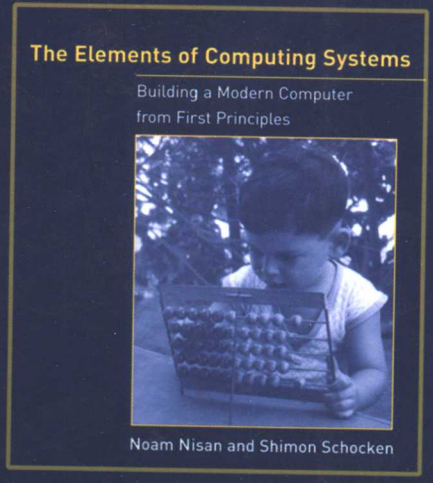
Broadview
WWW.BROADVIEW.COM.CN



The Elements of Computing Systems
Building a Modern Computer from First Principles

计算机系统要素
从零开始构建现代计算机

[美] Noam Nisan Shimon Schocken 著
周维 宋磊 陈曦 译 刘天田 审校

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

计算机系统要素

——从零开始构建现代计算机

The Elements of Computing Systems

Building a Modern Computer from First Principles

[美] Noam Nisan
Shimon Schocken 著

周 维 宋 磊 陈 曦 译

刘天田 审校

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内容简介

本书通过展现简单但功能强大的计算机系统之构建过程，为读者呈现了一幅完整、严格的计算机应用科学大图景。本书作者认为，理解计算机工作原理的最好方法就是亲自动手，从零开始构建计算机系统。

通过 12 个章节和项目来引领读者从头开始，本书逐步地构建一个基本的硬件平台和现代软件阶层体系。在这个过程中，读者能够获得关于硬件体系结构、操作系统、编程语言、编译器、数据结构、算法以及软件工程的详实知识。通过这种逐步构造的方法，本书揭示了计算机科学知识中的重要成分，并展示其它课程中所介绍的理论和应用技术如何融入这幅全局大图景当中去。全书基于“先抽象再实现”的阐述模式，每一章都介绍一个关键的硬件或软件抽象，一种实现方式以及一个实际的项目。完成这些项目所必要的计算机科学知识在本书中都有涵盖，只要求读者具备程序设计经验。本书配套的支持网站提供了书中描述的用于构建所有硬件和软件系统所必需的工具和资料，以及用于 12 个项目的 200 个测试程序。

全书内容广泛、涉猎全面，适合计算机及相关专业本科生、研究生、技术开发人员、教师以及技术爱好者参考和学习。

版权贸易合同登记号：图字：01-2006-3844

图书在版编目（CIP）数据

计算机系统要素：从零开始构建现代计算机 /（美）尼萨（Nisan,N.），（美）锡肯（Schocken,S.）著；周维，宋磊，陈曦译. —北京：电子工业出版社，2007.2
书名原文：The Elements of Computing Systems: Building a Modern Computer from First Principles
ISBN 978-7-121-03336-0

Ⅰ.计... Ⅱ.①尼...②锡...③周...④宋...⑤陈... Ⅲ.计算机系统 Ⅳ.TP30

中国版本图书馆 CIP 数据核字（2006）第 125896 号

责任编辑：周 筠
印 刷：北京智力达印刷有限公司
装 订：北京中新伟业印刷有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：787×980 1/16 印张：22.25 字数：380 千字
印 次：2007 年 2 月第 1 次印刷
定 价：45.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系电话：（010）68279077；邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

关键术语中英对照表

本书涉及的专业领域较为广泛，在阅读过程中会遇到各个领域的许多专业术语。下表列出了本书用到的主要术语的英文原文以及对应的中文译文，供读者参考和辨析（本书正文中出现的专业术语在必要时都标注了原文，以便于读者参考和使用索引）。

英文术语	中文译法
allocation	（内存）分配
assembly	汇编；汇编语言
assembler	汇编编译器
class	类
class type	“类”类型
compiler	编译器
constant	常量
constructor	构造函数
de-allocation	（内存）去配
dispose	清除
field	域；字段
function	函数；功能
identifier	标识符
invoke	唤起（执行）
label	标签
method	方法
object	对象
object-oriented	面向对象（的）
object-based	基于对象的
object instance	对象实例

