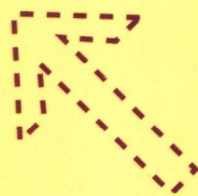


计算机与信息技术专业应用教材

计算机系统结构 教程



尹朝庆 编著



清华大学出版社

► 计算机与信息技术专业应用教材

计算机系统结构教程

尹朝庆 编著

清华大学出版社

北京

内 容 简 介

本书以提高计算机并行性的方法和技术为主线,以性能分析为依托,根据教学大纲介绍各类计算机的体系结构、关键技术和性能分析,并附以系统实例。

本书由两部分组成。第一部分主要介绍 SISD 计算机提高并行性的有关技术和性能分析,包括导论(第1章)、指令系统(第2章)、流水技术与流水处理机(第3章)和存储系统(第4章)。第二部分介绍属于并行计算机范畴的各类计算机的体系结构与关键技术,包括互连网络(第5章),SIMD 计算机的结构、常用算法和性能分析(第6章),MIMD 计算机的结构模型和访存模型,SMP、MPP、机群、SVM 等当代可扩充并行计算机体系结构,并行算法性能分析等(第7章)。

本书取材先进、内容精炼、体系完整,可作为高等院校计算机及相关专业的教学用书,也可供相关科技人员参考。

版权所有,翻印必究。举报电话:010-62782989 13901104297 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。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术,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

计算机系统结构教程/尹朝庆编著. —北京:清华大学出版社,2005.3

(计算机与信息技术专业应用教材)

ISBN 7-302-10626-6

I. 计… II. 尹… III. 计算机体系结构 IV.TP303

中国版本图书馆CIP数据核字(2005)第019189号

出版者:清华大学出版社

<http://www.tup.com.cn>

社总机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

组稿编辑:夏非彼

文稿编辑:洪英

封面设计:付剑飞

版式设计:科海

印刷者:北京市耀华印刷有限公司

发 行 者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:19 字数:462千字

版 次:2005年4月第1版 2005年4月第1次印刷

书 号:ISBN 7-302-10626-6/TP·7198

印 数:1~4000

定 价:26.00元

丛 书 序

为适应信息社会高速发展的需求，目前全国各类高等院校都在进行计算机教学的全方位改革，目的是规划出一整套面向计算机与信息技术专业、具有中国高校计算机教育特色的课程计划和教材体系，本丛书就是在这一背景下应运而生的。我们组织了由全国高校计算机专业的专家教授组成的“计算机与信息技术专业应用教材”课题研究组，通过对计算机和信息技术专业全方位的研讨，并结合我国当前的实际情况，编写了这套系统性、科学性和实践性都很强的丛书。

丛书特色

☑ 先进性：力求介绍最新的技术和方法

先进性和时代性是教材的生命，计算机与信息技术专业的教学具有更新快、内容多的特点，本丛书在体例安排和实际讲述过程中都力求介绍最新的技术和方法，并注重拓宽学生的知识面，激发他们学习的热情和创新的欲望。

☑ 理论与实践并重：阐明基础理论，强调实践应用

理论是实践的基础，实践是理论的升华；不能有效指导实践的理论是空头理论，没有理论指导的实践是盲目的实践。对于时代呼唤的信息化人才而言，二者缺一不可。本丛书以知识点为主线，穿插演示性案例于理论讲解之中，使枯燥的理论变得更易于理解、易于接受；此外，还在每一章的末尾提供大量的实习题和综合练习题，目的是提高学生综合利用所学知识解决实际问题的能力。

