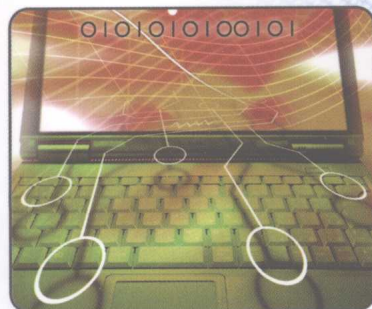


“十一五”高等院校规划教材

计算机系统结构



胡越明 编著



北京航空航天大学出版社

“十一五”高等院校规划教材

计算机系统结构

胡越明 编著

北京航空航天大学出版社

内 容 简 介

本书主要介绍计算机系统结构方面的内容,包括计算机系统结构的基本概念、CPU 设计原理和分析方法,以及多核和其他并行计算机系统的构成原理与软件平台技术。本书内容以介绍计算机系统基础知识为主,同时也介绍计算机系统产品中采用的新技术。本书每章配有大量的例题和习题,全书内容适合的学时数为 54~72 学时。

本书可作为计算机专业高年级本科生的教材,也可作为相关专业研究生的教材以及计算机工程技术人员参考书。

图书在版编目(CIP)数据

计算机系统结构/胡越明编著. —北京:北京航空航天大学出版社, 2007. 10

ISBN 978-7-81124-238-6

I. 计… II. 胡… III. 计算机体系结构 IV. TP303

中国版本图书馆 CIP 数据核字(2007)第 143026 号

© 2007, 北京航空航天大学出版社, 版权所有。

未经本书出版者书面许可,任何单位和个人不得以任何形式或手段复制或传播本书内容。

侵权必究。

计算机系统结构

胡越明 编著

责任编辑 杨 波 史海文

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail: bhp@263.net

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787 mm×960 mm 1/16 印张:22 字数:493 千字

2007 年 10 月第 1 版 2007 年 10 月第 1 次印刷 印数:5 000 册

ISBN 978-7-81124-238-6 定价:32.00 元

前言

近年来,计算机科学的发展异常迅速,新技术不断涌现,计算机学科的教材必须不断更新以赶上这一学科的飞速发展。为满足在计算机系统结构教学方面不断更新教学内容的需求,作者编写了本教材。本教材是作者近几年在上海交通大学的本科教学实践和教材编写经验的基础上,吸收了国外先进教材的内容和技术发展的最新成果后编写而成的,同时也吸收了近年来在微处理器产品中采用的最新技术成果。本书的知识点建立在计算机组成原理的基础上,读者在学本教材前应先学完“计算机组成”方面的课程。

本教材在国内原有教材的基础上,吸收了国际上先进教材中的有关内容,反映了计算机技术的最新技术和发展趋势,同时注意与相关课程的内容保持衔接。教材以基本原理的阐述为主,并结合较新的计算机系统实例,介绍了 Intel 处理器产品线等。第 1 章介绍计算机的基本组成和系统结构的概念,计算机系统的分类以及性能评价的基本方法。第 2 章介绍在目前先进的微处理器中采用的指令级并行技术,包括动态指令调度、分支预测、多重启动、推测执行和各种编译技术。第 3 章介绍 cache 方面的内容,包括在新型微处理器中采用的各种快速缓存的技术。第 4 章从系统结构的角度介绍输入/输出系统,包括 RAID 等海量存储系统和最新的输入/输出接口与总线。第 5 章介绍近年进入发展高潮的并行计算机的结构,包括互联网络、多核等多处理器系统,以及集群计算机系统等。随着多核处理器芯片的出现,桌上型计算机系统将普遍成为多处理器系统,这种并行系统实现中的一些概念和多核并行政程序设计的概念成为将来大部分软/硬件工程技术人员必须了解和掌握的知识。所以本教材的第 6 章介绍并行计算机系统的通信和同步机制、cache 一致性机制、并行软件的概念以及并行软件开发环境。本教材编制和收集了大量的例题和习题,习题除注重基础知识的掌握外,还致力于加强对学生的综合思考和创新能力的训练,具有实用性。学生通过这些例题和习题能够较好地掌握本课程的主要内容。本书全部内容适合的

