXT ENDS
end
```

2.5 混合编程需要注意的几个问题

2.5.1 在 C 程序中可以调用汇编子程序和汇编语言中定义的变量。但汇编语言中需用 public 说

用 Excel 求解线性方程组并验证是否正确

张 聪 史文红 王丹阳

(包头师范学院计算机系, 内蒙古 包头 014030)

摘 要: 本文介绍了利用 Excel 公式复制求解线性方程组, 并验证是否是方程的根。

关键词: 公式复制; 线性方程组

中图分类号: O 241.6 文献标识码: A 文章编号: 1007—6921 (2002) 09—0080—02

Excel 是一款具有强大电子表格功能的软件。电子表格实际由行与列组成的矩阵构成。矩阵中的每一个元素都作为一个存储单位, 它可以是数值、变量、字符或公式。有了这个条件, 我们就可以将线性方程组的系数按顺序分别放入相应的单元格中, 利用高斯消元法和 Excel 的公式复制功能就很方便地将线性方程组转化成等价的三角形方程组, 从而求出方程组的解。我们还可以根据题目要求对方程组的解进行验算或增减小数位数, 得到任意精度的计算结果。下面是一线性方程组。

$$10x_1 - 7x_2 + x_4 = 8$$

$$-3x_1 + 2x_2 + 6x_3 + 2x_4 = 5$$

$$5x_1 - x_2 + 5x_3 - x_4 = 5$$

$$2x_1 + x_2 + 2x_4 = 1$$

求此方程组的解

1 输入线性方程组的线性系数

表 1

	A	B	C	D	E
1					
2					
3	10	-7	0	1	8
4	-3	2	6	2	5
5	5	-1	5	-1	5
6	2	1	0	2	1

明, 且函数名和变量名前应带下划线。而在 C 语言中应将其说明为 extern。如上面例子中的 min num 函数。

2.5.2 汇编子程序中也可以调用 C 语言函数和变量。但调用时函数名和变量名前加下划线。

2.5.3 C 调用汇编子程序时, 参数通过堆栈传递给汇编程序。参数从左到右的顺序入栈。

2.5.4 汇编子程序的返回值通过 AX 和 DX 寄存器进行传递。返回值为 16 位时用 AX 寄存器, 32 位时高 16 位放在 DX 寄存器中, 低 16 位放在 AX 寄存器中。

2.6 编译连接方法

2.6.1 在 DOS 下, 用宏汇编对汇编子程序进行汇编, 生成后缀为 .obj 的目标文件。

2.6.2 在 TC 下, 建立工程文件。可以用 file 菜单下的 new 命令。工程文件内容为需编译连接的 C 程序文件名和它调用的汇编子程序的目标文件名。然

后将建立好的工程文件保存好。文件名的后缀必须是 .prj。

2.6.3 在 TC 下, 选择 project 菜单下的 project name 命令。输入刚建立好的工程文件名。

2.6.4 用 F9 键对工程文件进行编译连接。生成一个 .exe 可执行文件。

【参考文献】

- [1] 王士元. C 高级实用程序设计. 清华大学出版社, 1996, 3
- [2] 尹彦芝. C 语言高级实用教程. 清华大学出版社, 1992, 10
- [3] 谭浩强. C 程序设计. 清华大学出版社, 1999, 12
- [4] 沈美明, 温冬婵. IBM—PC 汇编语言程序设计. 清华大学出版社, 1993, 6

收稿日期: 2002 年 5 月 20 日