☑ 易教易学：创新体例，合理布局，通俗易懂

本丛书结构清晰，内容系统详实，布局合理，体例较好；力求把握各门课程的核心，通俗易懂，便于教学的展开，也便于学生学习。

丛书组成

首批推出的计算机与信息技术专业应用教材涵盖计算机基础、程序设计和数据库三大领域，共 16 本：

- 操作系统
- 计算机系统结构教程
- 数据结构与算法教程
- Java 语言程序设计
- Access 程序设计教程
- C 程序设计教程（基于 Visual C++平台）
- C 程序设计教程学习与上机指导（基于 Visual C++平台）
- C++程序设计教程
- C++程序设计教程学习与上机指导
- Visual FoxPro 程序设计教程
- Visual Basic 程序设计教程
- SQL Server 2000 关系数据库教程
- SQL Server 2000 关系数据库教程学习与上机指导
- 数据库原理与应用——基于 Access
- 数据库原理与应用——基于 Visual FoxPro
- 数据库原理与应用——基于 SQL Server 2000

丛书编委会

主任委员：李春葆
 副主任委员：苏光奎 朱福喜
 委员：尹为民 尹朝庆 李春葆 伍春香
 朱福喜 苏光奎 胡新启 徐爱萍
 曾 平 曾 慧

编者寄语

如果说科学技术的飞速发展是 21 世纪的一个重要特征的话，那么教学改革将是 21 世纪教育工作不变的主题。要紧跟教学改革，不断创新，真正编写出满足新形势下教学需求的教材，还需要我们不断地努力实践、探索和完善。本丛书虽然经过细致的编写与校订，仍难免有疏漏和不足，需要不断地补充、修订和完善。我们热情欢迎使用本丛书的教师、学生和读者朋友提出宝贵意见和建议，使之更臻成熟。

本丛书作者的电子邮件：licb@public.wh.hb.cn

本丛书出版者的电子邮件：feedback@khp.com.cn

2005 年 3 月

前 言

高性能计算机是国家经济和科技实力的综合体现，是促进经济和科技发展、社会进步和国防安全的重要工具。近 10 年来，高性能计算机得到迅速发展，高端产品的运算速度正向每秒百万亿次、千万亿次迈进。我国对高性能计算机的研究也十分重视，近几年来，不断推出具有相当高水平的并行计算机系统。高性能计算机的应用从科学研究、工程计算、虚拟试验等领域发展到更多商务应用领域，在社会信息化建设中将发挥十分重要的作用。

计算机系统结构(Computer Architecture)是高等院校计算机专业的一门专业主干课程。由于提高计算机性能的主要推动力是发展计算机体系结构的并行性，因此，在计算机科学与技术高速发展的今天，讲授和学习计算机系统结构，应以计算机并行化的概念、方法和技术为主线，以性能分析方法为依托，并将不断涌现的新概念、新技术、新方法及时补充到教学内容中去。这正是作者编著本书的宗旨。

本书共分 7 章，各章内容如下：

第 1 章导论，介绍计算机体系结构的基本概念、基本性能指标和并行计算机的发展。

第 2 章指令系统，介绍数据表示和指令格式优化设计方法，以及精简指令集计算技术的概念和关键技术。

第 3 章流水技术与流水处理机，讨论流水处理机的分类、调度和相关处理等关键技术以及性能分析方法；介绍几种多发射处理机的结构与性能比较；介绍向量流水处理机的结构和提高性能的方法。

第 4 章存储系统，介绍存储系统的层次结构与性能指标；介绍两种并行存储器；讨论虚拟存储器和高速缓冲存储器的地址变换和替换算法及其实现技术以及对存储系统性能的影响。

第 5 章互连网络，介绍规则定制的互连网络表示互连的互连函数，以及互连网络的结构参数和性能指标；讨论几种静态互连网络和动态互连网络的结构和性能；讨论由互连网络互连的处理结点之间的消息传递机制。

第 6 章单指令流多数据流计算机，介绍单指令流多数据流计算机的两种基本结构，并对两种结构的机器实例进行分析；讨论阵列处理机的常用算法和性能分析。

第 7 章多指令流多数据流计算机，在介绍多指令流多数据流计算机的结构模型和访存模型的基础上，主要围绕当代可扩放的并行计算机体系结构，讨论对称多处理机系统、大规模并行处理机系统、机群系统和分布式共享存储系统等主流并行计算机的结构和关键技术；最后，对并行算法的性能评测进行分析和讨论。

