面向工程教育的本科计算机类专业系列教材

Principles of
Computer Hardware
System Design

计算机硬件系统设计原理

刘子良 等 编著

高等教育出版社

面向工程教育的本科计算机类专业系列教材

Principles of
Computer Hardware
System Design

计算机硬件系统设计原理

刘子良　徐高潮　齐红
申铉京　魏晓辉　刘苗　编著

高等教育出版社·北京

内容提要

本书根据教育部提出的“基础学科拔尖学生培养试验计划（珠峰计划）”和“卓越工程师培养计划”编写而成。它将以往的“微型计算机技术及应用”“计算机组成原理”“计算机系统结构”“并行处理”和“精简指令集计算机（RISC）”等课程，从计算机系统硬件设计的角度出发，沿着单机指令串行执行→重叠→流水线、线性流水线→非线性流水线、指令的有序执行→乱序执行、单机→多核→多机系统、冯·诺依曼体系→非冯·诺依曼体系的脉络，有机地结合在一起，科学地、系统地、定量地揭示当代计算机的本质特征，运用到当前计算机系统设计问题中去。

全书共10章，包括计算机设计史简介、数学语言与计算机部件间的映像关系、运算方法采用与运算器设计、指令集结构设计、存储器系统设计、控制器设计、流水线控制技术、输入/输出系统设计、多处理机、其他设计思想简介。

本书可作为高等学校计算机科学与技术专业本科生的教材，也可作为教师的教学参考书。

图书在版编目（CIP）数据

计算机硬件系统设计原理/刘子良等编著. --北京：高等教育出版社，2016.2

ISBN 978-7-04-044227-4

Ⅰ. ①计… Ⅱ. ①刘… Ⅲ. ①硬件-系统设计-高等学校-教材 Ⅳ. ①TP303

中国版本图书馆CIP数据核字(2015)第275212号

策划编辑 倪文慧　责任编辑 倪文慧　封面设计 赵 阳　版式设计 童 丹
插图绘制 杜晓丹　责任校对 李大鹏　责任印制 韩 刚

出版发行 高等教育出版社
社　　址 北京市西城区德外大街4号
邮政编码 100120
印　　刷 涿州市星河印刷有限公司
开　　本 850mm×1168mm 1/16
印　　张 36
字　　数 820千字
购书热线 010-58581118
咨询电话 400-810-0598
网　　址 http://www.hep.edu.cn
http://www.hep.com.cn
网上订购 http://www.landraco.com
http://www.landraco.com.cn
版　　次 2016年2月第1版
印　　次 2016年2月第1次印刷
定　　价 53.00元

物料号 44227-00

前　言

这是一本根据教育部“基础学科拔尖学生培养试验计划（珠峰计划）”和“卓越工程师培养计划”编写的教材。作为计算机学科的“拔尖学生”和“卓越工程师”，了解和掌握计算机的设计思想十分必要。一台计算机的功能总是有限的，软科学的研发与多种软件的配置无疑是为了扩展机器的功能，这种扩展是基于计算机的。换句话说，计算机是“根”，软件则建立于这个根基之上。想想看，如果不清楚这个根，或者是对这个“根”缺乏了解，怎么会让它长出绿叶并开出美丽的花朵呢？

计算机设计是一个极富挑战性和令人心动的领域。在计算机技术发展的里程中，有成功、有教训、有借鉴。计算机设计思想的论述虽然偏重硬件层次，但主要还是着重阐述硬件与软件的边界。因此，编译器、操作系统及数据库的设计者和其他软件设计人员都需要了解本书讲解的基本原理。软件设计者对系统中硬件技术的理解程度决定了未来软件系统的性能；同样，硬件设计者对系统中软件技术的理解也决定了硬件设计中的逻辑思维和着眼点。

计算机设计并不是应用最佳设计算法的简单过程，许多决策取决于大量试探性的判断和经验，以及硬件和软件之间的交互与权衡。因此，计算机设计思想越来越成为一种博弈，它要以改变一个领域的结构和功能补偿另一个领域性能的失配。