前 言

学时数为 54~72 学时。

多核计算机时代的到来使本来面向高端应用的并行计算机系统得到普及,为此,本书在介绍多核并行计算机和并行计算方面用了较多篇幅。这是对计算机系统结构课程的一种探索。本书的编写得到 Intel 公司在技术上的帮助,在此向 Intel 公司表示感谢。本书的编写过程中,清华大学计算机系的薛魏老师、北京大学软件学院的张齐勋老师、复旦大学计算机系的冯红伟老师、浙江大学的陈天洲老师以及上海交通大学计算机系的邓倩妮副教授为本书提出了宝贵的意见,在此表示感谢。由于作者的水平有限,书中可能存在不妥之处,欢迎广大读者指正,以便于作者在今后版本的编写中不断完善。

胡越明

2007 年 8 月

目 录

第 1 章 计算机系统结构概论

1.1 概 述	1
1.1.1 计算机系统组成与系统结构的概念	1
1.1.2 语言与系统结构设计	4
1.1.3 软件与系统结构设计	6
1.1.4 应用需求与系统结构设计	8
1.1.5 器件与系统结构设计	11
1.2 计算机系统的分类	13
1.2.1 应用分类	13
1.2.2 结构分类	15
1.2.3 并行性分类	18
1.3 计算机指令集系统结构	20
1.3.1 指令系统设计	20
1.3.2 指令的数据访问方式	25
1.3.3 指令设计风格	28
1.4 计算机的性能评价	31
1.4.1 性能的衡量	31
1.4.2 性能的简单分析	33
1.4.3 性能的模拟	35
1.4.4 性能的测试	36
1.4.5 性能评价结果的统计和比较	41
1.4.6 阿姆达尔定律	44
习 题	45

目 录

第 2 章 计算机微观系统结构

2.1 指令级并行性	51
2.1.1 基本的指令流水线	52
2.1.2 指令的相关性	53
2.2 动态指令调度	63
2.2.1 基本思想	64
2.2.2 记分牌方法	68
2.2.3 Tomasulo 调度法	70
2.3 分支预测	75
2.3.1 分支预测	75
2.3.2 分支目标缓存	80
2.4 多重启动	81
2.5 推测执行	87
2.6 静态指令调度	94
2.6.1 静态指令调度	94
2.6.2 静态多重指令启动	100
2.6.3 软件流水	105
2.7 静态全局指令调度	108
2.7.1 路径调度	108
2.7.2 全局指令调度	110
2.7.3 条件指令	111
2.8 微观系统结构实例	114
2.8.1 P6 微系统结构	114
2.8.2 奔腾 4 的 NetBurst 微系统结构	116
2.8.3 SSE 技术	118
2.8.4 Core 微系统结构	119
2.8.5 IA-64 系统结构	121
习 题	123

第 3 章 计算机存储系统

3.1 主存储器	134
3.2 cache	139
3.2.1 cache 的地址映像	140
3.2.2 cache 的替换策略	145
3.2.3 cache 的更新策略	146

3.2.4	cache 的性能评价	149
3.3	辅助 cache	155
3.3.1	写缓存	155
3.3.2	踪迹 cache	157
3.3.3	替换 cache	159
3.3.4	伪相联 cache	161
3.3.5	cache 预取	162
3.4	多级 cache	164
3.5	虚拟存储器与 cache	169
3.5.1	虚拟存储器	169
3.5.2	实地址 cache	172
3.5.3	虚地址 cache	173
	习 题	174
第 4 章 输入/输出系统		
4.1	输入/输出系统结构	181
4.1.1	输入/输出总线	181
4.1.2	交换式输入/输出结构	187
4.1.3	输入/输出系统的性能	189
4.2	海量存储系统	193
4.2.1	RAID 系统	193
4.2.2	存储域网络	198
4.3	输入/输出总线实例	204
4.3.1	USB 总线	204
4.3.2	PCI Express 总线	213
	习 题	218
第 5 章 并行计算机系统结构		
5.1	并行处理器系统	221
5.1.1	集中式和分布式存储器系统	222
5.1.2	均匀访存与非均匀访存系统	224
5.1.3	芯片级并行系统与系统级并行系统	226
5.1.4	并行处理器系统的性能	229
5.2	并行计算机的互连网络	231
5.2.1	互连网络的分类	231
5.2.2	互连网络的特性	233