本书取材先进、内容简炼、由浅入深、体系完整。各章都精选了一批例题和习题，帮助读者进一步理解讲授的概念、技术和方法。本书各章的大部分习题在《计算机系统结构习题与解析》(尹朝庆编著，清华大学出版社，2004)一书中都有详细的解答。

作者在编写本书时，曾直接或间接地引用了许多专家、学者的文献和著作，在此向他们深表谢意。本书内容也曾由作者讲授过多次，历届学生曾提出过许多宝贵意见，他们的意见对本书内容的选取与组织有很大帮助，使之既取材先进，力图反映最新成果和发展趋势，又便于教学，在此一并表示感谢。

本书可作为高等学校计算机、自动化及电子工程等相关专业的高年级本科生、研究生的教材，也可作为有关专业的教师、研究人员和工程技术人员的参考书。

由于作者学识有限，写作时间仓促，疏漏和错误之处在所难免，恳请读者不吝批评指正。

尹朝庆
2005年3月

目 录

第 1 章 导论	1
1.1 计算机系统结构的基本概念.....	1
1.1.1 计算机系统的层次结构.....	1
1.1.2 计算机系统并行性的基本概念.....	6
1.2 计算机性能评测.....	9
1.2.1 计算机性能评测的作用与分级.....	10
1.2.2 计算机的基本性能指标.....	13
1.3 并行计算机的发展.....	19
1.3.1 支持并行计算机的技术进展.....	20
1.3.2 并行结构的发展.....	21
习题 1.....	23
第 2 章 指令系统	26
2.1 数据表示.....	26
2.1.1 浮点数据表示.....	26
2.1.2 自定义数据表示.....	30
2.2 指令格式优化设计.....	32
2.2.1 操作码优化设计.....	32
2.2.2 寻址技术.....	36
2.2.3 指令字格式优化设计的措施.....	38
2.3 精简指令集计算技术.....	40
2.3.1 RISC 的发展与特点.....	40
2.3.2 RISC 的关键技术.....	44
习题 2.....	48
第 3 章 流水技术与流水处理机	51
3.1 指令重叠与先行控制.....	51
3.1.1 指令的顺序执行方式与重叠执行方式.....	51
3.1.2 先行控制技术.....	52
3.2 流水线的分类与性能.....	54
3.2.1 流水线的分类与时空图.....	54
3.2.2 线性流水线的性能计算.....	58
3.3 非线性流水线的调度与性能计算.....	66

3.3.1	非线性流水线的最优调度方法与性能计算	66
3.3.2	非线性流水线最优调度的时空图	73
3.3.3	非线性流水线的动态调度的实现	74
3.4	流水线的相关问题与相关处理	75
3.4.1	局部相关及处理	76
3.4.2	全局相关及处理	77
3.4.3	相关对流水线性能的影响	81
3.5	多发射处理机及其性能	83
3.5.1	超标量处理机及其性能计算	83
3.5.2	超流水处理机及其性能计算	86
3.5.3	超标量超流水处理机及其性能计算	87
3.5.4	超长指令字处理机的基本结构与特征	88
3.5.5	多发射处理机的性能比较	91
3.6	向量流水处理机及其性能	94
3.6.1	向量处理方式	94
3.6.2	向量流水处理机的结构	96
3.6.3	向量流水处理机的性能	100
习题 3		102
第 4 章	存储系统	108
4.1	存储系统的层次结构与性能指标	108
4.1.1	存储系统的层次结构	108
4.1.2	存储系统的性能指标	110
4.2	并行存储器	112
4.2.1	单体多字并行存储器	112
4.2.2	低位交叉编址多体并行存储器	113
4.3	虚拟存储器	115
4.3.1	虚拟存储器的地址变换	115
4.3.2	页面替换算法及其命中率	123
4.3.3	堆栈型替换算法及其堆栈处理过程	124
4.4	高速缓冲存储器	129
4.4.1	Cache 的地址映像与地址变换	130
4.4.2	Cache 的替换算法及其实现	139
4.4.3	Cache 的性能分析	141
4.5	三级存储系统	145
习题 4		146
第 5 章	互连网络	152
5.1	互连函数	152

