

汇编语言与C语言的混合程序设计技术

王文东,李竹林,尚建人

(延安大学,陕西 延安 716000)

摘要:讨论汇编语言与C语言的混合程序设计技术。在大规模的程序设计中,汇编语言程序与高级语言程序相结合,充分发挥各自的优点,实现两种语言之间的混合编程,是解决实际问题的有效方法。在实际的程序设计中,C语言主程序可以调用汇编语言子程序,汇编语言主程序可以调用C语言子程序,也可以在C语言程序中嵌入汇编语言语句。在汇编语言与C语言的混合编程中,参数的传递、模块的连接以及调用关系的确定是值得注意的关键问题。

关键词:汇编语言;C语言;混合编程

中图分类号:TP313

文献标识码:A

文章编号:1673-629X(2006)08-0018-03

Design Technique of Mixed Programming Between Assembly Language and C Language

WANG Wen-dong, LI Zhu-lin, SHANG Jian-ren

(Yan'an University, Yan'an 716000, China)

Abstract: This paper discusses a kind of technique of the mixed programming design between assembly language and C language. In programming design, the mixed programming between assembly language and C language is a good method that can exert their strong point. In actual programming, it may transfer assembly language subprogram in the main programs of C language, and transfer C language subprogram in the main programs of assembly language, and it may insert assembly language sentence in C language program. In mixed programming of assembly language and C language, some questions that should be noticed include parameter transfer, module connection and transfer relation.

Key words: assembly language; C language; mixed programming

0 引言

汇编语言是一种面向机器的语言,其特点是运行速度快、占用存储空间小,可直接对硬件进行控制,在一些实时控制的场合,有着不可替代的作用。然而汇编语言也有它的不足,如编写及调试相对高级语言程序来说要困难、复杂一些,尤其在处理数据、数值的混合运算时更是如此^[1]。

作为一种高级语言,C语言毫无疑问是杰出的,它的功能丰富、表达能力强、使用灵活、开发效率高,因而比汇编语言使用更为广泛。通常在软件开发过程中,大部分程序采用高级语言编写,以提高程序的开发效率;但在要求执行速度快、占用空间少或要求直接控制硬件的场合,则利用汇编语言编写,以提高程序的运行效率。

1 汇编语言和C语言混合编程的方法

C语言和汇编语言混合编程的具体方法有以下几

种^[2]:

(1)进行混合编程时,一个任务往往被分为若干功能模块,每个模块以函数或子程序的形式存在,根据每一模块的特点选用适合的语言独立编程,然后使用相应的语言编译系统对其进行编译形成目标文件,最后将多个目标文件连接在一起形成一个完整的可执行文件。

(2)在C语言程序中使用汇编程序中定义的变量和常量。

(3)在C语言程序中直接内嵌汇编语句,可以实现C语言无法实现的硬件控制功能。

(4)在C语言程序中使用内部函数直接调用汇编语言语句。

1.1 C语言主程序调用汇编语言子程序

为了能正确地实现C语言程序对汇编语言程序的调用,汇编源程序必须严格按照编译系统要求约定的段顺序和规定的段组合。

1.1.1 汇编语言子程序的一般格式

汇编语言子程序的一般格式如下^[3]:

```
MYCODE SEGMENT PARA PUBLIC 'CODE'
PUBLIC _函数名;公共符号名说明
DGROUP GROUP MYDATA, MYBBS
```

收稿日期:2005-11-26

基金项目:延安大学专项科研基金资助项目(YDK2004-25)

作者简介:王文东(1972-),男,陕西横山人,讲师,主要从事软件工程与计算机应用技术研究。

```

MYDATA SEGMENT WORD PUBLIC 'CODE'
    ...定义变量,初始化数据
MYDATA ENDS
MYBBS SEGMENT WORD PUBLIC 'CODE'
    ...非初始化数据,开辟内存区
MYBBS ENDS
    ASSUME
CS:MYCODE,DS:DGROUP,SS:DGROUP
    ...
    _函数名 PROC NEAR/FAR
    PUSH BP
    MOV BP,SP
    PUSH SI
    PUSH DI
    ...程序主体语句
    POP DI
    POP SI
    POP BP
    RET
    _函数名 ENDP
    ...
MYCODE ENDS
END

```

1.1.2 被调用汇编语言子程序的格式约定

对被调用程序的格式与汇编语言程序的一般格式进行比较,可得出如下规律^[4]:

(1)由于程序中堆栈段是面向机器的,即 C 语言程序与汇编语言子程序共用了一个堆栈段,所以需要特别强调的是汇编程序的开头必须执行以下两条指令:

```

PUSH BP
MOVE BP,SP