目 录

5.2.3 静态互连网络	236
5.2.4 动态互连网络	241
5.3 芯片级并行性	249
5.3.1 多线程并行性	249
5.3.2 多核并行性	255
5.3.3 芯片级互连网络	257
5.3.4 并行处理器芯片实例	259
5.4 系统级并行性	262
5.4.1 板级并行系统	262
5.4.2 集群并行系统	264
5.4.3 分布式系统	271
习 题	273
第6章 并行计算机的同步与通信	
6.1 并行计算机系统的通信	277
6.1.1 共享存储器通信	278
6.1.2 互连网络的消息传递通信	282
6.2 cache与存储器数据一致性	288
6.2.1 cache一致性概念	288
6.2.2 总线监测方法	291
6.2.3 目录表方法	295
6.2.4 存储器数据一致性	297
6.2.5 支持数据一致性的通信接口	299
6.3 并行计算机的同步	302
6.3.1 硬件原语	302
6.3.2 用一致性机制实现锁	304
6.3.3 屏障同步	305
6.3.4 事务存储器	309
6.3.5 同步与多线程	311
6.4 并行计算机程序的软件支持	313
6.4.1 并行程序的概念	313
6.4.2 OpenMP	320
6.4.3 MPI	326
6.4.4 其他并行程序软件工具	329
习 题	332
参考文献	338

第 1 章

计算机系统结构概论

计算机系统是一个十分复杂的系统,研究计算机系统不仅要从微观结构着手,研究如何运用数字逻辑电路技术构成计算机的各种部件,而且要从宏观结构考虑,使我们能够用已有的技术和功能部件,以合理的成本实现性能更高的计算机系统,并使计算机硬件适合于软件和应用的需求,这样就引出了计算机系统结构的概念。本章介绍计算机系统结构的概念、系统结构的分类以及性能评价的方法。

1.1 概 述

计算机系统结构是计算机系统的宏观设计,它影响到硬件与软件的界面。影响计算机系统结构的因素主要是计算机程序设计语言、软件、应用和元器件技术。本节将介绍这方面的内容。

1.1.1 计算机系统组成与系统结构的概念

对于计算机系统的研究可以从系统结构、组成和实现方式这 3 个层次进行。计算机实现方式研究计算机的物理设计,它包括器件的设计与选择、各模块的机械参数设计、印制电路板的设计与制作、机箱的物理特性的设计和选择,以及芯片封装和部件组装技术的设计和选择,还有电源、冷却方式的设计和选择等。计算机实现方式主要取决于器件技术。器件技术是构成计算系统的基础,它对计算机的组成和系统结构设计具有重要影响,涉及物理学、机械设计等许多学科的知识。

计算机系统组成(computer organization)从逻辑电路角度研究计算机系统中各个组成部分的构成方法,包括 CPU 中运算器的电路设计、存储系统的电路设计、控制器的控制方式和数字电路设计、输入/输出接口的设计等。这方面的内容构成了计算机系统的内部特性,计算机的使用者和软件开发者不需要掌握这些特性。

计算机系统结构(computer architecture)研究计算机的概念性结构、功能和性能特性。它从一个更高的层次对计算机的结构和特征进行研究。这一层次不是研究计算机各功能模块的具体逻辑结构,而是对系统的宏观特性进行研究。这些宏观特性是上层软件所看到的特性,构

第1章 计算机系统结构概论

成计算机系统的外特性,即程序员看到的特性。由于计算机系统是包括软、硬件乃至固件资源的复杂系统,处在不同层次的程序员所看到的外特性是不完全相同的。所以,确切地说,计算机系统结构的外特性应是系统程序员所看到的外特性,即由系统软件设计者所看到的计算机的基本属性。如果对于两种不同逻辑电路的计算机,程序员所看到的特性是一样的,那么这两种计算机就具有相同的系统结构。程序员看到的最主要的特性是机器指令集合,它构成指令集系统结构(instruction set architecture)。具有软件兼容性的计算机产品具有相同的指令集系统结构。

