

世界著名计算机教材精选

# 计算机系统设计原理

Jerome H. Saltzer  
M. Frans Kaashoek

著

陈文光 张广艳  
陈 康 薛瑞尼

译



## PRINCIPLES OF COMPUTER SYSTEM DESIGN

清华大学出版社



世界著名计算机教材精选

**Principles of Computer System Design**

# 计算机系统设计原理

Jerome H. Saltzer      著  
M. Frans Kaashoek

陈文光 张广艳      等译  
陈 康 薛瑞尼

清华大学出版社  
北 京

Principles of Computer System Design  
Jerome H. Saltzer, M. Frans Kaashoek  
EISBN: 978-0-12-374957-4  
Copyright © 2011 by Elsevier. All rights reserved.

Authorized Simplified Chinese translation edition published by the Proprietor.

Copyright © 2011 by Elsevier (Singapore) Pte Ltd. All rights reserved.

Printed in China by Tsinghua University Press under special arrangement with Elsevier (Singapore) Pte Ltd. This edition is authorized for sale in China only, excluding Hong Kong SAR and Taiwan. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书简体中文版由 Elsevier (Singapore) Pte Ltd. 授予清华大学出版社在中国大陆地区(不包括香港、澳门特别行政区以及台湾地区)发行与销售。未经许可之出口,视为违反著作权法,将受法律之制裁。

北京市版权局著作合同登记号 图字:01-2009-6312 号

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

#### 图书在版编目(CIP)数据

计算机系统原理/(美)索尔特(Saltzer, J. H.), (美)卡休克(Kaashoek, M. F.)著;陈文光等译.  
—北京:清华大学出版社,2012.12

(世界著名计算机教材精选)

书名原文:Principles of Computer System Design

ISBN 978-7-302-29459-7

I. ①计… II. ①索… ②卡… ③陈… III. ①电子计算机—系统设计—教材 IV. ①TP302.1

中国版本图书馆CIP数据核字(2012)第162787号

责任编辑:龙啟铭

封面设计:傅瑞学

责任校对:梁毅

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:清华大学印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:24.75

字 数:617千字

版 次:2012年12月第1版

印 次:2012年12月第1次印刷

印 数:1~3000

定 价:49.00元

产品编号:035037-01

# 译者序

计算机系统作为计算机科学的核心内容之一，一直在传统的大学本科课程体系中占有重要地位：汇编语言、数字电路、计算机原理、计算机系统结构、操作系统、编译原理、数据库系统、计算机网络等都是目前大部分计算机系的必修课。

传统的课程体系主要面临 3 个问题：

(1) 孤立地在不同课程中学习知识，缺乏对计算机系统整体原理的把握。例如“虚存”这一概念，在计算机原理、计算机系统结构和操作系统课程中都有所涉及，分别从不同层面介绍了虚存，但学生学完之后可能还是对这一概念缺乏整体认识。

(2) 对编程和硬件设计能力要求很高。计算机学科的覆盖面十分宽广，既有偏工程方向的学生，也有偏理论方向的学生。在现有课程体系中，上述课程实验部分对他们一视同仁地要求编程或硬件设计等系统实现技能，会使得偏理论方向的学生丧失兴趣，而具体的实现技能对他们今后的工作本来也作用有限。

(3) 课时占用过多。在大学应开展通识教育已成为教育界的共识，因此专业必修课的课时必然受到压缩。上述计算机系统相关的课程已有 8 门课，在通识教育的趋势下很难为这些课程找到足够的课时。

因此，如何改进现有的计算机系统相关的课程体系，是国内外计算机教育界关心的一个重要课题。

美国麻省理工学院（MIT）在计算机本科教育方面具有全球领先地位，本书就是 MIT 课程号为 6.033 的“Computer Systems Engineering”（计算机系统工程）教材，体现了 MIT 对计算机系统本科课程体系的思考。

在 MIT 计算机系的本科课程体系中，“计算机系统工程”是计算机系统方面唯一的必修课，其他课程如操作系统，编译原理、数据库系统等课程都列为选修课。“计算机系统工程”着重介绍在计算机软硬件系统中通用的原理，例如抽象、命名、模块化等，并用实际的系统例子来说明这些抽象的概念。学生在这门课程的学习中，不需要编写程序或实现硬件系统，仅需完成若干设计报告以及练习使用若干计算机系统即可。可以看出，这样的课程设置恰好解决了传统计算机系统课程的 3 个缺点：学生对计算机系统的基本概念有了整体、深入的认识，对偏理论的学生来讲也很适用，同时大大缩减了必修课程的课时。因此，在这本书于 2009 年出版后，我们迅速推荐清华大学出版社引进了影印版，同时开始了这本书的翻译工作。希望本书翻译版的问世，能够进一步推动国内计算机系统方面的教育改革。