尽管计算机领域富有多样性，并且仍在改变，但是计算机的某些基本概念、基本技术和基本方法却始终在起作用，尤其是在不同指令集的差距不断缩小的今天。这就需要在处理那些看上去很明白其原理却鲜为人知的问题时进行深层次的挖掘，掌握其核心思想。这样不仅使读者了解到计算机设计中的一些规律，也为从事其他领域的研发夯实基础。

本书整合传统的“微型计算机技术及应用”“计算机组成原理”“计算机系统结构”“并行处理”和“精简指令集计算机（RISC）”等课程，从计算机系统硬件设计的角度出发，沿着单机指令串行执行→重叠→流水线、线性流水线→非线性流水线、指令的有序执行→乱序执行、单机→多核→多机系统、冯·诺依曼体系→非冯·诺依曼体系的脉络，有机地结合在一起，科学地、系统地、定量地揭示当代计算机的本质特征，运用到当前计算机系统设计问题中去。

本书共10章，章节组织及概要如下：

第1章计算机设计史简介。本章简单回顾从公元前50世纪有了计数的应用到电子计算机的前身——继电器计算机的漫长历程中，中外先驱者为实现自动计算所做的贡献；以冯·诺依曼机、系列机与兼容机技术、层次技术、虚拟机技术，以及从单机到多机、从单核到多核，展现计算机的设计历程。

第2章数学语言与计算机部件间的映像关系。本章目的在于表明数学词义（如整数、有符号数、实数等）的描述与计算机各部件之间的映像。显然，这个问题不解决，一个数学问

题用计算机来求解就是一句空话。本章针对算逻运算中所涉及的各种数据类型，以及在计算机中的表述与存放展开讨论。

第3章运算方法采用与运算器设计。通过运算方法和运算器设计的讨论，旨在阐述算法不同，实现方案也不同。算法的改造不是凭空而来的，而是在设计方案不断改造中形成的。机器算法与数学算法不同，它不是一个单纯的求解过程。一个算法的好与坏必须与方便机器实现结合起来，否则就不是一个好的算法。为此，书中进行了详细的讨论。

第4章指令集结构设计。指令集是硬件和软件的接口，或者说，指令集定义了硬件和编译器的接口，它是硬件和编译器都能理解的语言，这一认识对设计者尤为重要。为此，书中阐述了什么是一个好的指令系统，并从指令集的语义学和语法学的观点出发展示一个好的指令系统。

第5章存储器系统设计。如何搭建一个梦寐以求的无限容量的快速存储器是本章讨论的要点。从理想的角度，存储器应该是高速度、大容量、低成本。但现实中不可能同时满足这些要求，解决这个难题的方法是采用存储器层次结构，而不是依赖于单个部件或技术。本章就“层次”中的主存、Cache和虚拟存储器技术以及Cache、虚拟存储器和TLB之间在实、虚模式下的交互技术做进一步讨论。

第6章控制器设计。就目前控制器的设计方法而言，要么采用硬连线逻辑，要么采用微程序设计，要么是两者的结合。本章系统地阐述了硬连线控制器和微程序控制器的设计方法、步骤和实施细节。

第7章流水线控制技术。提高单机执行速度的重要一环是采用流水线技术。流水线是组织指令并发执行的一个非常有效的方法，但是，其中也有非流水控制单元中未出现的问题，即流水线的相关。这个问题处理不好，流水线中就会产生竞争冒险，使流水线断流。本章对设计中产生的各种相关原因加以分析，分门别类地进行解决，并对非线性流水线、动态流水线以及超标量流水线技术展开深入讨论。