对于一个计算机系列产品,每个具体型号产品的实现是各不相同的,而保持不变的特性就是该计算机的指令集系统结构。同一指令集系统结构的不同型号的计算机产品之间的主要区别是系统配置、系统性能和系统内部特征上的差别。在技术高速发展的时代,一个计算机产品的生命周期是短暂的,而计算机系统结构,特别是指令集系统结构则是相对稳定的。设计良好的计算机系统结构可以使计算机产品具有较长的生命周期。因此,在设计计算机产品时,应首先对其系统结构进行研究和设计,以使计算机产品系列具有长久的生命力。选择计算机产品时,除了要考虑软件兼容性等因素外,也要考虑选择系统结构优秀的产品。

作为一门学科,计算机系统结构研究的范围包括:

(1) 计算机外特性的定义

包括计算机的指令系统,即硬件能够直接识别并执行的操作命令的类型和格式的定义;数据表示,即硬件能直接识别和处理的数据类型;寻址方式,即指令对数据存储单元的指定方式;寄存器定义,即寄存器类型、数量和使用方式的定义;中断机构,即中断响应机制的定义;以及输入/输出系统和保护机制等的定义。

(2) 计算机软/硬件界面的设计

确定硬件的功能范围,确定各功能模块的设置以及连接方式,也就是确定哪些功能由软件实现,哪些功能由硬件实现。为此,计算机系统结构设计需要结合操作系统、程序设计语言、算法以及器件技术等多方面的因素,需要在性能和成本之间进行权衡。由于软件和硬件在逻辑功能上的等价性,软/硬件的界面可在很大范围内变化。计算机系统结构的设计需要对各种因素进行权衡,即在各种选择之间进行取舍。

(3) 新型计算机系统的设计

开发适合新型计算模型和具有更高性能的计算机系统,特别是并行计算机系统的设计,即开发具有更高并行性的新型CPU,以及由运算器或CPU为功能模块构成的更大系统。新型计算机的一个基本特征是并行性的开发,而并行系统可以有許多不同的构成方式,因此寻找一种并行计算机结构和运行机制,使其具有高效的运算性能,是计算机系统结构的重要研究内容。

(4) 计算机系统的性能、成本和可靠性的分析和评价

这些量化的评价结果是系统结构设计的重要依据,也是用户选择计算机的重要依据。影

响计算机性能的主要因素有 CPU 的运算速度、内存的容量和速度、输入/输出系统的传输带宽等。不同的计算机具有不同的结构特征,使同一种计算机对不同的应用程序有不同的性能表现。寻找一种适当的性能评价方法以客观反映计算机系统的性能也是系统结构的研究范围。

计算机中指令的集合构成计算机的指令系统,是计算机的最主要的外部特征。程序员对计算机的了解最主要的是计算机的指令集合以及每条指令的行为特征。计算机的指令反映了硬件所具有的功能,并在一定程度上反映了计算机的结构特征。指令系统构成了计算机系统的软件与硬件之间的界面,是软/硬件功能划分的结果。反映计算机指令系统特征的研究领域以及计算机指令系统设计是计算机系统结构学科领域的一个重要方面。

计算机中各个部件的连接结构也构成计算机系统的重要特征,它影响着计算机系统的性能。从连接结构上,早期的计算机系统结构主要以运算器为中心,包括控制器、存储器以及输入/输出单元。所有的输入/输出活动都必须经过运算器。存储器中存放有指令及数据。这一结构是美国科学家冯·诺依曼(von Neumann)首先提出来的,所以称为冯·诺依曼结构计算机。冯·诺依曼结构计算机的主要外部特征是:

① 指令和数据都以二进制数据字的方式存放在同一个存储器中。计算机中只有一个存储器,由计算机的状态来确定从存储器读出的字是指令或是数据。指令送往控制器译码,数据送往运算器进行运算,硬件并不对来自存储器的指令或者数据进行判别,软件需要保证这种正确性。