英文术语	中文译法
parse/ parsing	（语法）分析
parser	（语法）分析器
parse table	（语法）分析表
parse tree	语法分析树；分析树
pointer	指针
procedure	过程
procedural	过程化（的）
reference	引用
subroutine	子程序
symbol	符号
syntax	语法
syntax tree	语法树
syntax analysis	语法分析
syntax analyzer	语法分析器
token	字元
tokenize	字元化
tokenizer	字元转换器
type conversion	类型转换
variable	变量

技术编辑尽了最大努力来保持术语中文译文的明确性和限定性，但仍可能难免存在疏漏之处，欢迎广大读者批评指正。

若您对本书在翻译方面的任何技术议题存在疑问，或对本书的翻译质量有任何意见、建议和批评，都欢迎您写信给本书的技术编辑（*fangzhou@broadview.com.cn*）。博文视点欢迎并感谢读者指出本书中存在的任何错误，更欢迎大家对博文视点出版的任何图书提出意见、建议和批评。感谢您对博文视点的支持！祝您读书愉快。

本书技术编辑　方　舟
电子邮件地址：*fangzhou@broadview.com.cn*
2007 年 1 月

前言

Preface

What I hear, I forget; What I see, I remember; What I do, I understand.

耳听为虚；眼见为实；实践出真知。

不闻不若闻之，闻之不若见之，见之不若知之，知之不若行之；学至于行之而止矣。[1]

——孔子（公元前 551 ~ 公元前 479 BC）

过去，凡是计算机专业人员都对计算机的工作原理和工作方式了如指掌。计算机体系中的硬件、软件、编译器以及操作系统之间的交互既简单又透明，因此要把握计算机系统大局观并非难事。然而随着现代计算机技术的日趋复杂，这种明晰性不复存在：计算机科学领域里面大多数基本思想和技术都被隐藏在众多抽象接口以及私有实现的层面之下。这种复杂性导致了无法避免的结果，即领域专业化；这使得多门计算机科学领域应运而生，每个领域只涵盖整个学科中的某一个方面。

之所以要编写这本书，正是因为本书作者有感于：很多学习计算机科学的学生识木而不知林，疲于埋头学习程序设计、各种理论以及工程知识，却失去了对计算机系统整体的把握和理解，未曾停下来欣赏计算机系统大局观的美景。这个大局观为我们展示的是：硬件系统和软件系统如何经由隐藏的抽象、接口以及基于各种约定的实现所编织起来的网，从而紧紧地关联在一起。由于没有从表及里地透彻领略这个繁复大局观的魅力，使很多学生和计算机从业人员产生了不安的感觉，因为他们并没有完全透彻理解和掌握计算机的内部工作原理。

本书作者相信，理解计算机工作原理的最好方法就是亲自动手，从零开始构建计算机系统。由此我们提出了如下的概念：描述一种简单但功能较强的计算机系统，让学生从最基本的逻辑门开始构建其硬件平台和软件层级。要做就要把事情做好。之所以这么说是因

[1] 这句话出自荀子《儒效篇》。——审校者

为，从头开始构建完整的通用计算机系统是个艰巨的任务。因此，本书不仅阐述如何构建计算机系统，而且让读者亲自参与实践，了解如何有效地计划和管理大规模硬件/软件开发项目。此外，我们从最原始最基本的构建模块开始，通过递归向上和逻辑推理等手段，展示如何构建复杂且有用的系统。

范围
Scope

本书通过一系列的硬件和软件实践项目，向读者展示计算机科学知识中的大部分核心内容。这些实践项目将会为您展示计算机科学中的理论知识和应用技术是如何应用于工程实践中的。本书涵盖如下主题：

- **硬件**：逻辑门；布尔运算； multiplexor（多路复用器）；触发器（flip-flop）；寄存器（register）；RAM 单元；计数器；硬件描述语言（HDL，Hardware Description Language）；芯片的仿真及测试。
- **体系架构**：ALU/CPU 的设计与实现；机器代码；汇编语言程序设计；取址模式；I/O 内存映像。
- **操作系统**：内存管理；数学计算程序库；基本 I/O 驱动程序；屏幕管理；文件 I/O；对高级语言的支持。
- **程序设计语言**：基于对象（object-based）的设计和编程模式；抽象数据类型；作用域；语法和语义；引用（reference）机制。
- **编译器**：词法分析；自顶向下的语法分析；符号表（symbol table）；基于堆栈（stack-based）的虚拟机；代码生成；数组和对象的实现。
- **数据结构和算法**：堆栈；哈希表；链表；递归；算术算法；几何算法；运行效率。
- **软件工程**：模块化设计；接口/实现范式；API 设计和文档；主动式测试（即极限编程理论中的单元测试等）；广义的程序设计概念；质量保证体系。