5.1.1 互连函数的表示方法.....	152
5.1.2 几种基本的互连函数.....	153
5.2 互连网络的结构参数与性能指标.....	161
5.2.1 互连网络的结构参数.....	161
5.2.2 互连网络的性能指标.....	162
5.3 静态互连网络.....	163
5.4 动态互连网络.....	169
5.4.1 多处理机总线.....	169
5.4.2 交叉开关.....	171
5.4.3 多级互连网络.....	173
5.4.4 动态互连网络的比较.....	189
5.5 消息传送机制.....	190
5.5.1 消息格式与寻径方式.....	190
5.5.2 寻径方法与多播通信.....	193
习题 5.....	196
第 6 章 单指令流多数据流计算机.....	199
6.1 单指令流多数据流计算机的基本结构与特点.....	199
6.1.1 单指令流多数据流计算机的两种基本结构.....	199
6.1.2 单指令流多数据流计算机的主要特点.....	202
6.2 单指令流多数据流计算机实例分析.....	203
6.2.1 ILLIAC IV 计算机系统.....	203
6.2.2 BSP 计算机.....	208
6.3 阵列处理机的算法及性能分析.....	211
6.3.1 阵列处理机的差分计算.....	212
6.3.2 阵列处理机的常用算法及性能分析.....	213
习题 6.....	218
第 7 章 多指令流多数据流计算机.....	220
7.1 MIMD 计算机结构模型与访存模型.....	220
7.1.1 MIMD 计算机结构模型.....	220
7.1.2 MIMD 计算机访存模型.....	224
7.2 对称多处理机系统.....	228
7.2.1 对称多处理机的结构.....	228
7.2.2 高速缓存一致性问题.....	230
7.2.3 侦听高速缓存一致性协议.....	232
7.2.4 基于目录的高速缓存一致性协议.....	240
7.2.5 SMP 实例分析: SGI Challenge.....	244
7.3 大规模并行处理机系统.....	247

7.3.1 大规模并行处理机的结构	247
7.3.2 MPP 实例分析: SGI/Cray Origin 2000	249
7.4 机群系统	255
7.4.1 机群系统的结构	255
7.4.2 机群系统的单一系统映像	257
7.5 分布式共享存储系统	260
7.5.1 分布式共享存储的基本概念	260
7.5.2 共享虚拟存储系统	261
7.6 并行算法的性能评测	266
7.6.1 并行算法的性能指标	266
7.6.2 任务粒度对性能的影响	277
习题 7	281
专业术语中英文对照表	285
参考文献	292

第 1 章

导 论

计算机系统性能的不不断提高，主要是依靠器件的不断更新和计算机系统结构的发展。本章主要介绍计算机系统结构的基本概念、系统结构的发展和计算机的基本性能。

1.1 计算机系统结构的基本概念

回顾计算机的发展历史可以看出，计算机系统性能的不不断提高主要靠器件的变革和计算机系统结构的改进。由于器件的迅速发展，使得计算机硬件在体积、重量、速度、可靠性、稳定性等性能上有了极大的改善，而且价格不断降低。可以说，器件技术的突飞猛进为计算机的发展提供了必不可少的物质基础，它是推动计算机发展的重要因素之一。器件的换代是计算机换代的突出标志。

