

汇编语言与 C 语言及 Visual C++ 混合编程

翟乃强, 隋树林

(青岛科技大学 信息与控制工程学院, 山东 青岛 266042)

摘 要: 汇编语言开发的程序代码短、执行速度快, 而 C 和 C++ 语言是应用最广泛的高级开发语言。在编写硬件驱动, 工业实时控制以及大量的计算方面, 利用 C 或 C++ 与汇编语言混合编程可以提高程序的效率。本文给出了混合编程实现的具体步骤, 说明了混合编程需要注意的要点。

关键词: 汇编语言; C 语言; Visual C++ 6.0; 混合编程

中图分类号: TP 311.52 **文献标识码:** A

Assembly Language Mixed with C and Visual C++ Language for Programming

ZHAI Nai-qiang, SUI Shu-lin

(College of Information and Control Engineering, Qindao University of Science and Technology, Qingdao 266042, China)

Abstract: Assembly language has the advantages of less code, higher speed. C and C++ languages are advanced languages and are used widely. Assembly language, C language and Visual C++ mixed programming can improve programming efficiency in developing hardware drivers, real time control and mass calculation. This article gives the approach of mixed programming and explains the important aspects.

Key words: assembly language; C language; Visual C++ 6.0; mixed programming

汇编语言是直接操纵硬件的, 是除了机器语言以外最高效的计算机语言, 在某些方面汇编语言仍然具有强大的生命力, 但用汇编语言开发程序, 开发周期长, 容易出错且不易调试。要想既利用汇编语言代码短、执行速度快的特性, 又能快速的开发出实用的应用程序, 就需要汇编语言与高级语言混合编程。在高级语言中 C、BASIC、FORTRAN、PASCAL 均可以和汇编语言混合编程, 其中由于 C 语言独特的优点和其应用的广泛性, 使其与汇编语言的混合编程有实际意义。

1 汇编语言与 C 语言的混合编程

汇编语言和 C 语言的混合编程可以分为汇编语言程序调用 C 函数, C 程序调用汇编语言函数和 C

程序内嵌汇编指令三种。前两种要求 C 程序与汇编程序之间必须符合函数变量引用的规则, 并且随着汇编程序和 C 程序内存模式的不同, 规则也相应的变化^[1]。这里只介绍第三种: C 程序内嵌汇编指令。一般在 C 程序中想直接控制硬件或加快运行速度, 用较短的汇编程序实现时, 可以采用这种方法, 它的实现是:

(1) 在嵌入的汇编指令前必须用关键字 asm 说明 asm <操作码><操作数><; 或换行>。其中操作码是有效的 8086 指令及某些汇编伪指令 db,dw,dd 及 extern。操作数可以是 C 语言中的常量、变量和标号, 也可以是操作码可接受的数。内嵌的汇编指令用分号或换行作为结束, 需要注意的是不能像 MASM 中那样用分号作为注释的开始, 必须用 C

收稿日期: 2002-05-28

作者简介: 翟乃强 (1977~), 男, 硕士研究生; 指导教师: 隋树林 (1958~), 男, 教授

方式即/*.....*/来标出注释行。一条汇编指令不能超越两行,但同一行中可有多条汇编指令。

例如:

```
asm mov ax,ds; /*ds->ax*/
asm push ds;
```

或者用换行结束也可以,如下:

```
asm mov ax,ds
asm push ds
```

(2) 内嵌汇编指令中的操作数。操作数可以是C语言程序中的符号,编程时它会自动转换为适当的操作数。当汇编指令中使用寄存器名时,则不分大小写,并只能是8086的寄存器名,在C语言中,DI和SI两个寄存器常被用来存放Register变量,只有在C程序中没有指定Register变量时,汇编程序才可以任意使用这两个寄存器。

(3) 汇编指令操作数可以是结构数据。

例如:

```
struct student
{
    int a;
    int b;
}stu1;
```

在汇编指令中也可以用结构体的成员作为操作数

```
asm mov ax,stu1.b
```

(4) 转移指令的执行。

内嵌汇编指令可以用任何有条件和无条件转移汇编指令,他们只能在函数体内有效,不允许段间转移。由于在asm语句中无法给出标号,所以转移指令只能使用C语言中的goto语句用的标号。例如:

```
asm jmp exitfunc
...
exitfunc:
return 1;"
```

(5) 程序编译。

C程序中内嵌汇编指令时,编译连接只能用TCC命令行,而且必须有-B选项。下面给出一个较为完整的例子:

```
main()
{
    int i,a,b;
    printf("Input two int num:\n");
    scanf("%d %d",&a,&b);
```

```
i = max ( int v1 , int v2 );
printf("the maximum is %d",i);
}
```

```
int max ( int v1 , int v2 )
```

```
{
    asm mov ax,v1
    asm cmp ax,v2
    asm jge exitfunc
    asm mov ax,v2
```

```
exitfunc:
```

```
return ( _AX );
```