第8章输入/输出（I/O）系统设计。与计算机设计的其他领域相比，I/O系统是一个更复杂、更难解决的领域。例如，怎样通过总线“共享”和“多能性”定义一种模式，将一个新的设备很容易地加到总线上；CPU授权I/O操作方式的考虑和实施；I/O和操作系统的接口；存储系统虚/实模式下的DMA操作以及I/O处理性能的评价等。随着物联网的发展，I/O将会扮演越来越重要的角色。

第9章多处理机。由于计算机单机技术的发展受到诸如光速和物理尺寸的限制，使得高性能计算机走上了多机并行的道路。本章首先从历史观审视地评价SIMD计算机，告诉读者MIMD计算机无疑是多处理机的最佳选择。随着多核处理器的出现，它的地位越来越高。为此，书中对多处理器以及多核处理机做了较深入的论述。

第10章其他设计思想简介。传统的冯·诺依曼计算机无论是单处理机还是多处理机，指令都是按照它们在程序中出现的次序执行的。如果有一个例外的话，那就是基于数据相关的指令重排序。为了大幅度地提高计算机的并行能力，20世纪70年代以来提出了一些同冯·诺依曼机截然不同的设计思想，如数据流计算、归约机、脉动阵列、神经网络以及模式匹配驱动，书中对此进行简要论述，最后给出计算机设计性能和定量准则。

本书与“计算机硬件系统设计原理”课程配套使用，已在吉林大学计算机学院“唐班”试用两年。教材适用的授课学时为112学时，使用者须具备数字逻辑相关知识，考虑到与其他相关课程的衔接，可安排在两个学期讲授。

本书融汇作者近40年来的教学经验和科研成果，力求做到脉络清晰、简明易懂，可作为高校计算机科学与技术专业本科生的教材，也可作为教师的教学参考书。

由于作者水平有限，书中难免有错误和不妥之处，望读者多加指正。

刘子良

2015年8月于吉林大学

目　录

第1章　计算机设计史简介

本章介绍计算机的设计历程，首先谈自动计算的发展简史，接着讨论从 ENIAC 到冯·诺依曼体系，从中找到计算机的设计坐标系。20 世纪 60 年代初，IBM 公司提出了生产系列机思想，不仅使硬件设计得到长足发展，也为软件工作者营造了一个稳定的编程环境。人们通过层次解决了开发技术。虚拟机概念为应用软件开发人员提供了一条捷径，并使设计者充分利用硬件和软件逻辑上的等价性，而不是直接用硬件去造一台现代计算机。

随着计算机速度的提高，单机技术由于受到诸如光速和物理尺寸等本质限制，必须利用资源重复的办法，走多机并行的道路。

1.1　自动计算发展史

人类用数字要比用文字早得多，早在公元前 50 世纪就有了计数的应用，仅次于人类对火的利用。

起初，人类的计数工具是石块、贝壳、木棒等，有人推说这是算盘的雏形。而有关算盘的历史，还得追溯到我国春秋战国时期（公元前 770 年—公元前 221 年）的算筹，它的样子很像长短不同的筷子。人们利用长短不同的小棍摆成不同的行列来进行运算，称之为“筹算法”。这些在《九章算术》中有详细记载。

唐朝末叶，我国民间开始出现算盘，到南宋时期有了算盘口诀的流传。15 世纪中叶，珠算的语言开始形成，明清两代传到日本。这种算盘，每一数位有两组算盘珠（5 个算盘珠表示 0～5，2 个表示 0、5、10，两组算盘珠组合起来表示 0～9）。

为发明自己的数字系统，罗马人可能推迟了计算工具的发展。在西班牙学习的 Gerbert（十世纪，后为教皇西尔维斯特二世）学到了阿拉伯数字系统和 Moors 制造的计算机。但无论是 Gerbert 还是 Moors 都未能使其计算机工作。

1642 年，法国哲学家和数学家帕斯卡（Blaise Pascal）发明了第一台能做加、减运算的手摇计算器。这是一种数字装置，每位数由一个轮子的位置来表示。当轮子转过 0 时，一个咬合装置把下一个高阶位轮置到 1。它与算盘的不同之处是实现了自动进位，当时被用于法国的税收，获得了很大的成功。

