

简析 MIPS 汇编语言与高级语言

郑秋辉 杨西珊

(首都师范大学信息工程学院,北京 100048)

摘 要

本文介绍了 MIPS 汇编语言中的算术类指令和数据传送类指令,并把这两类指令与高级语言进行了对比.通过 MIPS 汇编语言的两类指令与高级语言的对比,以此了解高级语言与汇编语言之间的联系;从而较为深刻理解计算机系统软、硬件的层次关系.

关键词: MIPS 汇编语言, C 语言, 算术类指令, 数据传送类指令.

中图分类号: TP313

0 引 言

汇编语言具有运行速度快、生成目标代码短的优点;通过学习汇编语言,可以对机器底层的操作有大概的了解.应用高级语言编写程序,开发周期相对汇编语言较短,并且具有源代码易于维护的优点;可以编写逻辑功能复杂的程序.但是目标代码的运行速度要比汇编语言慢得多.

高级语言的编译器将源程序编译成目标代码,我们往往忽视了一个问题:一条高级语言源代码功能的实现,往往要依靠一条或几条汇编语句来完成,这种转化是如何完成的呢?为了简单、清晰的说明这种转化的过程,这里高级语言以 C 语言为例,通过一些常用的语句来进一步了解编译器是如何完成编译工作的,生成的汇编代码这里表示为 MIPS 汇编指令.

1 算术类指令

1.1 算术类指令的用途

在 MIPS 汇编语言中,算术类指令作为最基本的指令,在几乎所有的汇编语言程序中,都有广泛的应用.算术类指令主要的功能是完成寄存器间的算术运算.

1.2 算术类指令的指令格式

add a,b,c;

sub a,b,c;

这两条指令的功能是将两个变量 b 和 c 相加、减,并把它相加、减的操作结果放在变量 a 中.这种符号的表示非常严格,每条 MIPS 指令仅仅执行功能单一的操作,并且操作数的个数和顺序要严格限定,以便在对汇编语言源程序编译时产生唯一的机器码.

若将 b、c、d 和 e 的值得总和放在变量 a 中,就要分几次累加来得到最后的结果.具体的指令序列如下:

add a,b,c;

add a,a,d;

add a,a,e;

以上说明了 MIPS 汇编语言算术类指令(以加法为例)的指令格式;下边给出了 C 语言中几条简单的赋值语句和与之对应编译后的 MIPS 汇编指令.

a = b + c;

d = a - e;

编译后的 MIPS 汇编指令如下:

add a,b,c

sub d,a,e

2 MIPS 汇编语言中寄存器的表示方法

与高级语言程序不同,汇编语言的算术运算指

收稿日期:2009-10-08

令的操作数不能是存储器变量,它们必须来自数量有限的寄存器.高级语言的变量和寄存器的主要区别是寄存器的数量是有限的.在 MIPS 体系结构中,寄存器为 32 位.

尽管在指令中只能使用 0 ~ 31 的寄存器号, MIPS 仍规定:用一个 \$ 符号后边跟两个字符代表一个寄存器.用 \$s0、\$s1 .. 来代表寄存器,以对应于 c 程序中的变量,用 \$t0、\$t1, .. 来代表将程序编译为 MIPS 指令时所需的临时变量.

```
add $t0, $s1, $s2
```

```
add $t1, $s3, $s4
```

```
sub $s0, $t0, $t1
```

3 数据传送类指令

3.1 数据传送类指令的用途

MIPS 汇编语言的每一条指令都有固定类型和数量的操作数.有些指令仅仅含有一个简单变量,而有些指令却含有像数组这样复杂的数据结构.这些复杂的数据结构中数据元素往往比寄存器的个数多得多;虽然处理器能将少量数据保存在寄存器中,但却不能保存数组中所有的数据.因此,像数组这样的数据结构,将保存在存储器中.

MIPS 指令中算术类指令的操作数只限于寄存器.因此,要完成数据在存储器和寄存器之间的传送,必须要有与之相对应的指令.这就是数据传送类指令主要的用途和存在的原因.

3.2 数据传送类指令的指令格式

数据传送指令中的常量叫做偏移量(offset),存放基址的寄存器叫做基址寄存器(base register).

指令格式:lw \$t0, 8(\$s3)

这条指令完成的功能是从存储器到寄存器的数据传送,这类指令通常叫做取数指令.操作名称 lw (load word) 后边紧跟的数据是要取的寄存器,第二个参数是用来访问存储器的常数和寄存器,指令的常数部分和第二个寄存器中的内容相加即得存储器的地址.