介绍这些主题的目的只有一个：从零开始构建现代计算机系统。这同时也是我们选择讨论主题时所遵循的原则，即本书集中讨论构建具有完整功能的计算机系统所需要的最小集合。因此，该最小集合中包含了很多应用计算机科学中的基本思想。

课程
Course

本书的读者对象主要是高等院校的计算机科学以及其他工程学科的本科生和研究生。以本书为基础的课程与普通计算机科学课程是“纵向交叉”的，学生在普通课程中的任何时候都可以学习本书的内容。本书有两个明显的课程切入点，一个是“CS-2”（即学完程序设计之后），一个是“CS-199”（即学完所有相关课程之后，作为总结性的综合课程）。“CS-2”课程可作为面向系统的计算机科学入门，而“CS-199”课程可作为全面的、面向项目的系统构建课程。此类课程可称为计算机科学的系统结构，或计算系统要素、数字系统构建、计算机系统构建，或是“让我们来构建计算机”等等。根据涵盖的主题以及教学进度计划，本书可以在一个学期内讲授，也可以分为两个学期进行讲授。

本书是完全与其他课程独立的，所需的预备知识仅仅是基本的编程能力（任何语言）。因此，本书不仅适用于计算机科学专业的学生，而且也适用于计算机实用技术专业的学生（以此学习硬件结构、操作系统以及现代软件工程的基本知识）。对于任何技术或科学专业的学生，只要学完一门程序设计课程之后，就可以利用本书以及本书配套的支持网站作为自学教程。

本书的结构
Structure

在开始的简介内容里，我们提出了构建的方法并预览了本书所讨论的主要硬件/软件抽象。这为第 1 至 12 章的阐述奠定了基础。从第 1 章到第 12 章，每章都要讨论三个内容：1）一种关键的硬件抽象或软件抽象；2）该抽象的实现原理；3）一个实际项目，用来构建并测试所构建的系统。前 5 章主要讨论简单的现代计算机硬件平台的构建。第 6 至 12 章描述一个典型的多层（multi-tier）软件阶层体系的设计与实现，包括构建一门基于对象（object-based）的程序设计语言和一个简单的操作系统。图 P.1 展示了完整的构建计划。

本书遵循“**先抽象再实现**”的方式，每一章开头先有**背景知识**部分，用来描述相关概念和通用的硬件或者软件系统。接下来的**规范详述**部分提供了对系统抽象的明确描述，即

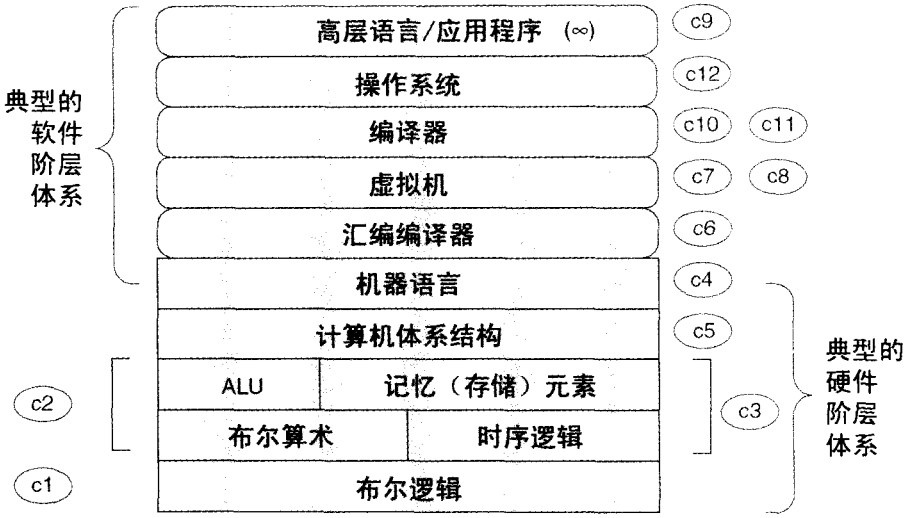


图 P.1 本书及课程概览，圆圈中显示了涉及到的章节

它能提供何种服务。再接下来的**实现**部分继续讨论如何来实现抽象。之后的**观点**部分强调每章当中没有考虑却又值得注意的一些问题。每章的最后都是**项目**部分，提供了逐步进行构建的说明、测试程序和用于实际进行构建并进行单元测试的软件工具。