但是，还应该看到，仅有器件的发展是远远不够的。人们还必须研究如何最合理地组织这些器件，如何最大限度地发挥这些器件的作用，如何构成综合性能最佳的系统，这就是计算机系统结构要研究的问题。恩斯洛（P.H.Enslow）曾经比较了 1965 年~1975 年的器件延迟时间和计算机指令执行时间之间的关系。结果表明，这 10 年间，器件的更新使器件延迟时间降低至原来的十分之一，但计算机指令执行时间却降低至原来的百分之一。由此可见，在这 10 年中，计算机性能提高的幅度比器件性能提高的幅度大得多。这种情况在近几年的计算机发展中更为明显。因此，除器件的发展以外，计算机系统结构的发展也是推动计算机发展的一个重要因素。所以，计算机换代的标志主要有两个，第一是计算机的器件，第二是计算机系统的结构。

推动计算机系统结构发展的关键是提高计算机系统的并行性。计算机系统结构从低级向高级发展的过程也就是计算机系统的并行性不断发展的过程。

1.1.1 计算机系统的层次结构

在计算机系统中使用的计算机语言可以分为一系列的层次（Level）或级，最低层语言的功能最简单，最高层语言的功能最强，更便于应用。从计算机语言的角度，可将通用计算机系统划分成多级层次结构，每一层以一种不同的语言为特征。按由低层到高层的顺序，各层分别是：微程序机器级、传统机器语言机器级、操作系统机器级、汇编语言机器级、高级语言机器级和应用语言机器级等。计算机系统的层次结构如图 1.1 所示。对每一层的

使用者来说，都可以把此机器级看成是一台独立的机器，都可以应用相应的机器级语言。

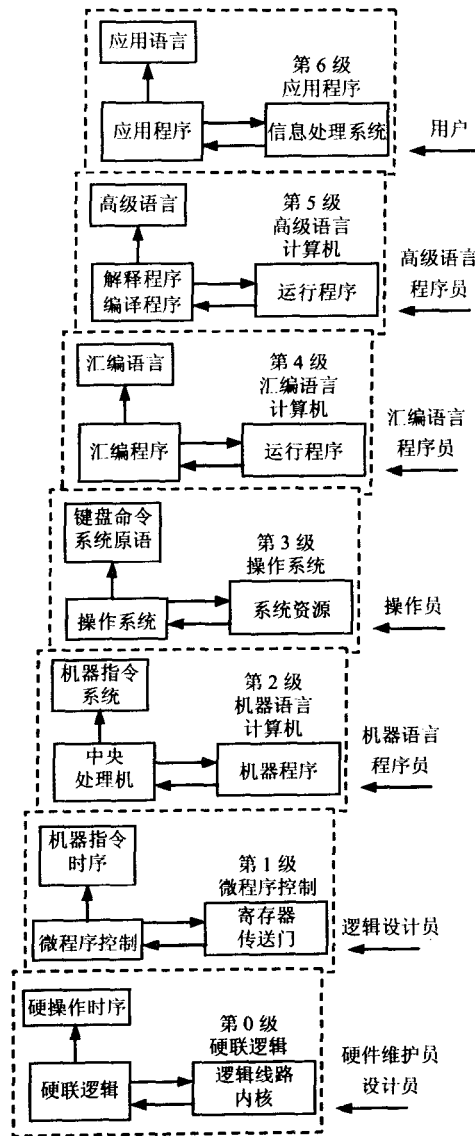


图 1.1 计算机系统层次结构

第0级和第1级是实现机器指定功能的中央控制部分。它根据各种机器指令操作所需要的控制时序，配备一套微指令，编写出微程序，来控制信息在各寄存器之间传送，这就是第1级机器。实现这些微指令本身的控制时序只需要很少的逻辑线路，可采用硬联逻辑实现，它就是第0级机器，是机器的硬件内核。

第2级是传统机器语言机器。这一级的机器语言是该机的指令系统。机器语言程序员用这一级提供的指令系统编写的机器语言程序由第1级的微程序进行解释。

第3级是操作系统机器。这一级的机器语言中的多数指令是机器指令系统中的指令，如算术运算、逻辑运算和传送等指令。此外，这一级还提供操作系统级指令，如打开文件、

读/写文件、关闭文件等指令。在用这一级语言编写的程序中，凡是使用由第2级指令系统提供的指令由微程序解释，而使用的操作系统级指令则由操作系统进行解释。操作系统是运行在第2级机器上的解释程序。

