

INTRODUCTION TO COMPUTER SYSTEMS

Problem Solving and Teaching Guide

计算机系统基础

习题解答与教学指导

袁春风◎主编

唐杰 苏丰 路通 汪亮◎编著



机械工业出版社
China Machine Press



INTRODUCTION TO COMPUTER SYSTEMS
Problem Solving and Teaching Guide

计算机系统基础

习题解答与教学指导

袁春风◎主编

唐杰 苏丰 路通 汪亮◎编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

计算机系统基础习题解答与教学指导 / 袁春风主编. —北京: 机械工业出版社, 2017.1
(计算机类专业系统能力培养系列教材)

ISBN 978-7-111-55481-3

I. 计… II. 袁… III. 电子计算机—高等学校—教学参考资料 IV. TP3

中国版本图书馆 CIP 数据核字 (2016) 第 286246 号

本书作为《计算机系统基础》教材的教学指导用书, 主要对每个章节的教学目标和内容安排、主要内容提要、基本术语解释、常见问题解答等给出系统性的说明和描述, 并在此基础上提供大量的单项选择题及其参考答案、分析应用题及其分析解答, 其中涉及计算机系统概述、数据的机器级表示与处理、程序的转换及机器级表示、程序的链接、程序的执行流程、存储器层次结构、异常和中断机制以及输入/输出子系统等计算机系统的核心内容。

本书提供了系统性的教学指导和丰富的习题及其解答, 可以作为高等院校计算机专业本科或高职高专学生计算机系统基础课程的教学辅助教材, 也可以作为有关专业研究生或计算机技术人员的参考书。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 吴晋瑜

责任校对: 李秋荣

印刷: 北京瑞德印刷有限公司

版次: 2017 年 4 月第 1 版第 1 次印刷

开本: 185mm × 260mm 1/16

印张: 19.25

书号: ISBN 978-7-111-55481-3

定价: 49.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东

前 言

后 PC 时代的到来,使得原先基于 PC 而建立起来的专业教学内容已经远远不能反映现代社会对计算机专业人才的培养要求,原先计算机专业人才培养强调“程序”设计也变为更强调“系统”设计。这需要我们重新规划教学课程体系,调整教学理念和教学内容,加强学生系统能力培养,使学生能够深刻理解计算机系统整体概念,更好地掌握软/硬件协同设计和程序设计技术,从而更多地培养出满足业界需求的各类计算机专业人才。不管培养计算机系统哪个层面的人才,计算机专业教育都要重视学生“系统观”的培养。

2014 年 7 月,机械工业出版社出版了由作者编写的主教材《计算机系统基础》,重点介绍应用程序员如何利用计算机系统相关知识来编写更有效的程序。教材以高级语言程序的开发和运行过程为主线,将该过程中每个环节涉及硬件和软件的基本概念关联起来,试图使读者建立一个完整的计算机系统层次结构框架,了解计算机系统全貌和相关知识体系,初步理解计算机系统中的一个抽象层及其相互转换关系,建立高级语言程序、ISA、OS、编译器、链接器等之间的相互关联;对指令在硬件上的执行过程和指令的底层硬件执行机制有一定的认识和理解,从而增强读者在程序的调试、性能优化、移植和健壮性保证等方面的能力,并为后续的“计算机组成与设计”“操作系统”“编译技术”“计算机体系结构”等课程打下坚实基础。

为了让学生更好地理解基本概念和基本原理,能够全面、系统、准确、牢固地掌握相关知识,作者在编写主教材时采用了“基本原理和实现细节相结合,历史发展过程和现实实际情况相结合”的方针,因而使得主教材的内容涵盖面广、细节内容较多、篇幅较大,给使用该教材的老师和学生带来了一些困难。为了更好地协助主讲老师用好主教材,也为了学生能更好地理解习题内涵,作者编写了本书,对主教材中每一章的内容进行了概括总结,给出了以下 6 个方面的教学辅助内容。