1673 年，德国哲学家和数学家莱布尼茨（Gottfried Leibniz）在 Pascal 计算器的基础上进一步研制出能做加法和乘法的机器。它由两部分组成，其中一部分就是加减机，与 Pascal 的计算器功能一样，乘法运算是利用反复相加来完成的，除法运算则是用反复相减来完成。这个方案到现在仍是应用的基础。它告诉我们如何利用简单机构来解决复杂的问题，从而使设计简单化，也成为应用数学在机器中的早期应用。

Leibniz 的更大贡献在于，他提出了人的逻辑思想是否能用机器所代替，这种思想成为后来数理逻辑思想的萌芽，即应创立一个逻辑系统，用数学语言把计算系统写成公式，并表达为逻辑形式，用于实践。Leibniz 的后半生一直从事这一工作，但未实现。

19 世纪中叶（1847 年），英国数学家乔治·布尔（George Boole）出版了《The Mathematical Analysis of Logic》，他认为可将代数形式表示为逻辑关系用于实际。Leibniz 与 Boole 的这种思想指导后世研究人员用逻辑设计实现算术和逻辑运算，为计算机理论研究开辟了新途径。这种理论研究在克劳德·香农（Claude Shannon）的电子开关电路设计中得到最早的应用，不过这是以后的事情了。

整个 18 世纪出现了一个十分平静的时期。相传 18 世纪末，J. H. Muller 设计了一台差分机，但是否做出样机存在疑问。1797 年，Charles Mahon 发明了逻辑演示器，一台巧妙地使用了齿轮和一个具有十多个支架的支撑装置。不过那时人们对自动计算机的认识还很肤浅，还不足以在理论方面有所突破，对计算机的概念仍处于模糊阶段，或者说只是计算机概念的探索阶段。因此，计算机的研制也仅限于关于计算操作进行简化的各种研究。

此时，纺织工业有创造性发展，这种发展对后来计算机的发展起了决定性作用。也就是说，计算机从许多方面借助了其他一些设备。1725 年，Basile Bouchon 在织布机上加了一条穿孔纸带，并在盒子里装了许多针，使针对着纸带，当针遇到纸有孔的地方就与之相应穿线使之织出美丽的图案。至此，发明出具有“存储”程序的第一条穿孔纸带，而后经 Joseph Marie Jacquard 加以改进，于 1801 年在法国巴黎成功地展示了一台穿孔卡片控制的织布机。

这一成果轰动整个学界。后来，Babbage 亲自到巴黎参观了 Jacquard 设计的纺织机，并买了一台仿制品献给 Sardinia 女王，同时提出了自己制造自动计算机的设想，得到英国皇家学会的赞助。至此，Babbage 开始了自己的研制工作。他设计的第一台差分机（difference engine）完成于 1822 年，用于计算导航表。1823 年，他着手为英国设计一台更大、更好的机器。

此后，Babbage 花了近 20 年的时间和精力，这台更大、更好的机器却始终未完成。他在 1835 年 5 月写给布鲁塞尔皇家科学院的一封信中曾对这台机器做了简要叙述。

这台计算机的功能结构类似现代计算机，有一个存储数码的存储器，以及一个称为“米尔”的运算器。数据和程序信息由穿孔卡片输入，输出借助打印机和穿孔卡片。运算器能执行加减乘除 4 种基本运算。这台机器还能执行判定操作，即根据计算结果符号改变计算操作。但这台机器采用蒸汽机作为动力，采用机械的方法是很难实现的。

1840 年，Babbage 在都灵（Turin）做的有关分析机学术报告中第一次提出了自动计算机的基本概念：存储（用卡片存储指令和数据）、运算器、输入/输出以及条件转移等，使计算机结构从概念探索过渡到产生概念的时代。