第4级是汇编语言机器。这一级的机器语言是汇编语言。用汇编语言编写的程序首先被翻译成第3级或第2级语言，然后再由相应的机器进行解释。完成汇编语言程序翻译的程序称为汇编程序。

第5级是高级语言机器。这一级的机器语言就是各种高级语言。用高级语言编写的程序一般由编译程序翻译成第4级或第3级机器上的语言。个别的高级语言也用解释的方法实现。

第6级是应用语言机器。这一级的机器语言是应用语言，是面对非计算机专业人员直接使用的语言。只需在用户终端用键盘或其他方式发出服务请求就能进入第6级的信息处理系统。

微程序机器级和用组合逻辑控制的机器语言机器级都是用硬件实现的，而采用微程序控制的机器语言机器级是用固件实现的。固件(Firmware)是一种具有软件功能的硬件，例如，把软件固化在只读存储器中就是一种固件。以硬件或固件为主实现的机器称为实际机器。

当前，应用语言机器级、高级语言机器级、汇编语言机器级和操作系统机器级都是以软件为主实现的，但也可根据用户对计算机性能/价格的不同要求增加硬件支持。以软件为主实现的机器称为虚拟机器(Virtual Machine)。

1. 低层机器级对高层机器级的支持

机器级的层次越高，语言的功能越强大，用户使用起来越方便。但是，各层机器级语言的功能是依靠下一层机器级的支持才能实现的，而且，这种支持要满足透明性要求。

低层机器级为高层机器级的功能实现提供支持，实现支持的技术是翻译和解释。翻译(Translate)是先用翻译程序(Translator)把高层机器级上的程序翻译成低层机器级上的功能等效的程序，然后在低层机器级上实现程序功能。解释(Interpretation)是用低层机器级的一些语句或指令来仿真高层机器级上的一条语句或指令的功能，高层机器级程序中的每条语句或指令被逐条解释来实现程序功能。

采用翻译技术实现的典型例子有：用翻译程序将应用语言机器级上的应用程序包翻译成高级语言程序；用编译程序将高级语言源程序转换成机器语言目标程序；用汇编程序将汇编语言源程序转换成机器语言目标程序。采用解释技术实现的典型例子有：用解释方式在机器语言机器级上执行高级语言程序；用微指令程序解释实现机器指令；用微程序或机器指令程序解释实现操作系统的操作命令等。

从计算机系统的某一层的使用者角度看，只需通过该层的语言就可以使用机器，而不必关心其下层的机器级是如何工作和如何实现对上层的支持的，这一性质称为透明性(Transparency)。

计算机系统中的“透明”与通常意义下的“透明”在含义上正好相反。通常意义下的“透明”是指公开的，什么都能看得到的，例如，玻璃是透明的，所以可以看到玻璃瓶内

的任何物体。计算机系统的“透明”是看不到的意思，即对某一层的使用者来说，他看不到该层以下各层的机器属性。

2. 计算机系统结构、计算机组成与实现

(1) 计算机系统结构

计算机系统结构 (Computer Architecture) 也称为计算机系统结构，经典的计算机系统结构的定义是指计算机系统多级层次结构中机器语言机器级的结构，它是软件和硬件/固件的主要界面，是由机器语言程序、汇编语言源程序和高级语言源程序翻译生成的机器语言目标程序能在机器上正确运行所应具有的结构和功能。

目前，经常使用的是广义的计算机系统结构的概念，它除了包括经典的计算机系统结构的概念范畴外，还包括了计算机组成和实现技术。

(2) 计算机组成与实现

计算机组成 (Computer Organization) 是计算机系统结构的逻辑实现，它主要研究硬件系统在逻辑上是如何组织的，机器级内部数据流和控制流的组成与逻辑设计。计算机实现 (Computer Implementation) 是指计算机组成的物理实现，主要着眼于器件技术和微组装技术。