(1) 教学目标和内容安排:给出相应章节的教学总体目标和基本教学要求,并较为详细地说明课堂教学内容和学生课后阅读内容的安排,以及每章的主要教学思路或教学方法。

(2) 主要内容提要:对主教材中相应章节内容进行浓缩,形成主干知识框架结构,便于学生将全书内容串接起来,形成本课程的知识框架体系。

(3) 基本术语解释:给出相关章节所涉及的基本术语的解释说明,并给出名词术语的中英文对照。

(4) 常见问题解答：提供了大量的常见问题，并给出对每个问题的解释说明。这些常见问题是作者在长期的教学过程中发现的、普遍存在于学生中的共性问题。

(5) 单项选择题：提供了相应章节内容的单项选择题及其参考答案，并对部分习题的答案进行分析解答。

(6) 分析应用题：提供了相应章节内容的分析应用题及其分析解答。

单项选择题和分析应用题这两个方面的教学辅助内容，主要是为了巩固学生所学的基本原理而设置的。通过对一些具体问题的分析，能够提高学生对基本原理的认识。

本书作为主教材的教学辅助资料，可以与主教材配套使用。同时，本书相对独立、自成体系，因此也可单独使用。本书既可作为“计算机系统基础”课程的教师参考书，也可作为学生学习“计算机系统基础”课程时的学习参考书。

本书由南京大学计算机系统基础课程教学团队编写，由袁春风教授主编。本书的编写得到了国防科技大学王志英教授、北京航空航天大学马殿富教授、西北工业大学周兴社教授、武汉大学何炎祥教授、北京大学陈向群教授等各位专家的悉心指导和热情鼓励。余子濠同学对书中部分程序进行了验证，并对全书文字内容进行了校对，提出了许多有益的修改意见。

特别感谢机械工业出版社华章公司为本书的编写工作提供了极大的支持，包括提供大量国外优秀的相关图书资料，并多次召集国内权威专家进行指导等。

由于计算机组成与系统结构相关的基础理论和技术在不断发展，新的思想、概念、技术和方法不断涌现，加之作者水平有限，在编写中难免存在不当或遗漏之处，恳请同行专家和广大读者对本书的不足之处给予指正，以便在后续的版本中予以改进。读者在主教材或本教辅教材使用过程中有任何问题或疑惑，都可与作者直接联系，联系方式：cfyuan@nju.edu.cn。

作者于南京
2016年10月

目 录

前言

第 1 章 计算机系统概述 1

- 1.1 教学目标和内容安排 1
- 1.2 主要内容提要 2
- 1.3 基本术语解释 3
- 1.4 常见问题解答 8
- 1.5 单项选择题 10
- 1.6 分析应用题 13

第 2 章 数据的机器级表示与
处理 19

- 2.1 教学目标和内容安排 19
- 2.2 主要内容提要 20
- 2.3 基本术语解释 22
- 2.4 常见问题解答 27
- 2.5 单项选择题 33
- 2.6 分析应用题 42

第 3 章 程序的转换及机器级
表示 77

- 3.1 教学目标和内容安排 77
- 3.2 主要内容提要 79
- 3.3 基本术语解释 81
- 3.4 常见问题解答 88
- 3.5 单项选择题 93
- 3.6 分析应用题 108

第 4 章 程序的链接 135

- 4.1 教学目标和内容安排 135
- 4.2 主要内容提要 136
- 4.3 基本术语解释 137
- 4.4 常见问题解答 140
- 4.5 单项选择题 142
- 4.6 分析应用题 149

第 5 章 程序的执行 159

- 5.1 教学目标和内容安排 159
- 5.2 主要内容提要 160
- 5.3 基本术语解释 163
- 5.4 常见问题解答 168
- 5.5 单项选择题 174
- 5.6 分析应用题 178

第 6 章 层次结构存储系统 186

- 6.1 教学目标和内容安排 186
- 6.2 主要内容提要 187
- 6.3 基本术语解释 191
- 6.4 常见问题解答 200
- 6.5 单项选择题 206
- 6.6 分析应用题 214

第 7 章 异常控制流 236

- 7.1 教学目标和内容安排 236
- 7.2 主要内容提要 237
- 7.3 基本术语解释 241

7.4 常见问题解答	243	8.2 主要内容提要	259
7.5 单项选择题	248	8.3 基本术语解释	262
7.6 分析应用题	252	8.4 常见问题解答	269
第 8 章 I/O 操作的实现	257	8.5 单项选择题	276
8.1 教学目标和内容安排	257	8.6 分析应用题	285

计算机系统概述

1.1 教学目标和内容安排

主要教学目标：概要了解整个计算机系统的全貌以及程序开发和执行的大致过程，并使学生掌握如何简单评价计算机系统的性能。

基本学习要求：

1. 简单了解计算机的发展历程、计算机分代标志以及摩尔定律的内容。
2. 了解计算机系统的基本功能以及实现基本功能所对应的部件。
3. 了解计算机系统中硬件和软件的基本概念及其相互关系。
4. 了解冯·诺依曼结构计算机的特点，以及计算机硬件的基本组成和各部件的功能。
5. 了解计算机软件的分类，以及各类系统软件和应用软件的功能。
6. 了解程序开发和执行过程，理解各种语言处理程序（翻译程序、编译程序、汇编程序）的概念。
7. 理解计算机系统的层次化结构。
8. 了解各类计算机用户在计算机系统中所处位置，以及本课程在计算机系统中所处位置。
9. 了解如何对计算机的性能进行测量和评价。
10. 了解有哪些因素会影响计算机的性能。

