

Assembly Language and
Computer Architecture
Using C++ and Java

汇编语言与计算机体系结构

—— 使用C++和Java

Anthony J. Dos Reis 著

吕宏辉 马海军 等译



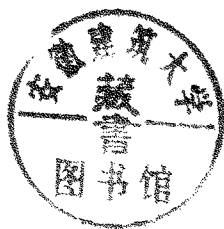
清华大学出版社

国外计算机科学经典教材

汇编语言与计算机体系结构 ——使用 C++ 和 Java

Anthony J. Dos Reis 著

吕宏辉 马海军 等 译



清华大学出版社
北 京

Anthony J. Dos Reis

Assembly Language and Computer Architecture Using C++ and Java

EISBN: 0-534-40527-4

Copyright © 2004 by Course Technology, a division of Thomson Learning.

Original language published by Thomson Learning (a division of Thomson Learning Asia Pte Ltd). All Rights reserved.

本书原版由汤姆森学习出版集团出版。版权所有,盗印必究。

Tsinghua University Press is authorized by Thomson Learning to publish and distribute exclusively this Simplified Chinese edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本中文简体字翻译版由汤姆森学习出版集团授权清华大学出版社独家出版发行。此版本仅限在中华人民共和国境内(不包括中国香港、澳门特别行政区及中国台湾)销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可,不得以任何方式复制或发行本书的任何部分。

北京市版权局著作权合同登记号 图字 01-2004-6537 号

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

汇编语言与计算机体系结构:使用C++和Java/(美)赖斯(Reis, A. J. D)著;吕宏辉,马海军等译. —北京:清华大学出版社,2006.1

ISBN 7-302-12003-X

I. 汇… II. ①赖…②吕…③马… III. ①汇编语言-高等学校-教材 ②计算机体系结构-高等学校-教材
IV. ①TP313 ②TP303

中国版本图书馆CIP数据核字(2005)第120273号

出版者:清华大学出版社 地址:北京清华大学学研大厦
http://www.tup.com.cn 邮编:100084
社总机:010-62770175 客户服务:010-62776969

责任编辑:冯志强

印装者:北京鑫海金澳胶印有限公司

发行者:新华书店总店北京发行所

开本:185×260 印张:40.75 字数:1013千字

版次:2006年1月第1版 2006年1月第1次印刷

书号:ISBN 7-302-12003-X/TP·7775

印数:1~3000

定价:79.00元

出版说明

近年来,我国高等学校的计算机学科教育进行了较大的改革,急需一批门类齐全、具有国际水平的计算机经典教材,以适应当前的教学需要。引进国外经典教材,可以了解并吸收国际先进的教学思想和教学方法,使我国的计算机学科教育能够与国际接轨,从而培育更多具有国际水准的计算机专业人才,增强我国信息产业的核心竞争力。Pearson、Thomson、McGraw-Hill、Springer、John Wiley 等出版集团都是全球最有影响的图书出版机构,它们在高等教育领域也都有着不凡的表现,为全世界的高等学校计算机教学提供了大量的优秀教材。为了满足我国高等学校计算机学科的教学需要,我社计划从这些知名的国外出版集团引进计算机学科经典教材。

为了保证引进版教材的质量,我们在全中国范围内组织并成立了“清华大学计算机外版教材编审委员会”(以下简称“编委会”),旨在对引进教材进行审定、对教材翻译质量进行评审。“编委会”成员皆为全国各类重点院校教学与科研第一线的知名教授,其中许多教授为各校相关院、系的院长或系主任。“编委会”一致认为,引进版教材要能够满足国内各高校计算机教学与国际接轨的需要,要有特色风格,有创新性、先进性、示范性和一定的前瞻性,是真正的经典教材。为了保证外版教材的翻译质量,我们聘请了高校计算机相关专业教学与科研第一线的教师及相关领域的专家担纲译者,其中许多译者为海外留学回国人员。为了尽可能地保留与发扬教材原著的精华,在经过翻译和编辑加工之后,由“编委会”成员对文稿进行审定,以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和能力所限,本套外版教材在出版过程中还可能存在一些不足和遗憾,欢迎广大师生批评指正。同时,也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材,共同为我国高等学校的计算机教育事业贡献力量。