电子计算机的前身是继电器计算机。1937 年，美国哈佛大学研究生霍华德·艾肯（Howard Aiken）为解决数学中的多项式计算开始了对计算机的研制工作。1939 年 IBM 公司表示关注，1944 年 Mark－Ⅰ型自动序列计算机问世。Aiken 开始研究不久后发现了 Babbage 所做的工作，他认为自己的机器是 Babbage 梦想的实现。

1.2 从 ENIAC 到冯·诺依曼体系

1.2.1 ENIAC

1943—1946 年，宾夕法尼亚大学研制的电子数字积分计算机（Electronic Numerical Integrator and Computer，ENIAC）是电子计算机发展中真正具有重要意义的一步。

一种新的设计思想或装置的完全实现，不仅需要新的概念，而且需要对这种装置存在需求（即要解决的问题），以及实现此设计思想或装置的手段。Babbage 的努力未获成功，关键在于尽管客观上需要（航海表），但尚未具备制造相关设备的手段。

1943 年美国卷入第二次世界大战，阿伯丁弹道研究所在提供部队的武器火力表以便对导弹研制进行技术鉴定问题上遇到困难，客观上需要有高速的计算装置。此时，电子触发器已研制出来，雷达的发展提供了脉冲线路和电子开关元件，可以说设计制造电子计算机所需要的手段已全部具备。

1943 年 4 月，在 Eckert 和 Mauchly 提出的“高速电子管计算装置”（后定稿为“论电子微分分析器”）基础上，美国陆军部与宾夕法尼亚莫尔分校签订合同，在美国陆军部资助并主持下，由 Eckert 和 Mauchly 设计研制 ENIAC。1946 年 2 月 15 日，ENIAC 正式公布，1947 年投入运行。

ENIAC 的问世，意味着现代计算机的历史从此开始。这台计算机总共用了将近18 000 只真空管，1 500 个继电器，耗电量 140 kW，占地 170 m^2，重 30 t，每秒能做 1 900 次加法。ENIAC 的可靠运行时间大约不超过 20 分钟，一般的计算机编程需要半小时到 1 ~ 2 天时间。

ENIAC 采用十进制表示数据，其工作存储器只有 20 个单元，用于存放数据。ENIAC 的编程是外排式的。先要人工对操作面板上的 6 000 多个电子开关进行机械定位，然后靠转插线插头，通过插入或拔出方式来编制程序，靠穿孔卡输入数据。

1.2.2 冯·诺依曼体系

随着 ENIAC 研究的深入，几位知名数学家也对 ENIAC 发生了兴趣，其中有阿伯丁弹道研究所和洛斯阿拉莫斯科学实验室的顾问约翰·冯·诺依曼（John Von Neumann）博士。

ENIAC 问世后，冯·诺依曼和 ENIAC 研究人员一起讨论了各种想法和建议，如怎样以数学形式存储程序。其实，研究工作开始不久他就发现用大量开关和插头插线来编程十分费时，枯燥乏味且极不灵活。另外，他对 ENIAC 用 10 个电子管表示一个十进制数的笨拙方式提出异议。冯·诺依曼等人帮助研究人员明确了一些观点，并于 1946 年 6 月在一篇题为“电子装置逻辑结构初探”的备忘录中提出一种存储程序计算机，名为电子离散变量自动计算机（Electronic Discrete Variable Automatic Computer，EDVAC）。其主要思想如下：

（1）提出程序可以用数字形式和数据一起在计算机内存中表示。

（2）废除十进制数表示，提出可用二进制数来替代。

（3）计算机采用以运算器为核心的集中控制。

（4）提出指令概念，指令由操作码和地址码两部分组成，指令在计算机中是顺序执行的，并受指令计数器的统一控制。

（5）存储器有4 096个字，每个字40位，每位均有0或1。每个字表示两条20位的指令或一个40位的有符号数。指令中用8位来区分指令类型，另外12位表示4 096个存储单元的一个地址。