(3) 结构、组成与实现之间的相互影响

相同结构的计算机可以因速度等性能要求不同而采用不同的组成，相同的组成可有多种不同的实现方法，这都取决于计算机系统的性能和价格要求以及器件技术的发展情况。

结构不同则可用的组成技术会有所不同，而不同的组成又会反过来影响系统结构的设计。因此，系统结构的设计不仅需要结合应用需求来考虑，要能为软件和算法的实现提供更多更好的硬件支持，同时要考虑可能采用和准备采用哪些组成技术，不能过多或不合理地限制各种可能的组成与实现技术的采用。

组成与实现可以折中权衡，主要取决于器件的来源和性能价格比。应该在可提供的器件及其实现技术的条件下，在使价格不增或少增的情况下尽可能提高系统的性能。

3. 软件可移植性

软件可移植性 (Software Portability) 是指在一台机器上编制的软件不用修改或只经少量修改就可在另一台机器上运行，使得同一个软件可以应用于不同的硬件环境中。这样，在过去的计算机系统上使用的大量成熟可靠的软件，特别是应用软件，就可以在新机器上使用而不必重新编写。

在计算机系统结构的范畴内，实现软件可移植性的基本途径有：统一高级语言，采用系列机，以及模拟和仿真等。

(1) 统一高级语言

统一高级语言是指设计出一种对各种应用领域都比较高效率的通用高级语言，在结构相同甚至完全不同的机器之间，通过配备不同的语言翻译程序就可以实现高级语言应用软件的移植。但是，不同的应用对高级语言的语法和语义结构的要求差别较大，使得难以实现一种统一通用的高级语言。从长远的目标来看，还是希望设计出一种统一高级语言，但是，

在目前相当长的一段时间内，统一只能是相对的。

(2) 系列机

系列机是指具有相同的系统结构，但具有不同组成和实现技术的一系列不同型号的机器。它们采用不同的组成和实现技术，从而具有不同的性能和价格，但从机器语言机器级的程序设计者角度来看，其机器属性是相同的。

系列机技术既可以使同一系列的机器在汇编语言上实现统一，又能在同一系列内开发出性能价格比更高的新机器。因此，在结构相同或相近的机器之间能实现汇编语言应用软件和部分系统软件的移植。

软件兼容 (Software Compatibility) 是指同一个软件可以不加修改地运行于结构相同的各档机器上，而且运行结果一致。软件兼容有向上兼容、向下兼容、向前兼容、向后兼容之分。向上(下)兼容是指在某档机器上编制的程序可不加修改地运行于比它高(低)档的机器上；向前(后)兼容是指在某型号机器上编制的程序可不加修改地运行于在它之前(后)投入市场的机器上。对于系列机，必须保证做到向后兼容，力争做到向上兼容。

软件向后兼容是系列机的根本特征，也就是说，在先生产出的机器上编制的程序可以不加修改地在后生产的新机器上运行，而后生产的新机器增强了功能并提高了速度，从而提高了性能。因此，对于那些不属于系统结构设计，而属于计算机组成和实现的内容，不管是增加、删除，还是修改，都不会影响汇编语言程序和机器语言程序在系列机上的向后兼容。但是，对于属于系统结构设计范畴内的内容，为保证软件向后兼容，只能增加其新的功能或部件，而不能删除或更改已有的功能或部件。例如，可以把 CPU 与主存之间的数据通路的宽度由 16 位扩展到 32 位，以加速主机内部信息的传送；也可以把单总线改为双总线，以减少公共总线的使用冲突。这些内容都属于计算机组成的范畴，在逻辑设计时就要对它进行设计。例如，指令的功能设计与指令字格式设计和编码属于系统设计设计的范畴，如果将原来全部采用定长操作码的指令改成操作码扩展编码，虽然可以减小平均指令字长，但是会直接导致以前编写的程序不能正确运行。