清华大学出版社

国外计算机科学经典教材

编审委员会

主任委员:

孙家广 清华大学教授

副主任委员:

周立柱 清华大学教授

委员(按姓氏笔画排序):

王成山	天津大学教授
王 珊	中国人民大学教授
冯少荣	厦门大学教授
冯全源	西南交通大学教授
刘乐善	华中科技大学教授
刘腾红	中南财经政法大学教授
吉根林	南京师范大学教授
孙吉贵	吉林大学教授
阮秋琦	北京交通大学教授
何 晨	上海交通大学教授
吴百锋	复旦大学教授
李 彤	云南大学教授
杨宗源	华东师范大学教授
沈钧毅	西安交通大学教授
邵志清	华东理工大学教授
陈 纯	浙江大学教授
陈 钟	北京大学教授
陈道蓄	南京大学教授
周伯生	北京航空航天大学教授
孟祥旭	山东大学教授
姚淑珍	北京航空航天大学教授
徐佩霞	中国科学技术大学教授
徐晓飞	哈尔滨工业大学教授
秦小麟	南京航空航天大学教授
钱培德	苏州大学教授
曹元大	北京理工大学教授
龚声蓉	苏州大学教授
谢希仁	中国人民解放军理工大学教授

前 言

本书特点

汇编语言和计算机体系结构都是难学的课程。传统的汇编语言是如此复杂和特别,以至于即使学习基本概念都可能是一项困难任务。计算机体系结构也有同样的问题,而且更严重。学习汇编语言,我们要研究一个复杂系统;而学习计算机体系结构,则要研究许多复杂系统。

那么怎样才能最有效地学习汇编语言和计算机体系结构呢?本书采用的方法是使用一个简单的计算机模型 H1。H1 非常容易学习,学生可以迅速用 H1 完成大量任务。另外,学生可以着重学习重要的系统概念,而不是无关紧要的细节。

使用简单的计算机模型介绍基本概念很难说是一项创新技术。但是,本书以两种创新的方式使用 H1 模型。第一,H1 几乎出现在每一章。结果是连贯的计算机系统的开发,而不是不相关主题的集合。第二,H1 不是静态模型,而是有一些学生可以修复的缺陷(事实上是重大缺陷)。H1 的缺陷不是本书的缺陷,相反,它们是本书用来使学生参与设计和实现过程的资源。

本书的最重要特点之一是提供大量帮助更好理解的练习。例如,学生可以用 C++ 或 Java 编写连接器,或者为 H1 实现一种面向堆栈的体系结构并评估其性能。

因为本书重点在重要的系统概念上面,所以本书为计算机科学和工程领域的高级课程提供了极好的准备。使用本书的学生将具备在高级编程、编译器设计、编程语言设计、操作系统、高级体系结构和计算机工程方面从事重要工作的知识。

H1 完美地示范了计算机体系结构和组织的基本方面。但是,它不能示范现代计算机中存在的某些高级功能。因此,笔者在第 14~16 章增加了关于现代系统的材料。这几章涉及存储系统、RISC、CISC、流水线技术、SPARC、Pentium 和 JVM(Java 虚拟机,Java Virtual Machine)。

书中各章可以以不同顺序讲述,某些章节可以跳过(参见后面的每章前提)。例如,第 5 章(数字逻辑电路)可以在第 1 章之后的任何时间讲述,或者跳过;第 16 章(Java 虚拟机)可以在第 7 章之后的任何时间讲述,或者跳过。

读者对象

本书包括汇编语言与计算机体系结构,可作为一个学期的大学课程。然而,本书同样很适合仅讲述汇编语言或计算机体系结构的课程。因为对硬件和软件的均衡对待,本书适合计算机科学和工程这两个专业。本书还非常适合作为计算机专业研究生教学计划的第一门课程,尤其适合那些没有深厚计算机科学或工程背景的学生。

本书前提

尽管使用了 C++ 和 Java,但本书的惟一预备知识是 C++ 或 Java 知识。开始的 C++ 示例非

常简单,慢慢才变得越来越复杂,而且都有详细的解释。这种对C++的细致处理,加上C++与Java的相似性,使本书完全适合只熟悉Java的学生。实际上,这样的学生将从本书获得额外好处:几乎不费力的C++入门。如果在某个课程密集的专业内没有C++的空间,则本书将使得在一门课程内包括C++、汇编语言和计算机体系结构成为可能。