（6）计算机由5个基本部分组成：运算器、控制器、存储器以及输入/输出设备。

这个备忘录被当作如今广泛使用的冯·诺依曼计算机的基础。尽管学术界给了冯·诺依曼太多的荣誉，而给予Eckert和Mauchly则太少，但从奠定了冯·诺依曼型计算机设计规范这一点来说，冯·诺依曼获得怎样的殊荣都不为过。后来，由于Eckert和Mauchly等多位ENIAC研究人员的离去削弱了EDVAC的力量而延误了工期，直到1952年该工程才完成。

1.3 系列机与兼容机技术

早期计算机的研制是“井水不犯河水”的。例如，IBM公司在研制IBM 360之前有5种体系结构各不相同的机器在研制，每种机器只能使用自己特有的指令编程，彼此互不兼容。随着新型机的出现，旧机器以及与该机器使用的软件将一起被淘汰。由于淘汰率高，因此浪费了大量的人力、物力。

随着研究的深入，人们发现：

（1）同一种指令集结构可以有多种组成方式，由于指令集不变，所以可运行同样的软件。

这一思想造就了系列机。IBM公司提出的生产系列机思想在现在看来不算什么，然而在20世纪60年代初这种思想是十分超前的。IBM得益于这种思想，在单一指令集的基础上相继推出IBM 370/115、125、135、145、158、168，从低速机到高速机得到了发展。尽管它们采用了不同的组成和实现技术，但从程序设计者角度所看到的机器属性都是相同的。

DEC公司也得益于这种思想，继PDP－11之后相继推出了VAX/730、750、780以及后来的8000、6200、6300、6400系列。DEC的策略核心就是使用单一指令集结构、数据类型和寻址模式。系列机都采用了I/O设备、寄存器和主存统一编址方式，其组成和实现却不完全相同。为此，DEC在十几年中得到快速发展，后被Compag公司收购。

（2）由于一种组成可以有多种实现，因此，它不仅造就了系列机，也造就了兼容机。

一直以生产大中型机为主的IBM公司于1981年推出了IBM PC。在这次研发中，IBM公司避开了从设计草图开始到用其生产的零件来组装整台机器的传统做法，而是选择Intel公司的

8088 作 CPU，全部采用市场上的组件组装出 IBM PC。这种思想就是采用了兼容机设计理念。IBM 接着又推出了 PC/XT 和 PC/AT，这一时期选用的是 8086、8088 和 80286 芯片。其后，Intel 公司又推出了 80386、80486、…、Pentium Ⅲ、Pentium Ⅳ等。其他公司如 AMD、Cyrix、TI（Texes Iastrument）也生产出与之相兼容的芯片。尽管各种芯片的性质不同，但从概念上说都属于 IBM PC/AT 的兼容机，其实质仍属于 8086 处理器体系结构，除了寄存器从 16 位变成 32 位、64 位，其余并没有本质区别。然而，计算机的实现却有了不少的变化。

长期以来，软件工作者都希望有一个稳定的编程环境使其编写的程序得到广泛应用；计算机设计者又希望根据硬件技术的发展不断推出新的机型。系列机、兼容机的出现，较好地解决了软件要求稳定和硬件要求不断更新产品之间的矛盾，对计算机的发展起到重要的推动作用。

然而，值得注意的是，系列机为了保持软件的兼容而要求体系结构不能改变，这无疑又制约了计算机系统结构的发展。如何解决这一问题呢？考虑到系列机的兼容性有向上兼容、向下兼容、向前兼容和向后兼容之分，为了适应系列机性能不断提高和应用领域不断扩大的要求，机器的体系结构也要能扩充为好。解决方法只能是采用一种折中方案，放松对软件兼容性的要求，对硬件的改变必须是原有体系结构的扩充，而不是任意更改或缩小。所谓向上（下）兼容，指的是在同一时期按某档机器编写的程序可不加修改就能运行于比它高（低）档的机器。所谓向前（后）兼容，指的是某个时期投入市场的某种型号机器编写的程序，可以不加修改地运行于在它之前（后）投入市场的机器。