译者在清华大学利用本书的部分内容在计算机系统结构课上进行了试讲，发现利用本书内容开设课程对教师具有极高的要求，需要对多种计算机系统有深入的了解，能够对所述原理旁征博引、信手拈来，否则学生很容易觉得所教的内容过于空泛。

无论高校开课与否，这本书以极其丰富的参考文献和网上材料，可以成为大学生很好的自学教材和参考书，也适合计算机专业人士自修提高使用。

推荐每一个严肃的计算机专业学生，都读一读这本书。

# 前 言

据我们所知，本书在内容和方法上是非常独特的。本书深入而且广泛地介绍了构造计算机系统的主要原理和概念，这里我们所讲的计算机系统，是指广义的计算机系统，包括操作系统、客户端-服务器应用程序、数据库系统、安全的 Web 网站以及容错的磁盘集群等。这些原则和抽象是不受时间影响的，不管读者是否是计算机系统专业的学生或专业人士，这些原则都具有重要价值。这些原则和抽象经过了多代计算机系统的证明，本书作者具有构建计算机系统的经验，并讲授了该课程数十年。

本书介绍了计算机系统中相当广泛的原理和抽象，并深入介绍了它们。本书使用伪代码介绍核心概念，使得读者可以测试他们对于该概念的具体实例的理解。通过使用伪代码，本书将客户端-服务器计算、远程过程调用、文件、线程、地址空间、尽量送达网络、原子性、验证的消息等概念的精华展现给读者。在本书的习题中，我们延续了这一方法，使得读者可以通过研究伪代码来探讨。

本书分成两部分，第一部分是读者拿到的书籍，仅包括前 6 章，第二部分包括第 7~11 章和一些辅助材料，已发布在网络上作为开放的教育资源。请参见后面的“到哪里找到第二部分和其他在线材料”。

## 为什么写这本书

许多计算机系统中的基本概念，例如，设计原则、模块化、命名、抽象、并发性、通信、容错以及原子性在计算机科学与工程课程体系中的好几门课程中都会介绍到。典型的计算机科学与工程课程体系通常包括两门基础课程，其中一门是编程课程，一门是硬件课程，然后就分为若干个方向，其中计算机系统相关的选修课通常包括：

- 操作系统；
- 网络；
- 数据库系统；
- 分布式系统；
- 编程语言；
- 软件工程；
- 安全；
- 容错；
- 并发性；
- 体系结构。

上面这个课程列表的主要问题是，在过去 30 年中这个列表在不断增长，大部分对计算机系统有兴趣的学生没有时间学习列表中的全部甚至是部分课程。常见的解决方案是“选

三门课”或“选操作系统和另外两门课”。结果是，大部分学生对于列表中的其他课程完全没有任何背景知识。更进一步，没有哪门选修课可以把其他选修课作为先修课程，所以这些课程中的共同内容在每门课中都被重复讲授。最后，对于不希望选择计算机系统作为专业方向但希望学习一些计算机系统背景知识的学生，除了选择一两个计算机系统领域进行深入学习外没有其他选择。

本书选取了上面所有课程中的共同机制和设计原则，并深入解释了一组仔细挑选过的重要思想。这使得可以基于本书开设一本本科生核心课程，适用于计算机科学与工程系的所有学生，不管他们是否希望把计算机系统作为自己的专业。一方面，将来只作为计算机系统用户的学生可以获得关于计算机系统的坚实基础；另一方面，希望把设计计算机系统作为自己未来职业的学生可以更加有效地学习不同的选修课，这些选修课和以前的课程项目相比，将具有更加深入的内容和更少的重复。这两类学生都能够在课程中学到经得起时间检验的概念，而不是只学习一些当前流行的短命技术。我们发现在 MIT 这种课程结构非常有效。