尽管对C++的处理是柔和的,但本书仍将详细提供许多对C++的分析,并概述一些编程语言。例如,本书精确展示了对对象的实现方法。即使对C++高度熟练的学生也能发现这部分内容的知识性和吸引力。

对于只熟悉C++的学生而言,关于Java的内容不应该有任何问题。第1章给出几个说明浮点误差的Java程序。第4章讨论了可能在Java程序内出现的大小端问题。第8章讨论了Java中的指针和数组。第9章讨论了Java中的布尔运算符。第11章为学生演示了如何用Java(或C++)实现编译器和连接器。第13章讨论了与JVM有关的堆栈体系结构的某些分支。第16章全部致力于JVM的结构和操作。本书还包括大量同时适合Java和C++的一般材料。已经熟悉Java的学生将获得对该语言更深、更综合的理解。

补充内容

软件包、演讲幻灯片以及其他补充内容可以从Course Technology站点得到,网址是www.brookscole.com。包括问题答案、额外问题、微代码、编译器和连接器的C++和Java源代码,以及教学建议和教师手册。

软件包运行在Microsoft Windows、Microsoft DOS、SPARC Solaris、X86 Linux和Macintosh OS X上面。对其他平台的支持正在考虑之中。该软件包包括模拟器/源级调试器/仿形器(sim)、编译器(mas)、连接器(lin)、库程序(lib)、3个用来分析对象以及执行过程和库模块的程序(pic、mex和see)、水平微代码编译器(has)、垂直微代码编译器(vas)、伪代码加密器(enc)、几个实用程序、完整文档,以及所有书中的示例程序。sim、mas、has和vas都可以重新配置,以适应新的指令集。

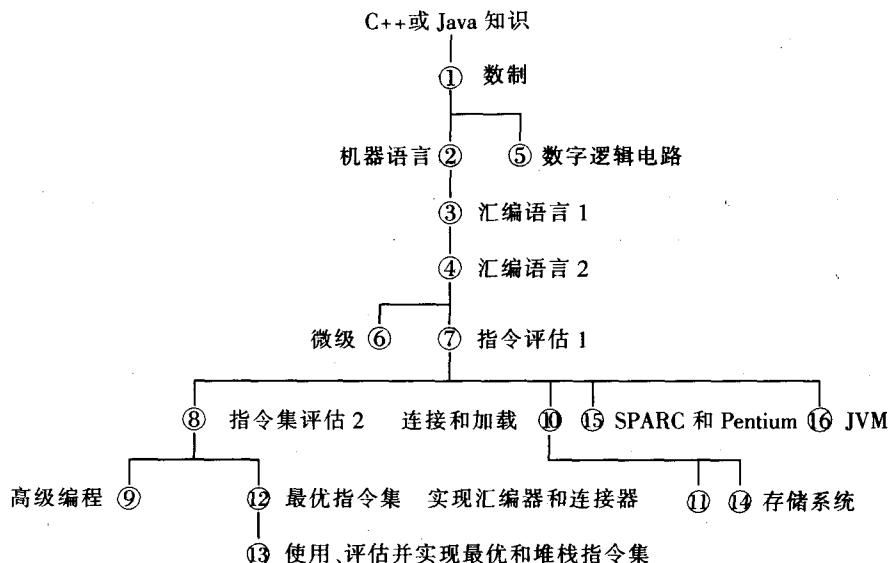
sim支持大量用于跟踪或调试机器语言程序或微代码的命令。特别有用的一个功能是在逐条机器指令的基础上显示微级(microlevel)活动。利用该工具,学生可以清楚地看到与任何一条机器指令相对应的微级活动。

sim可以使用未加密或加密的微代码。该特点使学生可以使用,但不能分析软件包内的加密微代码。在不能参考作者版本的“不利”情况下,学生可以编写自己的版本。

所有软件都使用了简单的命令行界面。命令行界面完全可以满足软件包内的程序类型,而且最易于学习使用。另外,命令行界面不因平台不同而变化。所有程序都可以不带任何命令行参数地被调用(在这种情况下,程序提示输入参数)。因此,我们可以在图形用户界面中方便地通过图标调用它们。