本章涉及的内容是计算机学科最基本的概念和知识，虽然没有特别难懂的部分，但是对于低年级学生来说，有些概念还是比较抽象和难以理解的，需要他们在对后面章节的不断学习过程中，去深化对它们的理解并熟练运用。遇到这些内容时，可以告诉学生相关内容在后面具体哪个章节中会详细介绍。

有关计算机发展历程的部分内容中，出现了许多学生难以理解的专业术语，因此，这部分内容可以跳过不讲，但是，对于冯·诺依曼计算机结构的特点、“存储程序”工作方式、“兼容性”概念、摩尔定律等内容，要求学生能够掌握和理解。

对于计算机层次化概念，它和计算机系统组成的内容是相互联系的，因为不同计算机用户眼中的计算机系统是不一样的。可以从最终用户感觉到的计算机硬件和软件的形态开始，逐步延伸到系统管理员、应用程序员、系统程序员以及系统架构师眼中的硬件和软件形态。这两部分内容对学生建立整个计算机系统全貌以及了解本课程在计算机系统的位置是非常重要的。

1.2 主要内容提要

1. 冯·诺依曼计算机结构的主要特点

冯·诺依曼计算机结构的主要特点包括：①计算机由运算器、控制器、存储器、输入设备和输出设备五大部分组成。②指令和数据用二进制表示，两者形式上没有差别。③指令和数据存放在存储器中，按地址访问。④指令由操作码和地址码组成，操作码指定操作性质，地址码指定操作数地址。⑤采用“存储程序”方式进行工作。

2. 计算机硬件的基本组成和功能

运算器用来进行各种算术逻辑运算；控制器用来对指令进行译码并送出操作控制信号；存储器用来存放指令和数据；输入/输出设备用来实现计算机和用户之间的信息交换。

3. 计算机系统的层次结构

计算机系统分软件和硬件两部分，软件和硬件的界面是指令集体系结构（ISA）。计算机系统从高到低粗分为应用软件、系统软件和硬件三个层次；不同计算机用户工作在不同的层次，从高到低细分为应用程序级（最终用户）、高级语言虚拟机级（高级语言程序员或应用程序员）、汇编语言虚拟机级（汇编语言程序员）、操作系统虚拟机级（系统管理员）、机器语言机器级（机器语言程序员）。

4. 硬件和软件的相互关系

计算机硬件和软件两者相辅相成，缺一不可。两者都用来实现逻辑功能，同一功能可用硬件实现，也可用软件实现。

5. 程序开发和执行过程

首先用某种语言（高级语言或低级语言）编制源程序；然后用语言处理程序（编译程序或汇编程序）将源程序翻译成机器语言目标程序。通过某种方式启动目标程序（可执行代码）执行时，操作系统将指令和数据装入内存，然后从第一条指令开始执行。每条指令的执行过程为：取指令、指令译码、取操作数、运算、送结果、PC指向下一条指令。可执行程序由若干指令组成，CPU周而复始地执行一条一条指令，直到程序所含指令全部执行完。

6. 基本性能指标和性能参数

计算机系统的基本性能指标包括响应时间和吞吐率。处理器的基本性能参数包括时钟周期（或主频）、CPI、MIPS、MFLOPS（或GFLOPS、TFLOPS、PFLOPS等）。

7. 性能测量

一般把程序的响应时间划分成CPU时间和等待时间，CPU时间又分成用户CPU时间和操作系统CPU时间。因为操作系统对自身所用的时间进行测量时不十分准确，所以，对CPU性能的测量一般是通过测量程序运行时的用户CPU时间来进行的。

8. 各种性能参数之间的关系

时钟周期和时钟频率互为倒数。CPU执行时间可用“CPU时钟周期数 × 时钟周期”来计算。CPU时钟周期数可用“程序所含指令条数 × 程序的CPI”来计算。

9. 性能评价程序

可采用一组性能评价程序（即基准程序）对机器性能进行评测。有些机器制造商会针对

基准程序中频繁出现的语句采用专门的编译器进行优化，使基准程序的运行效率大幅提高，因此有时用基准程序得到的评测结果不能说明问题。

1.3 基本术语解释

系列机 (family machine)