本书在达到其覆盖的广度的同时并没有牺牲深度。本书集中讨论底层的经得起时间考验的概念，而不是介绍对现有系统运行机制的细节描述，使得读者可以在其整个职业生涯中受益。本书的基本思路是进行教育而非工作培训。例如，书中的内容并不会讲授某个具体操作系统，或依赖某种特定的计算机系统结构。相反地，本书会介绍能够体现现代计算机系统主要思想的模型，并进行适当的简化。我们的教学模型是，如果读者能够理解本书中介绍的概念，就能够很容易地从其他书籍或是产品文档中学到对具体系统操作机制的细节。与此同时，本书使用伪代码片段来具体地展示概念，学生可以通过伪代码来检验自己是否真正理解了这些概念。

## 本书的读者

本书的读者包括从事下列工作的学生和职业人士：

- 设计计算机系统；
- 审查计算机系统的设计；
- 构建应用于信息管理的计算应用程序；
- 在某个机构内进行计算机系统集成；
- 评估计算机系统性能；
- 更新计算机系统；
- 继续学习网络、安全或事务管理等内容；
- 在计算机科学与工程的其他领域工作，但需要对计算机系统中的主要概念有基本了解。

### 难度

本书提供了计算机系统的入门介绍，并不会面面俱到或是在某个方面非常深入。本书的目标是让读者能够对计算机系统的复杂性获得洞察力，并在其职业生涯中获益。本书提供的内容可以让读者理解一些与计算机系统设计者进行交互所必需的概念，包括操作系统、

数据库系统、数据网络、计算机安全、分布式系统、容错计算和并发性等领域的坚实基础。读完本书后，读者应该可以进一步了解计算机系统的许多部分的详细设计与实现，阅读并理解关于计算机系统的最新专业文献，并了解应该问什么样的问题以及到哪里去找答案。

可以有若干种方法使用本书。可以将本书作为 1~3 学期的计算机系统系列课程教材。或是从本书中选取一两章，作为对传统本科或研究生选修课的引言部分，这些课程包括操作系统、数据库系统、数据网络、计算机安全、分布式系统、容错计算和并发性等。这样，本书可以在不同课程使用多次。另一种用法是在计算机系统的研究生课程中，把本书作为在本科学习过的相关课程复习，并补充学习在本科没有学过的部分。

### 先修课程

本书非常小心地限制了其先修课程。当作为教材使用时，本书可以用于已学过软件设计和计算机硬件组成原理入门课程的大三、大四学生，并不需要更加深入的计算机科学与工程方面的背景知识。本书在使用新的术语之前都会定义，并避免了使用过多的行话。当然，本书还是假定读者已经从暑期工作或是课程实验中获得了使用计算机系统的实际经验，读者并不需要能够熟练地使用某种计算机语言编程，但需要能把对计算机编程语言方面的知识用于理解书中伪代码所用的编程语言。

### 其他读者

专业人士也会发现本书是有用的。本书提供了一种现代和前瞻的角度介绍计算系统设计的方式：主要基于实施模块化的思想。这种视角反映了在过去的 10~20 年中，计算机系统设计师所面临的主要挑战从解决资源限制问题变成了如何能让系统具有可控的复杂度。对于那些在大学期间只学习了以资源管理为中心的部分计算机系统或操作系统课程的专业人士来说，这本书将帮助他们从现代和更广阔的视角来更新他们的知识。

## 如何使用本书

### 练习和问题集

本书每章的最后都有一些简答练习题，用来测试对该章所讨论的部分概念的理解。在本书后，有一些更长的问题集，用来考察用户是否能将所学的概念应用于实际生活中遇到的类似问题。在大多数情况下，这些问题集需要应用多个章节的概念。每个问题集都会标明其主要关注的章节，但后面章节的问题集通常会使用前面所有章节的概念。教师可以联系本书的出版社，获取这些练习和问题集的答案。

这些练习和问题集可以有如下用法：

- 作为学习工具。在这种模式下，可以把答案交给学生。鼓励学生自己完成练习和问题集，然后对照自己的答案和我们的答案，学生可以得到立即的反馈，纠正他们对概念的错误认识，对误解和二义性的内容提出问题。一种可以鼓励学生研究这些练习和问题集的方法是宣布在考试中将使用相同或类似的问题。
- 作为作业或考试材料。在这种模式下，课后练习和问题集被用来作为作业，学生提交自己的答案，老师评分后把正确答案一并返回。