② 指令按顺序串行地执行,并由一个控制器集中控制,采用一个程序计数器构成顺序指令的地址序列。

③ 存储器是一个单元定长的一维线性空间。存储器的地址是一个一维的数列,二维或者多维的数据结构需要映射到这个一维的空间中。

④ 使用低级机器语言。指令中包括操作信息和数据地址信息。操作信息的编码格式与数据的编码方式相互独立。

⑤ 单 CPU 结构,以运算器为中心,只有 1 个数据流和 1 个指令流。

半个多世纪以来,计算机系统结构已发生了许多变化。这些变化可归纳成两种类型。一种是改进性的变化,即基本上仍保留原来的结构模式,但需进行不断的改进以提高计算机系统的性能。另一种是革命性的变化,即采用与冯·诺依曼结构完全不同的计算模型以及系统结构。改进性的系统结构中,主要特点是在冯·诺依曼结构的基础上增加新的数据表示,如浮点数、字符串、十进制数的表示和多媒体数据表示;增加新的功能部件、新的操作指令;采用指令存储器与数据存储器(cache)分离的哈佛结构;采用虚拟存储器,扩展了程序的工作空间,方便了程序管理和高级语言程序设计;增设高速缓冲存储器(cache),以减少 CPU 与存储器间的过分频繁的数据传输;增加存储器带宽;采用流水技术(包括指令级流水和运算级流水),以加快指令及操作执行速率;采用多个运算和访存功能部件,使多个部件并行运算,从而提高系统的

第1章 计算机系统结构概论

运算速度。改进型系统结构从原来以运算器为中心逐步演变为以存储器为中心,并不断提高计算中的并行性。改进型系统结构的优点在于它能够保持与现有系统兼容,用户容易掌握,因而能够得到更为广泛的应用。

系统结构革命性的发展如20世纪70年代出现的数据流计算机系统结构、80年代出现的推理机系统结构、90年代蓬勃发展的人工神经网络计算机系统以及新世纪兴起的量子计算机系统研究等。数据流计算机根据数据运算之间的数据传递关系来启动并行运算,只要一个运算的操作数具备,就可以进行运算,运算完成后形成的结果就可以使其他的运算能够进行。数据运算不再按照程序员安排的顺序串行地进行,而是由硬件机制控制下并行地进行。推理机模仿人类逻辑推理的过程进行计算,可以完成复杂逻辑思维功能。人工神经网络计算机模仿动物神经系统处理外部信息的原理和过程,构造一个高度连接、高度并行的电子模拟神经网络,其中每个神经单元既是信息存储功能部件,又是信息处理功能部件。这样构成的信息处理系统具有生物的学习特征,而不需要在程序的控制下运行。这些新型计算机采用与传统计算机不同的计算模型和程序设计语言,适合于某一领域的计算,并不能代替传统的通用计算机系统而成为主流机型。量子计算机则是利用量子物理学的研究成果,研究亚原子结构的计算机制。

在采用指令流水线结构的计算机中,流水线结构的设计既与计算机组成有关,又与计算机系统结构有关。指令流水线中包含了控制器、寄存器和运算功能部件,构成了CPU的核心结构。这些部件的相互连接和协调工作是逻辑电路设计的范畴。指令流水线的结构会影响到指令的执行,不仅影响指令的执行速度,还会影响到指令序列执行的正确性。指令流水线的调度和并行执行有时还需要编译程序等软件的协助和配合。这样,指令流水线结构的设计成为软件需要看到的外部特征,这种特征是介于计算机组成和系统结构之间的。我们把指令流水线的结构称为微观系统结构(microarchitecture)或微系统结构。在第2章中将介绍计算机微观系统结构的内容。

以上介绍了计算机系统结构的概念。计算机系统结构的设计与许多因素相关,主要包括计算机语言、计算机软件、计算机的应用需求和实现计算机的元器件的发展。以下分别介绍这些方面的内容。

1.1.2 语言与系统结构设计

计算机程序设计语言提供了描述应用算法的规范。计算机系统结构的设计目标之一是提供一个指令系统,这个指令系统要能够有效地支持高级程序设计语言中的常见操作,使高级语言中的语句能够表示成较少的机器指令和较短小的指令代码。计算机的高级程序设计语言主要有过程式语言、函数式语言、逻辑推理式语言等类型。各种高级语言都要求系统结构提供一些特殊的支持,以使这些语言能够高效地运行。