系列机是指一个厂家生产的具有相同系统结构、不同组成和实现的一系列不同型号的机器。它应在指令系统、数据格式、字符编码、中断系统、输入/输出控制方式等方面保持一致，从而保证软件的兼容性。

兼容性 (compatibility)

兼容是一个广泛的概念，主要表示一种“互换”特性，包括软件兼容、硬件兼容等。

软件兼容 (software compatibility)

软件兼容是指在某种机型上开发的软件可以不加修改地在另外的机型上正确运行。一般在同一系列机内的软件是兼容的，有向上兼容和向下兼容两种形式。向上兼容是指高档机型上的程序能在低档机型上运行，向下兼容是指低档机型上的程序能在高档机型上运行，一般系列机满足向下兼容性。因为系列机中高档机的指令系统包含了低档机中的所有指令。

硬件兼容 (hardware compatibility)

硬件兼容也就是设备或部件兼容，是指设备或部件可以不加改动地用于多种计算机。这要求设备或部件符合某种标准化设计。

小规模集成电路 (Small Scale Integrated circuit, SSI)

集成度小于 100 的集成电路。

中规模集成电路 (Medium Scale Integrated circuit, MSI)

集成度在 100 到 1000 的集成电路。

大规模集成电路 (Large Scale Integrated circuit, LSI)

集成度在 1000 到 10 万的集成电路。

超大规模集成电路 (Very Large Scale Integrated circuit, VLSI)

集成度在 10 万到 1000 万的集成电路。

极大规模集成电路 (Ultra Large Scale Integrated circuit, ULSI)

集成度在 1000 万以上的集成电路。

摩尔定律 (Moore's law)

1965 年，摩尔预测：以后每年将缩小硅片中形成晶体管电路的细线尺寸的 10%，芯片制造商能够每 3 年发布新一代的芯片，其晶体管数为上一代的 4 倍。后来摩尔定律被表述成：由于集成电路技术的不断改进，每 18 ~ 24 个月，集成电路芯片上集成的晶体管数将翻一番，速度将提高一倍，而价格将降低一半。

通用电子计算机 (general-purpose electronic computer)

通用电子计算机是和专用电子计算机对应的，专用电子计算机只能专门用于某种应用，而通用电子计算机从定义上来说可以解决任何问题，只要这个问题可以用程序来表示。通用

电子计算机也被称为完备的图灵机。

算术逻辑单元 (Arithmetic Logic Unit, 简称 ALU)

对数据进行算术运算和逻辑运算处理的部件。

数据通路 (data path)

数据通路指的是指令在执行过程中数据所经过的部件以及部件之间的连接线路, 主要由 ALU 和一组寄存器、存储器、总线等组成。国内许多教科书中提到的运算器即是运算数据通路。

控制器 (control unit)

控制器也称为控制单元或控制部件。其作用是对指令进行译码, 将译码结果和状态 / 标志信号、时序信号等进行组合, 产生各种操作控制信号。这些操作控制信号被送到 CPU 内部或通过总线送到主存或 I/O 模块。送到 CPU 内部的控制信号用于控制 CPU 内部数据通路的执行, 送到主存或 I/O 模块的控制信号用于控制 CPU 和主存或 CPU 和 I/O 模块之间的信息交换。控制单元是整个 CPU 的指挥控制中心, 通过规定各部件在何时做什么动作来控制数据的流动, 完成指令的执行。

中央处理器 (Central Processing Unit, CPU)

中央处理器是计算机中最重要的部件之一, 主要由运算器和控制器组成。其内部结构归纳起来可以分为控制部件、运算部件和存储部件三大部分, 它们相互协调, 共同完成对指令的执行。

存储器 (Memory, Storage)

存储器用于存储程序和数据, 分为内存储器 (简称内存) 和外存储器 (简称外存)。内存存取速度快, 但容量小、价格贵; 外存容量大、价格低, 但存取速度慢。

内存 (memory)

从字面上来说, 内存是内部存储器, 应该包括主存 (Main Memory, 简称 MM) 和高速缓存 (cache), 但是, 因为早期计算机中没有高速缓存, 因而传统意义上的内存就是主存, 所以, 目前也并不区分内存和主存。内存位于 CPU 之外, 用来存放已被启动执行的程序及所用的数据, 包括 ROM 芯片和 RAM 芯片组成的相应 ROM 存储区和 RAM 存储区两部分。

外存 (storage)