关于使用软件包的指导在书中,这将使学习使用该软件更容易。学生再也不必费力浏览大量文档,而教师也不必再将课堂时间浪费在教学生使用软件上面。完整文档以文件形式包含在软件包中,以供需要的人使用。

每章的预备知识



每章摘要

第1章主要介绍数制。但是,本章没有包括关于进位和溢出标志位以及书中后面需要的关于有符号和无符号比较的某些难点。本章还使用了几个Java程序来说明可能随浮点数发生的各种计算误差。

第2章介绍了H1、机器语言和sim。

第3章介绍汇编语言、汇编器、汇编过程和顺序指令执行的机制。

第4章完成了对H1汇编语言基础的介绍。本章还讨论了可能在Java程序中出现的大小端问题。因为H1非常简单,所以第2~4章需要很少的课堂时间。

第5章介绍数字逻辑电路的基础,这是完整理解H1的内部组织所必需的。另外,本章还介绍了一些重要的电子学原理。第6章依赖本章,但只是最低限度的依赖。因此,如果教师能够在课堂上提供一些逻辑电路的说明,那么可以跳过本章。

第6章给出H1和V1(V1与H1类似,但却是垂直微编程的)的内部组织以及微编程的基本原理。本章以水平和垂直微代码实现了基本指令集,还讨论了硬接线控制。在只讲述汇编语言的课程中,可以跳过本章。

第7、8章是暴露标准指令缺点的重要两章。通过分析编译器生成的代码,学生将形成良好指令集应该做什么的认识。这两章还给出对C++的有价值的剖析。

第9章在汇编器和C++的级别上,讲述了编程方面的若干高级主题。这些主题包括指向指针的指针、引用调用的潜在错误(是否以显式实现)、值结果调用和可变长变元列表。大多数教师都愿意至少讲述本章的一部分。但是,本章是可以跳过的。

第10章介绍加载和连接机制。本章使用pic、mex和see程序来简化对象、执行过程和库模块的分析,还在汇编器级别上分析了启动代码。

第11章介绍关于实现汇编器和连接器的指导。这两个编程项目此时提供给学生是恰

当的。提供的指导适用于C++和Java实现。学生通常喜欢这些项目,并能从中学到很多知识。这些项目提供了宝贵的用Java进行系统编程的机会。大多数教师都将发现,本章需要非常少的课堂讨论。

第12章介绍了最优指令集,这是对标准指令集的改进,它修正了主要在第7、8章暴露的H1的缺点。

第13章测定了最优指令集在支持复杂编程实体和机制方面的表现,比如对象、虚函数和名调用。本章还以乘法为例,讨论了可以配置计算功能的不同级别——电路级、微级和机器级。本章介绍了H1的堆栈指令集,并将其与最优指令集进行比较;还提供了关于实现最优和堆栈指令集的指导(两个值得学生此时做的项目)。

第14章使用H1作为工具,解释了程序重定位的分支,并进一步研究了各种存储系统。这些主题包括分页、命令分页、相联存储器、中断、页布置策略、存储器保护、分段和高速缓冲存储器。

第15章使用SPARC和Pentium论述某些不适合用H1论述的重要主题(如RISC、流水线技术、代码优化)。

第16章以前面介绍C++和H1汇编语言时使用的相同方式介绍了Java和JVM(尽管是以更浓缩的形式)。

致谢

笔者在微级和机器级上对H1和V1的设计,是受到Miles Murdocca和Vincent Heuring所著《计算机体系结构原理》(Prentice-Hall,2000)、Vincent Heuring和Harry Jordan所著《计算机系统设计和体系结构》(Addison Wesley Longman,1997),特别是Andrew Tanenbaum所著《结构化计算机组织,(第3版)》(Prentice-Hall,1990)等书中类似模型的影响。

编辑过程中的审阅者可能会显著地帮助写成一本更好的书,本书的审阅者就属于这种情况。在手稿的大量修订过程中,他们的介入为笔者提供了必不可少的指导。感谢New Orleans大学的Mahdi Abdelguerfi、New Orleans大学的Ming Hsing Chiu、Western Washington大学的Annie Groeninger、Iona学院的Lubomir Ivanov,以及Georgia Perimeter学院的Gene Sheppard。