/*或者直接return;也可以,因为默认返回的是AX寄存器的值*/

```
}
```

注意:

(1) C程序调用汇编中的寄存器时,要用大写字母且前面加一个下划线。

(2) 在编译之前从TC下include文件夹中把stdio.h和stdarg.h复制到TC目录下,把lib文件夹中的c0s.obj,cs.lib,emu.lib和maths.lib复制到TC目录下。

(3) 若TC文件夹中没有TASM.EXE可以把微软的MASM.EXE复制到TC目录下并且改名为TASM.EXE。

(4) 编译时只能在纯DOS方式下(不是TC的SHELL OS),用TC中的TCC进行编译连接^[2]。编译时在命令行键入:

```
C:\TC\TCC -B asm.c
```

可以生成文件asm.bak、asm.obj、asm.map、asm.exe。其中asm.exe是得到的可执行文件。

2 汇编语言与VC++ 6.0的混合编程

VC++ 6.0是微软公司为开发32位应用程序而推出的基于C\C++的集成开发环境。他的强大功能使它成为最优秀的开发工具之一。他同样也支持内嵌汇编指令的混合编程,并且VC++ 6.0的编译器中已经集成了汇编指令编译器,使得编译连接非常方便^[3]。他的实现是:

(1) VC++ 6.0是微软的C\C++,他有区别于ANSI C和ANSI C++,在VC++中启用汇编语句的关键字是_asm或者__asm而不能用asm。

(2) _asm不能单独出现,他后面必须有汇编指令或者是由大括号括起来的多条指令。

例如: `_asm mov al, 2`
`_asm mov dx, 0xD007`
`_asm out al, dx`

或者

```
_asm
{
    mov al, 2
    mov dx, 0xD007
    out al, dx
}
```

(3) 在 `_asm` 所带的一组汇编指令中可以有标号, C/C++ 中的 `goto` 语句和汇编指令的跳转语句可以跳到汇编指令组中的标号处, 也可以跳转到汇编指令组外的标号处。例如:

```
void func ( void )
{
    goto C_Dest;
    goto A_Dest;
    _asm
    {
        jmp C_Dest
        jmp A_Dest
        A_Dest:
    }
    C_Dest:
    return;
}
```

(4) 在 `_asm` 所带的汇编语块中只能调用没有重载的全局的 C++ 函数, 也可以调用声明为 `extern "C"` 类型的函数。因为 C 标准库的函数全部声明为这种形式, 所以 C 标准库函数都可以调用。

(5) 对于类、结构体和共用体的成员变量, 汇编指令可以直接使用。这与 C 语言是一样的。一般来说, 在 VC++ 中插入汇编行主要用于与硬件联系紧密地软件开发和工业实时控制, 以及大量的计算等方面, 这样可以显著地提高程序的运行速度。下面是进行十亿次减法运算, 用汇编与 C++ 进行的比较。

用 C++ 进行 10 亿次减法运算的循环如下:

```
long nums2=10000;
long nums3=100000;
while(1)
{
```

```
    if(nums3==0) break;
    if(nums2==0)
    {
        nums2=10000;
        nums3--;
    }
    nums2--;
```

相同的计算改为内嵌汇编语句的代码如下:

```
long nums2=0, nums3=0;
while(1)
{
    _asm
    {
        mov ebx, 100000
lp2:    mov ecx, 10000
lp1:    loop lp1;
        dec ebx
        jz aa
        jnz lp2
    }
    aa:
    return;
```

上面两个程序的运行时间可以调用系统的时钟函数来得到。只用 C++ 语句大约要 10s, 而用混合语句大约要 7s。可以看出, 用汇编语句要比 C++ 语句节省了 30% 的时间。在一些要求实时性的工业软件以及具有大量计算的辅助软件中, 用内嵌的汇编语句来代替某些 C/C++ 语句能大大提高程序的运行速度。

3 结束语

汇编语言与高级语言的混合编程同时具有了两种语言的优点, 在工业控制及科学计算之中具有很强的实用性。

参 考 文 献

- [1] 王士元. C 语言高级实用程序设计[M]. 北京: 清华大学出版社, 1997
- [2] 尹彦芝. C 语言高级实用教程[M]. 北京: 清华大学出版社, 1992
- [3] Charles Wright(著). Visual C++ 程序员使用大全[M]. 邓劲生, 张晓明(译). 北京: 中国水利水电出版社, 2001