过程(process)指的是计算机的子程序。大多数结构化程序设计高级语言采用过程调用

方式形成程序结构,由程序员来安排各种操作的执行顺序。结构化的过程调用提高了程序的可读性、可修改性和可维护性。过程的调用以及返回要求系统结构提供的支持包括:保存调用过程的返回地址、保存调用过程的状态、从调用过程向被调用过程传递参数、将指令流转向被调用过程、为过程的局部变量分配存储空间、在被调用过程执行时访问参数、恢复调用过程的状态、回收局部存储空间和将指令流转向调用过程。此外,结构化程序设计语言还要求系统结构支持结构化数据类型,支持高级语言的控制流结构,支持程序的模块化。有些程序设计语言要求支持指针操作,有些语言要求支持动态数据类型的检查,即程序运行期间对数据类型的检查,还有一些语言要求执行时进行存储器访问的边界检查等。这些要求需要硬件和指令系统提供相应的支持,也将影响到系统结构的设计。面向对象的高级语言也是一种过程式程序设计语言,它将数据结构及其相应的操作方法构成一个封装,形成对象的概念。目前,广泛采用的 C/C++ 语言等都是过程式语言。

不同的过程式语言也会要求系统结构提供不同的支持,如对于 Java 语言的支持就形成了 Java 虚拟机(JVM)的概念。它可以用软件实现,也可以采用硬件实现。用硬件支持 Java 虚拟机需要在指令系统中提供相应的指令。虚拟机是在软件和硬件的基础上建立起的一个计算机平台。在这个平台上对软件形成了一个新的计算机的视图。

除了过程式语言以外,计算机的高级语言还有函数式语言(如 Lisp)、逻辑推理型语言(如 Prolog)等,这些语言都需要采用相应的系统结构以支持其高效的运行。如在硬件上支持 Lisp 语言和 Prolog 语言就需要采用不同的指令系统。这些语言要求实现与程序顺序无关的执行顺序,并提供相应的执行控制功能。反过来,系统结构也对语言的发展产生影响,如在数据流计算机的发展过程中,产生了适合于表示数据流语言;又如并行计算机的发展对并行程序设计语言的发展也产生了重要影响,出现了一些并行程序设计语言。

计算机程序设计语言在硬件的基础上进行虚拟化,建立起系统资源的不同视图,即虚拟机。例如,JVM 建立起运行 Java 应用环境的视图。虚拟机可以分为运行时(runtime)虚拟机和系统虚拟机两种。运行时虚拟机通过软件实现,作为一个运行的进程,运行在操作系统之上,可以看作是一个容器或者执行器,为应用程序的运行建立起一个环境,并且监控应用程序的执行。Java 虚拟机是一种运行时虚拟机,可以执行 Java 字节代码。它至少建立起 3 个应用级线程:执行线程、废区收集线程和 JIT 编译线程。线程是一个程序运行和调度的单位。一个进程中可以包含多个线程。同一个进程中的各个线程具有共享的代码区 and 数据区,但每个线程具有自己的堆栈区,保存各自的运行状态和局部变量。Java 虚拟机中的废区收集线程用于收集不再使用的存储器区域。JIT 编译线程将字节代码编译成机器代码。这种虚拟机通常还创建其他线程以完成内部的任务。运行时虚拟机与操作系统协同将这些线程映射成执行资源以达到最佳的资源使用。

计算机系统的虚拟化建立起一种与实际硬件不同的虚拟机。虚拟机平台可以有不同层次,JVM 属于应用程序建立起来的支持 Java 语言的虚拟机。操作系统可以向应用程序提供