笔者还想感谢学生们对手稿早期版本的有益反馈,感谢Vassar学院的John Collier和Brian Chickery在Macintosh方面的帮助,感谢与笔者同系的Andy Pletch在Unix方面的专业知识,感谢Addie Haas的建议和鼓励,感谢Matrix印刷厂的Merrill Peterson协调本书的印刷,并感谢Brooks/Cole出版社的Kallie Swanson在该项目还不成熟时就认识到它的价值。

Anthony J. Dos Reis

目 录

第 1 章 数制 1

1.1 简介 1

1.2 按位记数制 1

1.3 各数制的算术规则 4

1.4 数制转换 6

1.5 Horner 法 9

1.6 有符号二进制数 10

1.6.1 符号—绝对值
表示法 10

1.6.2 2 的补码表示法 11

1.6.3 1 的补码表示法 13

1.6.4 加 n 表示法 14

1.7 用 2 的补码加法计算
减法 14

1.8 2 的补码和无符号数
的范围 15

1.9 扩展 2 的补码和
无符号数 16

1.10 溢出 17

1.10.1 有符号数溢出 17

1.10.2 无符号数溢出 18

1.11 分析 2 的补码数 20

1.12 加法器电路 21

1.13 门 23

1.14 用加法器执行减法 24

1.15 比较有符号数 26

1.16 比较无符号数 27

1.17 正数、负数和无符号数的
一致对待 27

1.18 浮点数表示法 28

1.18.1 科学记数法 28

1.18.2 32 位 IEEE 754 浮

点数格式 29

1.18.3 下溢、溢出和特
殊值 30

1.18.4 某些 Java 程序中
的计算误差 30

1.18.5 长格式 35

1.19 小结 35

1.20 问题 36

第 2 章 机器语言 40

2.1 简介 40

2.2 计算机的组件 40

2.3 机器语言编程 44

2.4 在 sim 上运行机器语言
程序 48

2.5 问题 52

第 3 章 H1 汇编语言：第 1 部分 54

3.1 简介 54

3.2 汇编语言基础 55

3.3 注释汇编语言程序 58

3.4 使用标号 58

3.5 使用汇编器 61

3.6 低级语言与高级语言 65

3.7 汇编器工作原理 66

3.8 使用 dup 修饰符 67

3.9 操作数字段中的算术
表达式 68

3.10 指定当前位置 70

3.11 字符串 71

3.12 org 伪指令 74

3.13 end 伪指令 75

3.14 指令的顺序执行	76	5.3.1 门	133
3.15 问题	78	5.3.2 布尔函数	134
第4章 H1 汇编语言：第2部分	81	5.3.3 最小电路	135
4.1 简介	81	5.3.4 使用布尔代数使电路最小化	136
4.2 描述指令的简写形式	81	5.3.5 摩根定律的图示法	137
4.3 直接指令	82	5.3.6 使用卡诺图使电路最小化	138
4.4 堆栈指令	83	5.3.7 多路复用器	141
4.5 立即指令	85	5.3.8 解码器和编码器	142
4.6 I/O 指令	90	5.3.9 半加器和全加器	143
4.7 转移指令	96	5.3.10 一位移位器	144
4.8 间接指令	98	5.3.11 乘法器阵列	145
4.9 相对指令	102	5.3.12 算术逻辑单元	148
4.10 变址	103	5.3.13 三态缓冲器	149
4.11 连接指令	105	5.3.14 ROM	151
4.12 终止指令	109	5.4 时序逻辑电路	153
4.13 高级语言中指令的自动生成	109	5.4.1 SR 触发器	153
4.14 调试	110	5.4.2 时钟控制 D 触发器	155
4.15 使用存储映像式 I/O	115	5.4.3 时钟控制 JK 触发器	157
4.16 equ 伪指令	117	5.4.4 边沿检测	157
4.17 Java 程序的潜在问题：大小端	118	5.4.5 T 触发器	159
4.18 问题	121	5.4.6 RAM	159
第5章 电子学和数字逻辑电路基础	125	5.4.7 二进制计数器	160
5.1 简介	125	5.4.8 寄存器	161
5.2 电子学基础	125	5.4.9 时钟和序列发生器	163
5.2.1 导体和绝缘体	125	5.5 集成电路	164
5.2.2 一个简单电路	126	5.6 问题	165
5.2.3 有效电压	127	第6章 H1 和 V1 的微级	168
5.2.4 电源	127	6.1 简介	168
5.2.5 如何避免触电	128	6.2 H1 的微级	168
5.2.6 晶体管	129	6.2.1 H1 的组织概述	168
5.2.7 电容、电感和阻抗	131		
5.3 组合逻辑电路	133		