- 作为新的练习和问题集的点子的来源。

### 实例研究和阅读

为了补充书中的内容，读者应该补充阅读相关专业技术文献和实例研究。在最后一章后面，有一个精选的论文和参考书列表，这些论文和书籍提供了智慧、系统设计原则以及实例研究。通过采用不同的进度以及阅读深度，本书可以作为第 1 学期的本科生核心课程，第 2~3 学期的本科生系列课程，或者是研究生级别的计算机系统入门课程。

### 大作业

我们的经验是，对于这样一门接触到计算机系统的许多方面的课程，一些简单的上机实验（例如通过实验测试一台 PC 的 Cache 大小，或是在互联网上跟踪不对称的路由），加上 1~2 个纸面大作业（例如写一份 10 页纸的报告，描述如何为美国国会图书馆设计可靠的数字存储系统），可以成为教材内容的良好补充。另一方面，需要深入了解具体系统内部情况的大型编程作业会花费太多的时间，对于我们这样一门着重于广度的课程来说，会让学生负担过重。当然，在后续的选修课中确实应该包括需要大量编程的大作业，例如操作系统、网络、数据库或分布式系统等。因此，在 MIT，我们在后续的高级选修课上而不是在本课程留编程大作业。

### 支持

本书的支持通过一些在线资源完成，包括课程规划、阅读列表、问题集、课堂录像、小测验以及测验答案等。出版者的网站也为学生、专业人士和教师提供了一些资源和链接。MIT 的 6.033 课程使用了本书作为教材，其网站提供了教师和当前学生交互的情况，包括当前或最近学期的课程公告、阅读材料以及作业等。除了当前学期课程的材料外，该网站还包括了从 1995 年以来的课程材料，包括：

- 设计大作业；
- 上机实验；
- 考试和答案（部分与教材上的练习与问题集重复，但还包括了课外阅读材料的考试问题与答案）；
- 讲课与习题课计划；
- 阅读作业以及阅读报告的问题。

请参见后面的“到哪里去寻找本书第二部分以及其他在线材料”章节，以找到这些在线资源。

## 本书的内容组织

由于不是每个教师都希望使用本书的每一章，本书采用了一种创新的模式来展现（至少在本书出版的时候这是一种新的模式）：本书的前 6 章采用传统印刷书籍的形式，因为作者认为这些内容是什么计算机系统课程都需要的核心内容。剩下 5 章可以从作者或 MIT 通过在线资源的方式提供，这些资源采用 Creative Common License，允许免费、无限制的非商业使用和重组。这些在线章节也可以通过出版者的网站获得。在核心章节的内容中有很

多对后面章节的向后引用，例如“这个主题在第 7.4.1 节中有更深入的探讨[在线]”。<sup>1</sup>

## 主题

本书有 3 个主题贯穿始终。首先，正如本书的题目所述，本书强调了系统化设计原则的重要性。在第一次介绍每个设计原则的时候，我们用可帮助记忆的警句形式表达该原则。当在本书的后面再次提到该原则的时候，我们用其名字引用，并采用特殊的打印字体，以提醒读者该原则的广泛适用性。本书的第二个主题是以网络为中心的，在开始的章节就介绍了网络与通信，并在后续章节中使用这些内容。第三个主题是以安全为中心，我们在前面的章节中就介绍强制的模块化，并在后面章节中介绍如何实施更加严格的方法来提高安全性。安全性是作为本书最后一章出现的，不是因为安全是后来才添加的内容，而是因为安全性是实施模块化开发的逻辑结果。传统教材通常从资源分配问题来看虚存和线程，本文主要从加强模块化的角度来讨论这些问题，同时利用多处理器和大地址空间所带来的好处。

## 术语和例子

本书讨论了多个专门领域所共同的概念和设计原则：软件工程，程序语言，操作系统，分布式系统，网络，数据库系统以及系统结构。有经验的计算机专业人士可能会发现至少在本书的某些部分使用了不太常见的例子、思维方式或是术语，与在某个领域内解释其经典问题的传统方式不符。但不同领域的人可能会给出不同的不符内容，产生这种情况的原因是：历史上不同领域的人遇到了具有相同基本概念和设计原则的问题，但在不同领域中使用了不同语言、不同的角度、不同的例子和不同的术语来解释。

对于每个概念，本书作者选取了从教学角度最有效的解释和例子，并尽可能地采用广泛使用的术语。在不同领域具有相互冲突的术语的场合，我们通过词汇表和边栏来提供对术语的讨论。这样做的结果是，我们采用了一种新的但是有效的方法来向新一代计算机科学与工程专业的大学生讲授计算机系统设计的基础知识。以本书内容为基础，当读者再阅读更加深入的书籍或论文时，或选修高级选修课时，就能够很快识别出在专门领域术语包装下的那些熟悉的概念。科学家可以说“物理与测量的单位无关”来解释这种方法，在计算机系统领域同样的原则也适用：“概念与术语无关”。