下边给出了一条简单的 C 语句,虽然这条 C 语句只有一个操作,但有一个操作数在存储器中, MIPS 汇编的算术类指令仅仅能完成寄存器间的算术运算,所以首先要将 a[8] 传送到寄存器,再完成算术运算.这个数组元素的物理地址由寄存器 \$s3 中的数组基址加上元素序号 8 计算得到并放在临时寄存器 t0 中,以便在下一条指令中使用.

$$g = h + a[8]$$

这条指令对应的 MIPS 汇编指令为:

```
lw $t0, 8($s3)
```

```
add $s1, $s2, $t0
```

与取数指令相反的指令叫做存储指令;它将数据从寄存器传送到存储器.存储指令的格式和取指令相似:指令格式依次为操作名称、存储的寄存器、数组元素的偏移量和基址寄存器. MIPS 指令中操作数的地址由常数和基址寄存器的内容共同决定.实际的 MIPS 存储指令的名字为 sw(store word).

$$A[12] = h + A[8]$$

上边给出的这条 c 语句中有两个操作数在存储器中,因此需要更多的 MIPS 指令.具体的 MIPS 汇编指令如下,取字指令(lw)选择 A[8] 时使用了偏移量,加法指令将结果放在临时寄存器 \$t0 中:

```
lw $t0, 32($s3)
```

```
add $t0, $s2, $t0
```

```
sw $t0, 48($s3)
```

上边各例都是使用常量来访问数组,在实际的编程中,更多的情况是使用变量来访问数组.下边给出了一条使用可变下标来访问数组的例子:

$$g = h + A[i]$$

假设 A 是有 100 个元素的数组,其基址放在寄存器 \$s3 中.并且,变量 g、h 和 i 分别放在寄存器 \$s1、\$s2 和 \$s4 中.在将 A[i] 取出并放到一个临时寄存器之前,必须得到它的地址.在将 i 加到数组 A 的基址以得到 A[i] 的地址之前,由于使用字节寻址,所以必须将下标乘以 4;当然,也可以用乘法指令实现上述操作,但这里暂不涉及,用加法实现 i 乘以 4 的运算($i + i = 2i$, $2i + 2i = 4i$).

这条 C 语句对应的 MIPS 汇编代码为:

```
add $t1, $s4, $s4
```

```
add $t1, $t1, $t1
```

```
add $t1, $t1, $s3
```

```
lw $t0, 0($t1)
```

```
add $s1, $s2, $t0
```

4 MIPS 汇编语言与高级语言的对比

很多人可能认为底层的机器语言像人类语言一样种类繁多,但实际上机器语言之间却十分相像,因此如果学会了一种语言,其他的也就很容易懂了.这种相似是由于所有的计算机都是建立在硬件技术之上,而这些硬件技术又是基于相似的原理的.另外,

还由于总有一些基本操作是所有计算机都提供的,而且计算机设计者的目标是相同的,即找到一种语言,使得人们用它易于设计硬件且易于编写编译器,同时可以使得机器的性能最好、价格最低。

通过对于 C 语言与 MIPS 汇编语言简单的算数类指令和数据传送类指令的比较,并且给出它的基本原理以及相应的指令结构,不难发现高级语言与汇编语言之间有很多相似之处。通常情况下,一条 C 语句功能的实现,要依靠一条或几条 MIPS 汇编指令来完成。

对于机器底层的操作,功能越单一,硬件电路越容易实现;而对于使用者而言,往往需要逻辑功能较为复杂的操作,编写功能复杂的程序。我们使用高级语言编写程序,可以脱离底层具体的硬件操作,在短时间内就可以编写出可读性强、易于维护、逻辑功能复杂的代码;而高级语言源程序到汇编级指令的转化,往往由编译器编译完成,这使得程序员可以较为轻松的完成程序的编写;但对于机器底层的操作和低级语言的了解绝不是没有必要的,只有当我们理解了汇编语言与高级语言各自的特点以及它们之间

的对应关系,才能更深入的研究计算机软、硬件的层次关系。

5 结 语

软、硬件和计算机各种语言之间有着千丝万缕的联系,回顾以前所学的一些看似独立的科目,寻找它们之间的区别和联系,经常反思、总结,会得到新的发现。学习高级语言的程序设计,仅仅能掌握编程的基本方法,但却不能理解计算机底层是如何工作的;学习汇编语言,对于计算机底层的操作可以加深理解;高级语言功能的实现,要依靠一条或几条汇编指令来执行。

通过平时的学习,应该学有所思,寻找学科之间的联系,学知识绝不能不求甚解;应该经常进行积累、反思、总结它们之间的相同与差异;可能现在会困于很多的不解,但在求知的路上会走得更远。本次仅仅介绍了 MIPS 汇编语言的两类指令以及简单的 C 语言语句,旨在抛砖引玉,激发大家发现问题、解决问题的兴趣。

参 考 文 献

- [1] David A. Patterson, John L. Hennessy. 郑纬民等译. 计算机组成和设计:硬件/软件接口(第二版). 北京:清华大学出版社,2003.
- [2] 蒋本珊. 计算机组成原理. 北京:清华大学出版社,2008.
- [3] Dominic Sweetman 著,李鹏,鲍峥,石洋译. MIPS 体系结构透视. 北京:机械工业出版社,2008.

A Brief Introduction of Assembly language and Advanced language

Zheng Qiuhui Yang Xishan

(College of Information, Capital Normal University, Beijing 100048)

Abstract

This article introduced the MIPS Assembly language in the arithmetic instruction and data transfer instruction, and to compare the two types of instruction and the Advanced language. Through contrast of the MIPS Assembly language and Advanced language in two types of instruction, to understand the relationship between Advanced language and Assembly language; and thus a more profound understanding of computer systems, hardware and software hierarchy.

Key words: MIPS Assembly language, C programming language, Arithmetic Instruction, Data transfer instructions.

作者简介 杨西珊,首都师范大学信息工程学院,讲师,研究方向:计算机科学与技术. 郑秋辉(1988-),男,首都师范大学信息工程学院,本科生,计算机科学与技术专业。