6.2.2	微指令	170	6.6.3	汇编垂直微 代码	205
6.2.3	存储器数据 总线	171	6.7	问题	210
6.2.4	存储器地址 总线	172	第 7 章	评估 H1 的指令集体系 结构: 第 1 部分	214
6.2.5	寄存器库	172	7.1	简介	214
6.2.6	指定微操作	175	7.2	哑编译器	214
6.2.7	驱动寄存器库的 控制输入端	175	7.3	全局变量	218
6.2.8	整合 H1 各部分	176	7.4	局部变量	221
6.2.9	分支多路复 用器	178	7.5	变化的相对地址	229
6.2.10	定时问题	180	7.6	参数和传值调用	230
6.2.11	微指令格式	181	7.7	返回值	235
6.2.12	序列发生器驱动的 控制输入端	181	7.8	需要相对指令的 原因	236
6.2.13	基本指令集的微 代码	182	7.9	确定动态局部变量和 参数的地址	238
6.2.14	硬件/微代码 权衡	187	7.10	间接引用指针	241
6.3	在 sim 上使用新的微 代码	189	7.11	问题	243
6.3.1	如何使用! 伪 指令	189	第 8 章	评估 H1 的指令集体系 结构: 第 2 部分	249
6.3.2	创建配置文件	189	8.1	简介	249
6.3.3	创建二进制微 代码	191	8.2	引用调用	249
6.3.4	用新指令集 汇编	192	8.2.1	引用调用的 实现	249
6.3.5	在微级上跟踪	193	8.2.2	C++引用参数的 概念视图	256
6.3.6	加密微代码	198	8.2.3	不间接引用C++引 用参数的情况	257
6.3.7	编写并优化复杂 的微代码	198	8.2.4	传值调用与引用 调用的比较	258
6.4	解释机器语言指令	200	8.2.5	常数和表达式作为 变元	259
6.5	硬接线控制	201	8.3	函数重载和重命名	263
6.6	垂直微编程	202	8.4	结构	267
6.6.1	使用 V1	202	8.5	Java 中的指针	271
6.6.2	构成更短的微 指令	203	8.6	函数指针	272
			8.7	数组	274

8.7.1 定义并访问 数组	274	第 11 章 用C++或Java实现 汇编器和连接器	376
8.7.2 数组作为 变元	278	11.1 简介	376
8.7.3 Java 中的 数组	282	11.2 编写汇编器	376
8.8 控制语句	283	11.2.1 一个简单汇编器 的规范	377
8.9 有符号数和无符号数 比较	285	11.2.2 操作码表	378
8.10 多字加法	285	11.2.3 符号表	380
8.11 位级操作	286	11.2.4 使用二进制 文件	382
8.12 递归	287	11.2.5 读取输入文本文件 并创建标题	385
8.13 问题	293	11.2.6 汇编机器字	387
第 9 章 高级汇编语言编程	301	11.2.7 将机器正文写入 输出文件	388
9.1 简介	301	11.2.8 分析输入	388
9.2 指向指针的指针	301	11.2.9 Java 程序的大小端 问题	388
9.3 引用调用及其他地方存在 的潜在错误	307	11.2.10 添加对 public、 extern 和 end 伪 指令的支持	389
9.4 关系和布尔表达式	310	11.3 编写连接器	390
9.5 字符串	313	11.3.1 一个简单连接器 的规范	390
9.6 值结果调用	315	11.3.2 构建 P、E、R 和 S 表	391
9.7 变长变元列表	321	11.3.3 重定位地址并解析 外部引用	393
9.8 问题	326	11.3.4 添加库支持	395
第 10 章 连接和加载	329	11.4 问题	396
10.1 简介	329	第 12 章 最优指令集	397
10.2 浮动字段	330	12.1 简介	397
10.3 显示标题和机器代码 正文的程序	334	12.2 新增和改进的指令	398
10.4 小 s 项	336	12.2.1 mult、m、div 和 rem	399
10.5 禁止重定位	337	12.2.2 addc 和 subc	400
10.6 分别汇编	338	12.2.3 scmp	401
10.7 乘法子例程	348		
10.8 使用库	350		
10.9 分别汇编的优点	357		
10.10 启动代码	357		
10.11 C++模块的分别编译	367		
10.12 问题	371		