外存储器主要有磁盘存储器、磁带存储器和光盘存储器等。磁盘是最常用的外存储器, 通常它分为软盘和硬盘两类。容量极大、价格便宜的磁带机和光盘组等称为海量存储器, 常用作数据备份, 也称为辅存 (Accessorial Memory, 简称 AM) 或二级存储器 (Secondary Memory, 简称 SM)。

系统软件 (system software)

系统软件是介于计算机硬件与应用程序之间的各种软件, 它与具体应用关系不大。系统软件包括操作系统 (如 Windows)、语言处理系统 (如 C 语言编译器)、数据库管理系统 (如 Oracle) 和各类实用程序 (如磁盘碎片整理程序、备份程序)。

应用软件 (application software)

应用软件是指针对使用者的某种应用目的所编写的软件, 例如办公自动化软件、互联网

应用软件、多媒体处理软件、股票分析软件、游戏软件、管理信息系统等。

操作系统 (Operating System, 简称 OS)

操作系统是计算机系统中负责支撑应用程序运行环境以及用户操作环境的系统软件,其目的是使计算机系统所有资源最大限度地发挥作用,并为用户提供方便的、有效的、友善的服务界面。操作系统是一个庞大的管理控制程序,大致包括五个方面的管理功能:进程与处理器管理、作业管理、存储管理、设备管理和文件管理。目前比较流行的操作系统主要有两个家族:类 UNIX 家族和微软的 Windows 家族。

最终用户 (end user)

使用应用程序完成特定任务的计算机用户称为最终用户。大多数计算机使用者都属于最终用户。例如使用炒股软件的股民、玩计算机游戏的人、进行会计电算化处理的财会人员等。

系统管理员 (system administrator)

指利用操作系统提供的功能对系统进行配置、管理和维护以建立高效合理的系统环境供计算机用户使用的操作人员。其职责主要包括安装、配置和维护系统的硬件和软件,建立和管理用户账户,升级软件,备份和恢复业务系统及数据等。

应用程序员 (application programmer)

指使用高级编程语言编制应用软件的程序员。

系统程序员 (system programmer)

指设计和开发系统软件的程序员,如开发操作系统、编译器、数据库管理系统等系统软件的程序员。

高级语言 (high-level programming language)

高级语言也称为高级编程语言或算法语言,是面向问题和算法的描述语言。用这种语言编写程序时,程序员不必了解实际机器的结构和指令系统等细节,而是通过一种比较自然的、直接的方式来描述问题和算法。

汇编语言 (assembly language)

汇编语言是一种面向实际机器结构的低级语言,是机器语言的符号表示,与机器语言一一对应。因此,汇编语言程序员必须对机器的结构和指令系统等细节非常清楚。

机器语言 (machine language)

机器语言是指直接用二进制代码(指令)表示的语言。用户必须用二进制代码来编写机器语言程序。因此,机器语言程序员必须对机器的结构和指令系统等细节非常清楚。

指令集 (instruction set)

指令集是一台计算机能够执行的所有机器指令的集合。指令按功能可以分为运算指令、移位指令、传送指令、串指令、程序控制指令等类型。

指令集体系结构 (Instruction Set Architecture, ISA)

指令集体系结构是计算机硬件与系统软件之间的接口,指机器语言程序员或操作系统、编译器、解释器设计人员所看到的计算机功能特性和概念性结构。其核心部分是指令系统,同时还包含数据类型和数据格式定义、寄存器组织、I/O 空间的编址和数据传输方式、中断

结构、计算机状态的定义和切换、存储保护等。ISA 设计得好坏直接决定了计算机的性能和成本。

透明性 (transparency)

由于计算机系统采用了层次化结构进行设计和组织, 因此面向不同的硬件或软件层面进行工作的人员或用户所“看到”的计算机是不一样的。也就是说, 计算机组织方式或系统结构中的一部分对某些用户而言是“看不到”的或称为“透明”的。例如, 对于高级语言程序员来说, 指令格式、数据格式、机器结构、指令和数据的存取方式等都是透明的; 而对于机器语言程序员和汇编语言程序员来说, 指令格式、机器结构、数据格式等则不是透明的。

源程序 (source program)

编译程序、解释程序和汇编程序统称为语言处理程序。各种语言处理程序处理的对象称为源程序, 用高级 (算法) 语言或汇编语言编写, 如 C 语言源程序、Java 语言源程序、汇编语言源程序等。

目标程序 (object program)

编译程序和汇编程序对源程序进行翻译得到的结果程序称为目标程序或目标代码 (object code)。

编译程序 (compiler)

编译程序也称为编译器, 是用来将高级语言源程序翻译成汇编语言或机器语言目标代码的程序。

解释程序 (interpreter)