不同厂家生产的具有相同系统结构的计算机称为兼容机 (Compatible Machine)。

(3) 模拟与仿真

模拟与仿真能实现在结构不同的机器之间的机器语言程序的移植。模拟 (Simulation) 是用机器语言程序解释另一机器的机器指令来实现软件移植的。在两种机器的机器指令系统差异较大时，使用模拟方法会使程序运行速度严重下降。仿真 (Emulation) 是用微程序直接解释另一机器的机器指令来实现软件移植的。采用仿真方法可以提高被移植软件的运行速度，但在机器结构差异较大的情况下，难以实现仿真。因此，在不同系列机器之间实现软件移植时，可将模拟和仿真两种技术结合起来使用，对于频繁使用且容易仿真的这部分机器指令采用仿真的方法，以提高被移植软件的运行速度；对于较少使用、对速度要求不高且难以仿真的这部分指令及 I/O 操作采用模拟的方法。

模拟与仿真的区别是：模拟是用机器语言程序解释指令，其解释程序存储在主存中；仿真是用微程序解释指令，其解释程序存储在控制存储器中。

1.1.2 计算机系统并行性的基本概念

并行性 (Parallelism) 是指同一时刻或同一时间间隔内发生两种或两种以上性质相同或不不同的事件。只要时间上相互重叠, 就存在并行性。并行性实际上包含同时性与并发性两层含义。同时性 (Simultaneity) 是指两个或多个事件在同一时刻发生。并发性 (Concurrency) 是指两个或多个事件在同一时间间隔内发生。

1. 发展计算机系统并行性的技术途径

可以通过 3 类技术途径来提高计算机系统的并行性, 这就是时间重叠、资源重复和资源共享。

(1) 时间重叠

时间重叠 (Time Interleaving) 是在并行性概念中引入时间因素, 让多个处理过程在处理时间上错开, 轮流重叠地使用同一套硬件设备的各个部件, 提高多个处理过程的并发性。

对指令内部各操作步骤采用流水的工作方式就是时间重叠的最简单的典型例子。如图 1.2 所示, 若一条指令的解释可分为取指、分析、执行 3 个操作步骤, 并分别使用各自相应的硬件完成, 设每个操作步骤完成所需时间皆为 Δt , 那么, 第 k 条指令、第 $k+1$ 条指令和第 $k+2$ 条指令就可以在时间上重叠起来, 3 条指令彼此在时间上错开 Δt , 以流水方式被解释执行。如果这 3 条指令串行执行, 那么需要 $9\Delta t$ 。如果按流水方式执行, 如图所示, 则只需要 $5\Delta t$ 。

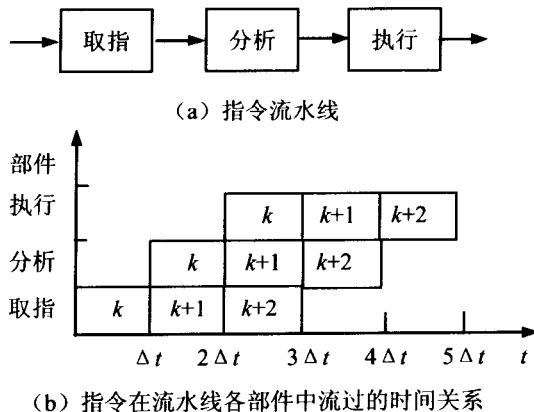


图 1.2 时间重叠的例子

由此可见, 时间重叠并没有缩短一条指令的执行时间, 但却加快了程序的执行速度。时间重叠原则上不需要增加过多的硬件设备就可以提高计算机系统的性能价格比。时间重叠的典型例子有标量流水处理机、超流水处理机和向量流水处理机等。

(2) 资源重复

资源重复 (Resource Replication) 是在并行性概念中引入空间因素, 通过重复设置硬件资源分别同时用于多个处理过程, 实现多个处理过程的同时性。