12.2.4	ucmp	402
12.2.5	shll 和 shrl	403
12.2.6	shra	404
12.2.7	addy	405
12.2.8	or、xor、and 和 flip	407
12.2.9	cali	409
12.2.10	sect 和 dect	409
12.2.11	sodd	410
12.2.12	esba、reba、bpbp、 pobp 和 pbp	411
12.2.13	cora	416
12.2.14	bcpy	417
12.3	链表——使用最优指令集 示例	418
12.4	关于最优指令集的进展 报告	424
12.4.1	主存储器的数量 不足	424
12.4.2	存储器存储字符串 的效率不高	424
12.4.3	缺少加和减的立即 指令	425
12.4.4	缺少变址寄 存器	425
12.4.5	累加器型寄存器 太少	425
12.4.6	swap 指令破坏 sp 寄存器	425
12.4.7	缺少乘法和除法 指令	426
12.4.8	sp 寄存器作为栈顶 指针和基址寄存器 的双重用途	426
12.4.9	难以获得堆栈上 变量的地址	426
12.4.10	缺少块复制 指令	426
12.4.11	难以调用地址已	

	知的函数	426
12.4.12	aloc 和 dloc 指令 的局限性	426
12.4.13	难以执行有符号 数和无符号数 比较	426
12.4.14	难以执行多字 加法	426
12.4.15	不支持位级 操作	427
12.5	问题	427

第 13 章 使用、评估并实现最优指令集和堆栈指令集 430

13.1	简介	430
13.2	H1 上的乘法	430
13.3	二维数组	437
13.4	汇编语言中的面向对象 编程	441
13.4.1	C++ 的结构、类和 对象	441
13.4.2	继承	452
13.4.3	通过指针调用成员 函数	457
13.4.4	虚函数和 多态性	464
13.5	名调用	472
13.6	面向堆栈的体系结构	479
13.6.1	体系结构与 组织	479
13.6.2	以栈顶代替 ac 寄存器	480
13.6.3	4 位操作码的 可用性	484
13.6.4	特殊的堆栈 操作	485
13.6.5	更简单的编译	486
13.6.6	返回值	487
13.6.7	使用堆栈指令集	

示例	489	第 15 章 某些现代体系结构	530
13.6.8 比较最优指令集和 堆栈指令集	491	15.1 简介	530
13.6.9 关于 JVM 的几点 观察	492	15.2 CISC 与 RISC	530
13.7 编写最优和堆栈指令集的 微代码	493	15.3 SPARC: RISC 体系 结构	532
13.8 微编程的挑战	496	15.3.1 重叠的寄存器 窗口	532
13.9 问题	497	15.3.2 保存和恢复 指令	536
第 14 章 存储系统	503	15.3.3 加载和存储 指令	538
14.1 简介	503	15.3.4 转移、call 和 jmp 指令	543
14.2 地址的传送	503	15.3.5 指令流水线	546
14.3 多程序系统	505	15.3.6 连接指令	552
14.4 直接存储器存取	505	15.3.7 寻址方式	553
14.5 分时系统	506	15.3.8 一个简单的汇编 语言程序	555
14.6 重定位寄存器	506	15.3.9 合成指令	557
14.7 虚拟存储器	508	15.3.10 编译器生成的 代码	559
14.7.1 简单分页 系统	509	15.3.11 存储映像式 I/O	565
14.7.2 关联存储器	512	15.4 Pentium: CISC 体系 结构	566
14.7.3 中断	514	15.4.1 寄存器结构	566
14.7.4 请求式分页	516	15.4.2 汇编语言	567
14.7.5 请求式分页的页面 替换策略	518	15.4.3 编译器生成的 代码	568
14.7.6 页面大小的 考虑	519	15.4.4 I/O 指令	570
14.7.7 超级用户/用户 模式	520	15.5 问题	571
14.7.8 存储器保护	521	第 16 章 Java 虚拟机 (JVM)	576
14.7.9 分页的分段	521	16.1 简介	576
14.8 高速缓冲存储器	522	16.2 JVM 的结构	577
14.8.1 缓存的替换 策略	523	16.3 某些使用操作数堆栈的 普通操作	578
14.8.2 写策略	523	16.4 方法的调用和返回	579
14.8.3 存储器到缓存的 映射方案	524		
14.9 可共享的程序	526		
14.10 问题	527		