## 引用

本书没有采用学术引用的方法来指明每个概念或想法的来源，如果这么做的话，本书将会比现在厚一倍。我们使用引用来指明那些我们认为值得知道的材料。只有一个例外之处：某些章节是关于战争故事的，这些故事在代代相传时被不断修改。因此我们给出了这些故事已知的最初来源，使得读者可以验证这些故事。

# 章节内容

与 ACM/IEEE 推荐的课程体系的关系。2001 年和 2004 年 ACM/IEEE 推荐的计算机科

1 [在线]表示在线电子文档，可从相关链接上下载，后同。

学与工程课程体系描述了两个层次。第一层由一些模块组成，这些模块组织起来成为恰当的计算机工程与应用教育内容。第二层是对如何把这些模块组合成以学期为单位的课程提出了建议。本书可以看成是一种对这些模块的现代化的、独特的组合方法，与 ACM/IEEE 计算机科学 2001 建议的 CS226c（操作系统与网络（压缩版））有些类似，但增加了命名、容错、原子性以及系统和网络安全方面的内容。本书与 ACM/IEEE 计算机工程 2004 建议的 CPED203（操作系统和以网络为中心的计算）也比较相似，增加了命名、容错、原子性和加密协议方面的内容。

**第 1 章：系统。**这一章介绍了作者关于如何思考关于系统问题的一般性的哲学思想，并用例子展示计算机系统与其他工程系统的相似和不同之处。本章还介绍了 3 个主要思想：

- (1) 系统性设计原则的重要性；
- (2) 模块化在控制大系统复杂度方面的作用；
- (3) 实施模块化的方法。

**第 2 章：计算机系统组成的基本方法。**本章介绍了在计算机系统中实施模块化的三个关键方法：抽象、命名和分层。关于抽象的讨论从计算机系统的角度简要回顾了计算机系统结构，并为本书余下章节建立了平台，但并没有简单重复那些读者可能已经知道的材料。命名模型是计算机系统的模块化的基础，但主要在程序设计语言的高级课程中介绍。本章进行了实例分析，考察了命名、分层和抽象分别是如何在 UNIX 文件系统中得到应用的。实例分析是通过一系列伪代码片段来展现，为解释这些概念提供了实际的例子，并为后续章节提供了参考的基础。

**第 3 章：命名方案的设计。**这一章继续讨论在系统设计中的命名问题，介绍了命名所需考虑的实际工程因素，强调了命名在将一些模块组织成计算机系统时的重要作用。本章包含了实例分析和一组战争故事（War Story）。实例分析以互联网上的 URL 为例来讨论几乎所有的命名设计因素。战争故事是一组实际的命名失败的例子，表明当设计者不了解设计原则时可能会发生什么问题。

**第 4 章：通过客户端和服务来保证模块化。**前三章说明了模块化的重要性。本章则介绍了客户端-服务模式作为实施模块化的一种方法。客户端-服务模式是一种广泛使用的模块间交互的方法，避免了不相关模块的干扰。本章还是以网络为中心这个主题的开始章节，这个主题将贯穿本书余下的部分。在这一章，我们仅将网络看成是抽象的通信系统，用来在客户端和服务之间提供强边界保护，我们通过两个实例分析来进一步讨论相关概念的细节。第一个例子是互联网域名服务系统（DNS），为第 3 章和第 4 章讨论的概念提供了实际展现。第二个实例分析是 Sun 公司的网络文件系统（NFS），我们以第二章中对 UNIX 文件系统的实例分析为基础，对远程服务队应用程序编程接口语义的影响进行了说明。

**第 5 章：通过虚拟化实现模块化。**这一章介绍了如何通过虚存和虚拟处理器（通常称作线程）来进行模块化。不管是对内存还是线程，我们都从没有资源限制的环境开始讨论。关于虚存的讨论从一个具有许多线程的无限地址空间开始，然后增加保护机制（寻址域和用户/内核态区分），以防止线程无意中破坏其他线程的数据。最后，我们考察在有限的地址空间下的情况，需要引入虚拟地址以及地址转换，以及由此产生的跨地址空间