解释程序将源程序的一条语句翻译成对应的机器语言目标代码, 并立即执行, 然后翻译下一条源程序语句并执行, 直至所有源程序中的语句全部被翻译并执行完。因此, 解释程序并不输出目标程序, 而是直接输出源程序的执行结果。

汇编程序 (assembler)

汇编程序也是一种语言翻译程序, 它把用汇编语言写的源程序翻译为机器语言目标程序。汇编程序和汇编语言是两个不同的概念, 不能混为一谈。

响应时间 (response time)

响应时间也称为执行时间 (execution time) 或等待时间 (latency time), 是指从作业提交开始到作业完成的时间。一般一个程序的响应时间除了 CPU 执行程序包含的指令执行时间外, 还包括等待 I/O 的时间、系统运行其他用户程序所用的时间以及操作系统运行的时间等。

吞吐量 (throughput)

在有些场合下, 吞吐量也称为带宽 (bandwidth), 是指在一定的时间内所完成的工作量。

CPU 执行时间 (CPU execution time)

CPU 执行时间也称为 CPU 时间 (CPU time), 是指在程序运行过程中, CPU 真正用于程序执行的时间。它不包括因为等待 I/O 操作完成所需的时间, 也不包括 CPU 执行其他程序所需的时间。对用户来说, 能直接感觉到的只能是响应时间, 而不可能是 CPU 执行时间。CPU 执行时间被进一步分为两部分: 一部分是用来运行用户程序代码的时间, 称为用户 CPU 时

间 (user CPU time); 另一部分是为了执行用户程序而不得不运行的一些操作系统代码所花费的时间, 称为系统 CPU 时间 (system CPU time)。

系统性能 (system performance)

通常用没有任何其他负载的情况下程序的响应时间来表示系统性能。

CPU 性能 (CPU performance)

通常以用户 CPU 时间来表示 CPU 性能。

时钟周期 (clock cycle, tick, clock tick, clock)

所有计算机执行指令的过程都是分成若干步骤和相应的动作来完成的, 每一步动作都要有相应的控制信号进行控制, 这些控制信号何时发出、作用时间多长, 都要有相应的定时信号进行同步。因此, CPU 必须能够获得用于同步的时钟定时信号, 也就是 CPU 的主脉冲信号, 其宽度称为时钟周期。

时钟频率 (clock rate, 主频)

CPU 的主频就是 CPU 的主脉冲时钟信号的频率, 是 CPU 时钟周期的倒数。

CPI (Cycle Per Instruction)

CPI 是衡量 CPU 性能的一种基本参数, 它表示执行一条指令所需的平均时钟周期个数。

基准测试程序 (benchmark)

基准测试程序是专门用来进行性能评价的一组程序, 这些程序能够很好地反映机器在运行实际负载时的性能。可以在不同机器上运行相同的基准测试程序来比较不同机器的运行时间, 从而比较其性能。

SPEC 基准测试程序集 (SPEC benchmark)

系统性能评价标准 SPEC (System Performance Evaluation Cooperative) 是一个应用广泛而且全面的基准程序集, 用于对 CPU 性能进行评测, 分为整数程序集 SPECint 和浮点程序集 SPECfp 等。

SPEC 比值 (SPEC ratio)

SPEC 比值是将测试程序在 Sun SPARCstation 上运行时的执行时间除以该程序在测试机器上的执行时间所得到的比值。比值越大, 机器的性能越好。

阿姆代尔定律 (Amdahl's law)

阿姆代尔定律的主要含义是指系统优化某部件所获得的系统性能的改善程度, 取决于该部件被使用的频率或所占总执行时间的比例。该定律很好地诠释了改善“系统瓶颈”性能的重要性。

MIPS (Million Instructions Per Second)

MIPS 用来衡量单位时间内执行指令的条数, 具体是指每秒执行多少百万条指令。

加法指令执行速度 (add instruction execution speed)

最早用来衡量计算机性能的指标是完成单个运算 (如加法运算) 指令所需要的时间。当时大多数指令的执行时间是相同的, 并且加法指令能反映乘、除等运算, 而其他指令的时间也大体与加法指令相当, 故加法指令的速度有一定的代表性。加法指令执行速度的计量单位为 KIPS (每秒千条指令) 和 MIPS (每秒百万条指令)。

平均指令执行时间 (average instruction execution time)