16.5 执行引擎的实现	581	附录 B 微指令级汇总 (mic.txt)	606
16.6 明智之举: JVM 使用堆栈 体系结构	582	附录 C 标准指令集的微代码 (s.has)	609
16.7 一个简单的例子	582	附录 D SPARC 汇总 (sparc.txt)	614
16.8 常数池	584	附录 E java 虚拟机字节码汇总 (jvm.txt)	620
16.9 类文件格式	585	附录 F ASCII 汇总 (ascii.txt)	625
16.10 字节码的空间效率	589	附录 G 数制汇总 (number.txt)	627
16.11 控制语句	590	附录 H 了解C++对 Java 程序员的 帮助	628
16.12 反汇编类文件	591		
16.13 对象和数组	593		
16.14 JVM: 抽象机器	595		
16.15 问题	595		
附录 A 指令集汇总 (s.txt、b.txt、o.txt、 k.txt)	598		

第 1 章 数 制

1.1 简介

在一个有名的笑话里,一名速读课程的毕业生骄傲地宣称他能够在 10 分钟内读完托尔斯泰的《战争与和平》。当有人问他这是一本什么样的书时,这名毕业生答道:“我认为这本书与俄罗斯有关。”完成计算机课程——特别是汇编语言和计算机体系结构课程的学生,对于所学课程的认识通常都有类似不充分的回答。毕竟,计算机是横跨许多技术的复杂系统。即使要彻底理解一个计算机系统,都可能需要数年的刻苦研究。然而,有一种方法可以快速获得对计算机基本概念的理解:研究一种非常适合学习这些概念的计算机。我们在本书中就要这样做。我们将研究一种足够简单,可以快速掌握的计算机,但这种计算机又足够符合实际,可以精确模拟现代计算机的本质特征。

在本章,我们首先研究数制。人类使用十进制。但是计算机使用一种不同的数制——二进制。十进制数使用 10 个符号(0、1、2、…、9),这些符号被称为数字。二进制数只使用两个符号(0 和 1)——称做位。第三种数制是十六进制(简写为 hex),它可以方便地作为计算机内二进制数的简写形式。因此,十六进制和二进制在计算机领域内都很重要。十六进制数使用 16 个符号(0、1、2、…、9、A、B、C、D、E、F),我们称之为十六进制数字,有时也简称数字。

1.2 按位记数制

十进制、二进制和十六进制都是按位记数制。在按位记数制中,每个位置对应不同的权。以十进制数 25.4 为例,从右向左 3 个位置的权是 $\frac{1}{10}$ 、1 和 10:

$$\begin{array}{ccccccc} 2 & 5 & . & 4 & & & \\ \hline 10 & 1 & & \frac{1}{10} & & & \text{权} \end{array}$$

每个位置的权决定该位置的数字对数值的贡献大小。例如,因为 25.4 中的“2”占据 10 的位置(即该位置的权是 10),所以它对这个数的值贡献了 $2 \times 10 = 20$ 。十进制数的值是每个数字乘以权之后的和。因此,25.4 的值是:

$$2 \times 10 + 5 \times 1 + 4 \times \frac{1}{10}$$

在十进制数中,相邻位置的权从右向左以 10 为系数增加。小数点左边第一个位置的权总是 1。因为十进制数有 10 个符号(0~9),并且权以 10 为系数增加,所以我们将十进制称做基数为 10 的按位记数制。