```

(2)因为 C 语言程序允许子过程使用 SI 和 DI 存放局部变量,只有当变量多于两个时,才需要存储在堆栈中,所以在模式出现了: PUSH SI 和 PUSH DI 等语句。

(3)对于将被 C 语言程序调用的过程或子程序,必须用 PUBLIC 伪操作说明该过程或子程序是供外部调用的,如 PUBLIC 函数名。

(4)在整个程序中分别定义了代码段 MYCODE、初始化数据段 MYDATA 和非初始化数据段 MYBBS,并将 MYDATA 和 MYBBS 两段合成为段组 DGROUP,同在一个段组中的各段的基址是共享的。

(5)在“程序主体语句”部分一定要有关于函数值的返回语句,也就是说该返回值必须放置在 RET 语句之前。

(6)被调用过程需要通过 RET 命令返回。

(7)结束处的 END 语句只表示程序结束,没有标识启动位置。

1.1.3 C 语言主程序中调用的格式约定

为了让 C 语言主程序能成功调用汇编语言子过程, C 程序中要使用关键字“EXTERN”对调用函数做显式说明,将被调用的汇编语言过程说明为外部函数。当外部引用

函数无返回值或是 INT 型时,对它的说明可以缺省。具体格式如下:

EXTERN 返回值类型 过程名(参数说明)

1.2 汇编语言主程序调用 C 语言子程序

汇编语言的一些优势是 C 语言所无法比拟的,比如汇编程序在输入时可以采取各种进制数据以及直接读取数据等,为此,可以用汇编程序作为主程序调用 C 语言子程序^[4]。

汇编语言主程序的格式基本上与汇编程序子程序的一般格式一致,只需在程序开始处加入语句:

EXTERN 函数名

说明这个函数是外部的,是将被调用的 C 语言子程序。

省去主过程语句中关于 BP, SI, DI 的堆栈操作语言。在主过程语句中通过 CALL 语句实现对外部函数的调用。格式如下:

CALL NEAR PTR 函数名

对于 C 语言子程序作为子程序时,程序中应只包含 C 函数,无需再作 EXTERN 说明。

1.3 C 语言程序中嵌入汇编语言语句

在 C 语言中允许直接编写汇编语言代码, C 程序中嵌入汇编语句可以没有分号,具体语法如下^[5]:

asm <opcode> <operands>

若要嵌入一组汇编语句,则需要用括号‘{’和‘}’把它们括起来。例如:

```

asm {
    mov ax,data1
    xchg ax,data2
    mov data1,ax
}

```

C 语言允许嵌入四类汇编命令:一般指令、串操作指令、跳转指令和数据定义与分配指令^[4]。

2 汇编语言与 C 语言连接的关键问题

汇编语言和 C 语言混合编程,需要解决 3 个主要问题:函数调用过程中参数的传递、汇编程序与 C 程序模块的连接以及调用关系的确定^[4]。

2.1 参数传递

汇编语言程序和 C 语言程序在调用时会涉及到参数传递,一般采用堆栈进行传递。在汇编语言中将 BP 作为基址寄存器,调用程序先将来自于 C 语言程序中的参数依次压入堆栈中,然后当需要使用这些参数时,再用 BP 加上不同的偏移量依次对堆栈中的数据进行存取操作。

2.2 调用关系的确定

对于将要涉及到调用的过程或函数需要事先说明和建立调用与被调用关系。被调用的过程或函数应预先说明为外部类型,以便被外部模块引用,而调用程序也需要在程序中说明将要引用的外部模块的名称。

在说明调用关系时还应根据不同的存储模式确定相应的汇编语言格式。C 程序小模式对应汇编程序的近类

型过程,而 C 程序大模式则对应汇编程序的远类型过程。

2.3 汇编程序与 C 程序的模块连接

为了保证汇编语言和 C 语言模块文件的正确连接,必须做到两点:一是汇编模块必须采用和 C 模块一致的存储模式;二是汇编模块必须遵守和 C 兼容的命名约定,包括函数和变量命名约定^[1]。

C 语言共有 6 种不同的存储模式^[6]:微模式 Tiny、小模式 Small、中模式 Medium、紧凑模式 Compact、大模式 Large 和巨模式 Huge。其中微模式的程序、数据、堆栈都在同一段内;小模式有一个数据段和一个代码段。混合编程时必须保持汇编语言和 C 语言的存储模式一致。在汇编语言中使用伪指令 `Model * * *`,表示在汇编语言中采用的存储模式,其中 `* * *` 表示上述 6 种模式中的一种。如 C 采用了小模式,则在汇编语言中要使用 `Model Small` 伪指令,这时两者的存储模式是一致的。在小模式下所有的指针都是 Near 型近指针,程序运行效率高,一般情况下应尽量采用小模式进行混合编程。