通信问题。

类似地，对线程的讨论从假设处理器数与线程数一样多开始，先集中于讨论如何协调线程的并发行为。然后我们讨论当实际处理器数有限的情况，此时我们需要线程管理机制。在讨论线程协调的时候我们用到了事件计数和序列号等机制，这些机制在实际中并不常见，但对我们的例子而言非常自然。传统上，线程协调问题是新手学习时最为困难的概念之一。因此，我们设计了问题集，用信号量和条件变量来让读者考查他们对这些原理的掌握。

本章通过既使用文字也使用伪代码来清晰说明这些抽象的概念是如何实际工作的，并使用了与实际生活相似的问题。此外，对线程协调的讨论可以看成是理解原子性的第一步，我们将在第九章讨论原子性[在线]。

本章最后包括了一个实例分析和一个应用。实例分析讨论了在 Intel x86 处理器家族中模块化是如何演进的。应用则是如何利用虚拟化来创建虚拟机。本章内容的重点是实现模块化，并尽量利用现代硬件体系结构的优势（多核处理器，64 位地址空间，可直接访问的内存以 GB 计），而不是资源管理。

**第 6 章：性能。**本章集中讨论了在很多类型的计算机系统中都遇到的共性性能瓶颈，这些系统包括操作系统、数据库、网络以及大型应用。本章探讨了传统操作系统书籍的两个传统主题——资源调度和多层存储系统。但我们对性能优化重要性的讨论是在如下背景下展开的：某些底层硬件每 10 年会改进 1000 倍，但对其他性能指标的影响却可能并不大。因此，我们探讨调度问题时，讨论了磁盘臂的调度问题而不是分时处理器的调度器。

第 7~11 章是在线的，作为本书的第二部分。第二部分的介绍请参见“关于第二部分”，关于如何找到第二部分的章节可以参见“到哪里去寻找本书第二部分以及其他在线材料”。

进一步阅读推荐包括了阅读列表以及为什么这些材料值得进一步阅读的说明。选择这些书籍和论文的重点在于它们能提供洞察力而非细节知识。

**问题集：**本书作者不仅把考试当做一种评估方法，也当做一种教学的手段。因此，有些练习和本书最后的问题集（所有这些都是从讲授本课程的考题中改编而来的）并不是对概念的简单练习。通过做这些习题，学生探索替代方案，学习教材上方法的变种，并熟悉那些在实际系统中提出或使用过的有趣甚至怪异的想法。问题集通常需要创造性地应用所学的概念，目标是理解使用相关方法时需要做的设计权衡。

**词汇表：**正如前面提到的，计算机系统领域的文献是由不同专门领域构成的，这些领域有自己的系统相关概念的词汇表。本书采用了一致性的术语，词汇表对重要的术语进行了定义，并指明哪一章引入了该术语，在很多情况下解释了不同人在不同领域所使用的不同术语。为了词汇表的完整性和易检索，词汇表包括了本书第二部分引入的术语。

**概念索引：**这个索引指向对概念的定义和讨论。此外，还列出了每个设计原则的每个应用（为了完整性起见，本索引包括了在本书第二部分引入的概念，并列出了其所在的章节号）。

## 到哪里去寻找本书第二部分以及其他在线材料

(1) Saltzer 和 Kaashoek 教授以及 MIT OpenCourseWare<sup>2</sup>提供了第 7~11 章的免费在线版本、附加的问题集、词汇表以及完整的目录。所有上述资料都以 PDF 格式提供, 包括每个章节的 PDF 和一个整个的 PDF 文件。这些材料可以在如下链接找到:

<http://ocw.mit.edu/Saltzer-Kaashoek>

(2) 本书的出版商维护了相关的在线资源:

[www.ElsevierDirect.com/9780123749574](http://www.ElsevierDirect.com/9780123749574)

点击链接“Companion Materials”(相关资料)就可以找到本书的第二部分以及其他资源, 包括用各种格式表示的书中所用的图。提供给教师的附加资料(需要注册)可以通过点击链接“Manual”找到。

(3) 教学和其他支持材料可以在以下链接找到:

<http://ocw.mit.edu/6.033>

(4) MIT 使用本教材的当前的课程网站, 包括以前课程的网站的存档, 在以下链接可以找到:

<http://mit.edu/6.033>

(网站上某些有版权或涉及隐私的材料仅限当前的 MIT 学生访问)