以平均指令执行时间衡量性能的方法也称为等效指令速度法或 Gibson 混合法。随着计算机的发展,不同指令所需的执行时间差别越来越大,人们就根据等效指令速度法通过统计各类指令在程序中所占比例进行折算。设某类指令 i 在程序中所占比例为 w_i , 执行时间为 t_i , 则等效指令的执行时间为: $T=w_1 \times t_1+w_2 \times t_2+\cdots+w_n \times t_n$ (n 为指令种类数)。如果指令执行时间用时钟周期数来衡量,那么平均指令执行时间就是平均 CPI。对平均指令执行时间求倒数能够得到平均 MIPS 值。

峰值 MIPS (peak MIPS)

选取一组指令组合,使得平均 CPI 最小,由此得到的 MIPS 就是峰值 MIPS。有些制造商经常将峰值 MIPS 直接当作 MIPS,因此,实际上的性能要比标称的性能差。

相对 MIPS (relative MIPS)

根据一种公认的参考机型来定义相应的 MIPS 值,其值的含义是相对于参考机型 MIPS 的多少倍。

MFLOPS (Million floating-point operations per second)

MFLOPS 是计算机浮点数运算速度的一种计量单位,表示每秒执行多少百万次浮点数运算。它是基于所完成的单精度浮点数的操作次数而不是指令数来衡量的。类似的速度单位还有 GFLOPS($1G=10^9$)、TFLOPS($1T=10^{12}$)、PFLOPS($1P=10^{15}$)、EFLOPS($1E=10^{18}$) 等。

1.4 常见问题解答

1. 计算机系统就是硬件系统吗?

答:这样说是不完整的。一个完整的计算机系统应该包括硬件系统和软件系统两部分。硬件系统包括运算器、控制器、存储器、输入设备和输出设备五大基本部件。软件系统分为系统软件和应用软件两大类。系统软件包括操作系统、计算机语言处理程序(各种程序翻译软件,包括编译程序、解释程序、汇编程序)、服务性程序、数据库管理系统和网络软件等;应用软件包括各种特定领域的处理程序。计算机系统硬件和软件是相辅相成的,缺一不可。软件是计算机系统的灵魂,没有软件的硬件不能被用户使用,犹如一堆废铁。

2. 同一个功能既可以由软件完成,也可以由硬件完成吗?

答:软件和硬件是两种完全不同的形态,硬件是实体,是物质基础;软件是一种信息,看不见、摸不到。但是它们都可以用来实现逻辑功能,所以在逻辑功能上,软件和硬件是等价的。因此,在计算机系统中,许多功能既可以直接由硬件实现,也可以在硬件的配合下由软件来实现。例如,乘法运算既可以用专门的乘法器硬件实现,即机器提供专门的一条乘法指令;也可以用乘法子程序来实现,即不提供乘法指令,而由加法指令和移位指令等组成的一个指令序列来完成乘法运算。

3. 解释程序和编译程序有什么差别?

答:编译程序和解释程序是两种不同的翻译程序。二者的不同在于编译程序将高级语言源程序全部翻译成目标程序,每次执行程序时,只要执行目标程序,因此,只要源程序不

变,就无需重新翻译;解释程序是将源程序的一条语句,翻译成对应的机器目标代码并立即执行,然后翻译下一条语句并执行,直至所有源程序中的语句全部被翻译并执行完。所以解释程序的执行过程是翻译一句,执行一句。解释的结果是源程序执行的结果,而不会生成目标程序。

4. 要计算机做的任何工作都要先编写成程序才能完成吗?

答:是的,要计算机完成任何事情,都必须先编制程序。程序是由指令构成的。不管是用哪种语言编写的程序,最终都要翻译成机器语言程序才能让机器理解。机器语言程序是由一条一条指令组成的程序。CPU的主要功能就是周而复始地执行指令,因此,要计算机完成的所有功能都是通过执行一条一条指令来实现的,也就是由一个程序来完成。有时我们说某个特定的功能是由硬件实现的,但并不是说不要编写程序,如乘法功能可由乘法器这个硬件实现,但要启动这个硬件(乘法器)工作,必须先执行程序中的乘法指令。

5. 指令和数据在形式上没有差别,且都存于存储器中,计算机如何区分它们?

答:指令和数据在计算机内部都是用二进制表示的,因而都是0、1序列,在形式上没有差别。在指令和数据取到CPU之前,它们都存放在存储器中,CPU必须能够区分读出的是指令还是数据。如果是指令,CPU会把指令的操作码送到指令译码器进行译码,而把指令的地址码送到相应的地方进行处理;如果是数据,则送到寄存器或运算器。那么,CPU如何识别读出的是指令还是数据呢?实际上,CPU并不是把信息从主存读出后靠某种判断方法来识别信息是数据还是指令的,而是在读出之前就知道将要读的信息是数据还是指令了。执行指令的过程分为取指令、指令译码、取操作数、运算、送结果等。所以,在取指令阶段,总是根据程序计数器(PC)的值去取指令,所以取来的一定是指令;取操作数阶段取来的一定是数据。

6. 衡量计算机系统性能的主要指标是什么?