第1章 计算机系统结构概论

虚拟机平台,使不同的应用程序进程之间相互隔离,就像各自单独运行在一台计算机上一样。同样,底层的监控软件也可以在硬件的基础上建立起虚拟机,使多个操作系统能够运行在这个平台上。这种虚拟机构成系统虚拟机,它建立起一个完整的系统软件执行环境。各个虚拟机使用虚拟化的网络接口和磁盘,运行各自的操作系统软件。这样建立的多个虚拟机可以运行在同一个硬件平台中,多个同时运行的操作系统共享一个硬件平台。建立这个虚拟化层的软件称为虚拟机监控器(VMM)。VMM 提供底层平台的虚拟化,使每个虚拟机中的操作系统感觉不到虚拟机的存在,就像占用了全部硬件资源一样。这种虚拟机建立起虚拟处理器的概念,能够提供更加有效的底层的物理处理器硬件资源,但是需要系统结构提供相应的支持。

再向下层,微程序可以通过指令仿真建立起不同指令系统的虚拟机,甚至可以实现动态的指令系统。这种虚拟机建立在微程序的基础上,是在早期的计算机中主要为了解决软件兼容问题而提出的。由于指令仿真能降低系统的性能,所以现在很少采用这种虚拟机。

1.1.3 软件与系统结构设计

软件对计算机系统结构的设计有很大的影响。这种影响主要体现在编译程序和操作系统这两个方面。

编译程序是将用一种高级程序设计语言编写的源程序转换成由机器语言描写的与源程序等价的目标程序的软件。现今绝大多数程序都是用高级语言编写的,这意味着大多数指令是由编译程序生成的,编译程序需要根据计算机的指令系统特征来生成指令代码。早期的计算机较多地考虑了便于汇编程序的设计,现在设计指令系统时则必须考虑到高级语言和编译技术的影响,考虑指令的并行执行,使计算机具有最佳的性能。编译程序生成的指令代码与人工编码的指令代码具有不同的统计特征。提高大部分目标代码中的指令的执行速度是系统结构设计中需要考虑的一个重要因素。

编译技术对系统结构设计的影响有 3 个方面:数据的存储结构、指令序列的特征和控制结构特征。数据的存储结构影响到数据的访问方式。现行高级语言采用 3 种数据存储结构:堆栈、全局数据区和堆区。堆栈可用于存放函数或过程间传递的参数。全局数据一般是一些数组和其他结构的数据。堆区存放动态数据,通常用指针访问。

指令序列的特征反映出各种指令的使用概率,以及指令之间的顺序关系等。在指令系统设计中,需要根据编译程序产生的指令特征,根据程序的实际需要设置指令。统计表明,编译程序生成的计算机指令不是在各种指令之间均匀分布的,编译生成的大部分指令是全部指令集合中的一小部分。例如,在 80x86 的指令系统中,有 10 条指令是最常用的,占了 96%,如表 1-1 所列(摘自参考文献[1])。提高这 10 条指令的执行速度对于提高计算机的性能至关重要。在提高计算机性能的时候,首先要考虑提高这些指令的执行速度。

表 1-1 80x86 中的 10 条常用指令的使用概率

名 次	指 令	执行概率百分比
1	load	22%
2	bxx	20%
3	compare	16%
4	store	12%
5	add	8%
6	and	6%
7	sub	5%
8	move register-register	4%
9	call	1%
10	return	1%

计算机系统性能在很大程度上依赖于编译程序的代码优化。通过代码优化可以减少程序中的冗余指令,提高程序的执行速度。代码优化可以对寄存器的使用进行合理安排,以减少对主存的访问。在采用流水技术的计算机中,代码优化还可为避免流水线停顿而进行指令代码的调度。一些新型微处理器都通过静态指令调度来开发指令级和循环级的并行性,静态指令调度依赖于编译技术,有关这方面的内容将在下一章中详细介绍。所以,系统结构对编译程序的设计有很大的影响,反之,编译技术也对系统结构设计产生重要的影响。

计算机系统结构须对操作系统提供支持,因而操作系统的发展影响到系统结构的设计。操作系统可分为单用户系统、多用户系统和实时系统等,不同类型的操作系统对系统结构有不同的要求。系统结构对操作系统的支持主要体现在硬件上和指令系统上支持系统服务机制、存储器管理与存储保护、进程同步机制或线程同步机制、进程切换机制或线程切换机制、故障诊断机制等。操作系统为应用程序的运行提供支持,并隐藏底层硬件的差别。操作系统在硬件的基础上建立起多个进程和线程,为多个应用程序共享计算机资源提供支持。在多核等并行计算机系统中,也需要操作系统提供支持,以充分发挥多个处理器或者处理器核的计算能力,使多个处理器相互协调地运行一个应用程序。