项目
Projects

本书介绍的计算机系统是实际可用的系统，可被实际地构建出来，并能够正常运作。只要读者跟随本书的脚步，逐步构建出这个计算机系统，就会对计算机系统有更透彻的理解，这与只阅读而不动手实践所达到的学习效果相比有天壤之别。因此，本书在内容的编排和选择上着重针对那些卷起袖子准备动手实践，从零开始构建计算机系统的读者。

每一章都完整阐述了一个单独的硬件或软件开发项目。用来构建计算机系统硬件平台的前四个项目（第 1 至 4 章的项目），是利用一个简单的硬件描述语言（HDL）来实现的，并可在与本书配套的硬件仿真器上进行模拟测试。后续第 5 至 9 章的 5 个项目（即汇编编

译器，虚拟机第 I 部分和虚拟机第 II 部分，以及编译器第 I 部分和编译器第 II 部分）可以采用任何现代编程语言来编写。最后第 10 至 12 章的 3 个项目（底层编程、高级编程和操作系统）则采用之前项目中实现的汇编语言和高级语言来编写。

项目提示 本书总共有 12 个项目。按照一般的大学课程安排，完成每个项目大约会占用一周课外作业时间。这些项目完全是各自独立的，学生可以按任何顺序来完成（或者选择其中的几个来完成）。当然，“完整的体验”则需要按照顺序来逐个实现，这只是其中一种选择而已。

在讲授这门课程的时候，我们会做两个大“让步”。首先，除非是一些关键的情况，否则我们不去考虑优化问题，这个重要课题还是留给相应的专业课程来涵盖。其次，在开发翻译器软件包（汇编编译器、VM 实现和编译器）的时候，我们提供了无错误的测试文件（源程序），以便让学生在翻译器的输入没有错误的前提下对自己构建的模块进行测试。因此，学生不必编写处理错误和异常的代码，从而使得软件项目更易于管理。当然，处理错误输入是一门重要课程，但我们认为学生可以在其他课程中学习这些相关的内容，比如专门的编程和软件设计课程。

软件
Software

本书的支持网站（*www.idc.ac.il/tecs*）提供了用于构建书中描述的所有硬件和软件系统所需的工具和资源。其中包括硬件仿真器、CPU 仿真器、VM 仿真器和汇编编译器的可执行版本，虚拟机、编译器，以及本书所介绍的操作系统。网站还提供了所有项目需要的资源——大约两百个测试程序和测试脚本，用于各个项目的开发和单元测试。提供的所有软件工具和项目资源都可以在任何装有 Windows 或 Linux 操作系统的计算机上使用。

致谢
Acknowledgments

本书中的所有软件都是由 Interdisciplinary Center Herzliya 的 Efi Arazi 计算机科学学院的学生们开发的。首席软件架构师是 Yaron Ukrainitz，开发人员包括 Iftach Amit、Nir Rozen、Assaf Gad 和 Hadar Rosen-Sior。与这些学生兼开发人员一起工作非常愉快，能够在他们的教育生涯中担任他们的老师，我们深感荣幸和自豪。也要感谢我们的助教，Muawyah Akash、David Rabinowitz、Ran Navok 和 Yaron Ukraintz，他们帮助我们针对这门课程作了很多前

期探索性的教学（这正是本书的基础蓝本）。还要感谢 Jonathan Gross 和 Oren Baranes，他们在 Danny Seidner 博士的精心指导下从事着相关项目的研究；感谢 Uri Zeira 和 Oren Cohen，他们为 Jack 语言设计了集成开发环境；感谢 Tal Achituv 在开源问题方面的建议；感谢 Aryeh Schnall 仔细阅读本书并提出了一些编辑方面的细致建议。

在编写本书的同时还要兼顾我们日常的教学任务是不太容易的，所以我们还要感谢 Efi Arazi 计算机科学学院的主任 Esti Romen 在困难时期为我们承担了很多责任。最后，我们要感谢那些使用过本书早期版本并帮助我们改正了很多错误的许多学生。希望他们在学习过程中领会到 James Joyce（詹姆斯·乔伊斯，著名作家）所洞察的真谛："mistakes are the portals of discovery（错误是发现探索的温床）"。

Noam Nisan
Shimon Schocken

目录

Contents