答:计算机性能的好坏主要体现在速度方面,而衡量速度快慢主要有两个指标:响应时间和吞吐率。响应时间是指从作业提交开始到作业完成所花的时间。一般一个程序的响应时间除了CPU执行该程序包含的指令所用的时间外,还包括等待输入/输出操作所需时间和操作系统运行这个程序所花的时间开销等。吞吐率是指单位时间内完成的工作量。

7. CPU的时钟频率越高,机器的速度就越快,对吗?

答:在其他因素不变的情况下,CPU的时钟频率越高,机器的速度肯定越快。但是,程序执行的速度除了与CPU的速度有关外,还与存储器、I/O等模块的存取速度、总线的传输速度、cache的设计策略等有很大关系。因此,机器的速度不是只由CPU的时钟频率决定。

8. CPI的含义是什么?执行时间(响应时间)与CPI是什么关系?

答:CPI是指每条指令执行时所用的时钟周期数。通常,一条特定指令的CPI是一个确定的值,而某个计算机或程序的CPI则是一个平均值。一个程序的执行时间取决于该程序所包含的指令数、CPI和时钟周期。在指令数和时钟周期一定的情况下,CPI越大,执行时间越长。

9. 用户真正感觉到的程序执行时间是否就是执行一个程序中所有指令所用的时间?

答:不是。执行一个程序中所有指令所用的时间实际上比用户真正感觉到的程序执行时

间要短。因为在一个程序执行过程中，可能还会执行操作系统代码或其他用户程序，并且有时还可能等待 I/O 操作。比如，在 UNIX 操作系统中，假定用 time 命令测试某程序执行时间的结果为：

80.3u 10.1s 2:02 74%

则说明该程序所有指令的执行时间（即用户 CPU 时间）只有 80.3s，执行操作系统代码所用时间（即系统 CPU 时间）是 10.1s，用户感觉到的总响应时间为 $2 \times 60 + 2 = 122\text{s}$ ，其中的 $(80.3 + 10.1) / 122 = 74\%$ 是 CPU 时间，剩下 26% 的时间（即 30 多秒）用来等待 I/O 操作或运行其他用户程序。

10. 计算机的 MIPS 数越大，说明性能越好，对吗？

答：不对。MIPS 数反映的是机器执行定点指令的速度。但是，不同机器的指令集不同，指令的功能也不同，也许一个机器上一条指令的功能在另外一个机器上要用多条指令来完成，这样，同样的指令条数所完成的功能可能完全不同。另外，不同机器的 CPI 和时钟周期也不同，因而一条指令所用的时间也不同。所以，用 MIPS 数来对不同的机器进行性能比较是不太客观的。

11. 基准测试程序执行得越快，说明机器的性能越好，对吗？

答：不对。一般情况下，基准测试程序能够反映机器性能的好坏。但是，如果制造商针对基准测试程序中频繁出现的语句采用专门的编译器，使基准测试程序的运行效率大幅提高，那么基准测试程序测试的结果就不能说明问题。

1.5 单项选择题

- 以下有关对摩尔定律的描述中，错误的是（ ）。
 - 每 18 个月，集成电路芯片上集成的晶体管数将翻一番
 - 每 18 个月，集成电路芯片的速度将提高一倍
 - 每 18 个月，集成电路芯片的价格将降低一半
 - 集成电路技术一直会遵循摩尔定律发展下去
- 从计算机的主要元器件来看，计算机发展所经历的过程为（ ）。
 - 晶体管、电子管、SSI、MSI、LSI、ULSI、VLSI
 - 电子管、晶体管、SSI、MSI、LSI、VLSI、ULSI
 - 电子管、晶体管、LSI、MSI、SSI、VLSI、ULSI
 - 晶体管、电子管、MSI、SSI、LSI、ULSI、VLSI
- 一个完整的计算机系统包括硬件和软件。软件又分为（ ）。
 - 操作系统和语言处理程序
 - 系统软件和应用软件
 - 操作系统和高级语言
 - 低级语言程序和高级语言程序
- 以下给出的软件中，属于应用软件的是（ ）。
 - 汇编程序
 - 编译程序
 - 操作系统
 - 文字处理程序