在 80X86/Pentium 系列机中就采用了这种思想，即向下、向前不做要求，向上兼容也不一定做到，只要保证向后兼容就可以了。比如说在同一时期有两种机型 80286 和 80386，前者是 16 位机，后者是 32 位机。显然，在 80386 机上编写的程序是不能在 80286 机上运行的，所以向下兼容不做要求。同样，80286 机上编写的程序，80386 机若不增加相应软件也很难运行，所以，向上兼容也不一定做到。然而 80386 必须满足向后兼容，因此在后期投放市场的 32 位机中增加了虚拟 86 模式（V86），这时有一个实际的操作系统在控制整台计算机。为了运行老的 8088 程序，操作系统创建了一个特殊的独立 8088 环境。

80X86、Pentium 系列机的实践证明，只要保证向后兼容这一基本原则，并在此前提下不断地改进机器的组成和实现，计算机设计过程就有了生命力。

1.4 层次技术

语言与层次，这是促进计算机设计发展的因素中极为重要的一环。计算机设计从最初的一、二层发展到现在的多层，使一台处理机具有更强大的处理能力，显然这是每个从事设计的技术人员十分感兴趣的问题。下面就这个问题展开讨论。

1. 语言与层次

人们如何把自己的思想告诉计算机？靠的是语言。语言是用于传达信息的约定和规则

集合。

然而，在方便人的使用和方便计算机的实现上存在着巨大差距。人可能要做X，而计算机只能做到Y。这就有问题了，如何解决呢？为此，人们发明了一系列程序设计语言。

例如，计算 d = a + b - c。

（1）可以用计算机内含的机器指令（即机器语言）编程。若令a、b、c、d分别代表地址0011、0012、0013、0014中的数据，在DJS-21机中可用如下指令序列来实现：

```
操作        地址        注释
012         0011        ;(0011) = a→ACC
018         0012        ;(ACC) + b→ACC
019         0013        ;(ACC) - c→ACC
004         0014        ;(ACC)→014
035                     ;停机
```

（2）由于机器语言对不同计算机来说大不相同，也就是说，每台计算机只懂得自己的那一套机器语言，为了改善机器语言的可读性产生了汇编语言。它使用英文助记符表示操作码，用变量A、B、C、D分别代表数据a、b、c、d所存放的地址，则上述问题可用如下指令序列实现：

```
Computer:  GET   A      ;(A) = a→ACC
           ADD   B      ;(ACC) + (B)→ACC
           SUB   C      ;(ACC) - ( C)→ACC
           PUT   D      ;(ACC)→D
           STOP         ;停机
```

如此种种。假如将机器指令组成的语言称为 L_0，用英文助记符组成的语言称为 L_1，…，依此类推。这样，L_0，L_1，…，L_i无限继续下去，直到最后找到一种最适合于设计人员的语言。

如果从语言的角度出发，就可以将系统按功能划分成不同语言级组成的层次结构。如：

L_0 语言级 对应于第0层；

L_1 语言级 对应于第1层；

…

L_i 语言级 对应于第 i 层。

随着程序设计语言被划分成一系列的层次（level）或级，会带来什么问题呢？

2. 如何让只能执行某种语言编写的程序的计算机执行另一种语言编写的程序

例如，如何让只能执行 L_0 语言编写的程序的计算机执行 L_1 语言编写的程序，一般有三种方法：

（1）在执行用 L_1 语言编写的程序之前，通过编译器让 L_1 语言生成一个等价的 L_0 语言指令序列。生成的程序全部由 L_0 指令组成，执行过程中运行的是新生成的 L_0 程序。

（2）用 L_0 语言编写一个称为解释器的程序，将 L_1 程序作为输入语句送入解释器，并逐条