早期操作系统以进程为单位对程序的运行进行管理。每个程序在运行之前需要将程序代码装入内存的某个区域,此外还要建立起程序的数据存储区域和堆栈区域,这样就建立起一个进程。一个应用程序可在计算机中重复启动运行,从而形成多个进程。为了避免同一个程序的 2 次运行事例中在存储器中建立起 2 个重复的代码存储区域,新型操作系统都支持线程的管理。线程作为进程中的一个对象,具有自己的堆栈区,但是与同一个进程中的其他线程共享代码区域和数据存储区域,从而可以减少存储代码的开销,并且便于线程间的数据共享。多线程的方式可以更好地支持并行计算机的运行。具有并行计算特征的应用程序可以被编译程序

第1章 计算机系统结构概论

分解成多个线程,由操作系统将它们分配在一个或者多个处理器核上运行,从而加快程序的运行速度,提高计算机的性能。为了协调线程的执行,操作系统需要提供线程同步机制、通信机制和互斥机制等,这些机制最终需要由系统结构提供底层的支持。

操作系统的系统服务机制是在运行中为应用程序提供的系统调用机制。操作系统在运行时将一些基本操作的子程序调用到内存中,这些系统子程序可以被应用程序调用,形成系统调用。操作系统通过提供系统调用为应用程序提供运行服务,如磁盘访问和文件系统的操作等常用子程序的调用。这种系统函数调用通常采用中断机制实现。操作系统的存储器管理功能要求系统结构支持地址的变换、访问权限的检查等。操作系统还要控制各个进程在执行时间上的相互协调,即进程或线程同步。线程同步通常通过对共享变量的访问来实现。这种同步机制要求系统结构提供专门的指令以实现对共享变量的原子的访问(不可打断的访问)。进程切换时要求保存大量的进程状态信息(称为进程现场),系统结构支持的进程切换机制应当能够高效地实现进程现场信息的保存。故障诊断要求系统结构提供保存故障信息以及指令跟踪功能。许多计算机系统提供了硬件支持的故障诊断机制,如在 Intel 的微处理器系统结构中提供了故障诊断寄存器。

1.1.4 应用需求与系统结构设计

计算机的应用领域与计算机系统结构设计密切相关。现在,计算机已经被应用于人类生产和生活的各个方面,包括科学计算、工程和信息处理等。不同的应用需求自然需要有不同的系统结构的计算机。

科学计算的应用领域包括气象学、天文学、量子化学、空气动力学、核物理学、图像学、模式识别、基因工程学、分子生物学、医药学等,可用于进行新型药物设计、生物分子结构、催化剂和酶的性质、人类基因、新材料性质、湍流、海洋回流、核聚变能发电、核爆炸模拟、加密和解密、全球天气预报、灾害性风暴预报、地震预测、地质勘探等研究。这些科学研究领域的发展对于人类社会文明的发展具有非常重要的影响,也对计算机的性能提出了挑战。用计算机进行计算已成为与理论分析和实验并列的第三种科学研究手段。科学计算的特点是计算量大,数值变化范围广,计算精度要求高,因此又称为数值计算。它要求计算机具有很高的运算速度,很大的内存容量,对输入/输出能力的要求则相对较低。

计算机在工程中的应用包括工程设计、自动控制和自动测量等。工程设计是近年来新兴的计算机应用领域。计算机辅助工程设计的特点是涉及大量的图形数据的处理。在工程设计系统中,要求计算机具有高速三维图形功能。在自动控制领域,计算机被应用于生产过程和设备工作过程的控制。在控制应用中一般采用嵌入式计算机系统,将计算机系统与应用系统相结合,构成一个整体。这种系统通常要求计算机具有实时性,即计算机对外界的反应要求及时,能够及时发出操作控制命令。计算机自动测量设备可对测试数据进行分析 and 转换,形成智能化的测量仪器。