编译系统在对 C 源程序编译时,会在其中的变量名、函数名前加下划线。为使连接后的程序保持一致性,被 C 语言调用的汇编子程序中,所有标识符都要加下划线。在 C 语言中,C 对所调用的汇编子程序中的函数、变量,都要用 EXTERN 伪命令予以说明,一般将其放在各函数体外。同时,为了使汇编语言的标识符能在 C 语言中可见,必须用 PUBLIC 操作符定义它。在汇编语言中使用 C 语言的函数和变量时,应在函数和变量名前加下划线,

并在汇编程序的开始部分,对调用的 C 函数和变量用 EXTERN 加以说明^[7]。

3 结束语

汇编语言和 C 语言的混合编程方法可以使汇编语言与 C 语言之间取长补短,充分发挥各自优势,相互交叉调用,进行参数传递,共享数据信息和数据结构,使由此开发的软件更实用、更安全可靠,使开发和编程工作达到事半功倍的效果。

参考文献:

- [1] 郭小梅. 汇编语言与 C 语言的混合编程及应用[J]. 南京农专学报, 2002, 18(4): 59-62.
- [2] 黄梅妹. C 语言与汇编语言的混合编程[J]. 厦门科技, 2000 (1): 19-20.
- [3] 张雪兰. 汇编语言程序设计[M]. 北京: 北京理工大学出版社, 2001.
- [4] 侯昌昌, 朱玉龙. 汇编语言与 C 语言在编程中的交叉应用[J]. 安徽工业大学学报, 2003, 20(2): 148-151.
- [5] 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.
- [6] 吕文强. C 语言的 DOS 系统程序设计[M]. 北京: 清华大学出版社, 1995.
- [7] 钟晓捷, 黄梅妹. 汇编语言程序设计[M]. 北京: 电子工业出版社, 2000.

(上接第 17 页)

的各种资源,可以被迅速找到并集合到一起,并实现它们之间通畅的沟通和协作。

基于这三大基本思想体系设计的协同软件一方面打造了高度“协同”的管理和办公环境,另一方面大大深化了 OA 的应用,从而完全突破了传统 OA 的局限,有效帮助企业整合各种资源,提升管理。

随着 OA 自身不断地完善和提升,随着管理思想和信息技术的发展,相信协同软件将越来越体现出它的独特优势,并在如下三方面进一步提升其应用:

1) 协同化。协同化不仅仅是企业管理理念发展的趋势,也是信息技术发展的趋势。协同 OA 一方面将加强自身“内含”的协同性,更多地突破资源之间的壁垒,提供一个畅通无阻的工作环境;另一方面,由于协同 OA 的定位并不在于取代其他专业管理软件,而在于提供融合孤立资源的平台,因此它也将加强其“外延”的协同性,以各种技术手段实现与其他专业软件更多的互动和对接。

2) 智能化。协同 OA 不仅要能充当企业办公的助手,更要能进一步充当“参谋”的角色。这就意味着协同 OA 不仅仅是信息的载体,更是信息的分析工具,能够实现对数据的加工和转换,提供从基本查询、报表和智能分析的

一系列工具,并以各种形象的方式展现,从而为企业考察运营情况、分析当前问题所在、制定未来发展计划等方面提供更多的决策支持。

3) 知识化。协同软件还应是企业知识管理的工具。尤其是在知识经济下,知识已经成为企业能够获得发展的核动力,而不再是办公事务本身带来的短期利益。协同软件将利用自身的优势,更多地关注知识的收集、积累与继承,并提供相应的知识管理工具,以帮助提升企业及员工的内在能力,以实现其可持续性的发展^[5]。

参考文献:

- [1] 宁力莎,任永昌,王英. 企业 OA 软件系统开发平台的规划[J]. 辽宁农业职业技术学院学报, 2005, 7(2): 1-2.
- [2] 吴取劲,刘杰,夏石莹. OA 系统工作流的工程模型与实现[J]. 河海大学常州分校学报, 2005, 19(2): 1-3.
- [3] 周建涛,史美林,叶新铭. 柔性工作流技术研究的现状与趋势[J]. 计算机集成制造系统, 2005, 11(11): 1-2.
- [4] 范玉顺. 工作流管理技术基础[M]. 北京: 清华大学出版社, 2001.
- [5] 李新. 一种协同办公系统的设计[J]. 计算机应用与软件, 2005, 22(8): 3-4.