## 致 谢

本书始于作者在 MIT 电子与计算机工程系讲授的计算机系统工程课程(课程号 6.033, 原课程号是 6.233)讲义, 该课程从 1968 年开始讲授。本书得益于 40 年来许多教师、访问者、习题课教师、助教以及学生的评论与建议。超过 5000 名学生使用过本书的草稿, 我们从他们学习经历中(以及由于课本的原因造成的困扰)得到了如何编写本书的指导。我们对他们的贡献表示感谢。此外, 以下方面值得特别致谢。

### 1. 命名(第 2.2 节和第 3 章)

关于命名的内容, 其概念和组织是与 Michael D.Schroeder 进行广泛讨论后的结果。命名模型(以及部分实现)与 D.Austin Henderson 的博士论文中的模型非常接近。Stephen A. Ward 建议对命名模型做某些有用的一般化, Roger Needham 对本书的早期版本提出了一些概念上的建议。本书的早期版本, 包括将命名模型应用到寻址结构和文件系统的深入实例, 以及相关历史文献, 曾在 Rudolf Bayer 等人编辑的 *Operating Systems: An Advanced Course, Lecture Notes in Computer Science 60* 的第 99~208 页作为该书第 3 章发表, 该书由 Springer-Verlag 于 1978 年出版, 1984 年重印。Ion Stoica, Karen Sollins, Daniel Jackson, Butler

---

<sup>2</sup> MIT 的 OpenCourseWare (开放课件) 计划提供了对许多 MIT 课程的在线非商业的免费访问, 目标是建立起科学与工程领域课程的标准。除了本书第 7~11 章外, OpenCourseWare 网站还发布了 MIT 使用本书的课程 6.033 的一些在线资料。有兴趣使用本书作为教材的老师可以在该网站上找到课程计划、阅读列表、问题集、课程录像、测验和答案等。

Lampson, David Karger 以及 Hari Balakrishnan 贡献了一些其他的主意。

## 2. 实施模块化和虚拟化（第 4 章和第 5 章）

第 4 章深受 David L.Tennenhouse 关于相同主题的讲座的影响。这两章均得到以下人士的许多反馈并进行了改进，提供意见的人士包括：Hari Balakrishnan, Russ Cox, Miachael Ernst, Eddie Kohler, Chris Laas, Barbara H. Liskov, Nancy Lynch, Samuel Madden, Robert T. Morris, Max Poletto, Martin Rinard, Susan Ruff, Gerald Jay Sussman, Julie Sussman 以及 Machael Walfish。

## 3. 网络（第 7 章[在线]）

本章组织结构确定得益于与 David D. Clark 和 David L.Tennenhouse 的交谈，本章部分内容的表达基于 Clark 的讲座。Robert H. Halstead Jr. 撰写了关于网络的早期讲义，我们也借用了他的一些思想。Hari Balakrishnan 提供了许多建议和修订意见，并帮助将一些混乱的解释写清楚，Julie Sussman 和 Susan Ruff 指出了本书表达方面许多可改进的地方。拥塞控制方面的内容受益于与 Hari Balakrishnan 和 Robert T.Morris 的讨论，并部分基于 Raj Jain 的思想。

## 4. 容错（第 8 章[在线]）

本章的许多概念和例子最初来源于 Claude Shannon, Edward F. Moore, David Huffman, Edward J. McCluskey, Butler W.Lampson, Daniel P.Siewiorek 以及 Jim N. Gray。

## 5. 事务与一致性（第 9 章[在线]和第 10 章[在线]）

本书中关于交易和一致性的章节随着课程开发了 40 年，得到了许多来源的思想和帮助。版本历史的概念由 Jack Dennis 提出，all-or-nothing 和 before-or-after 原子性的特定形式由 David P.Reed 提出。Jim N.Gray 不仅提出了这两章的许多思想，还提供了广泛的反馈（这不意味着他同意书中的思想，实际上他强烈反对其中某些思想的重要性）。其他有帮助的评论和建议来自 Hari Balakrishnan, Andrew Herbert, Butler W. Lampson, Barbara H. Liskov, Samuel R. Madden, Larray Rudolph, Gerald Jay Sussman 以及 Julie Sussman 等。

## 6. 计算机安全（第 11 章[在线]）

第 11.1 节和第 11.6 节从 Jerome H. Saltzer 和 Michael D. Schroeder 发表在 Proceedings of the IEEE (1975 年 9 月, 第 1278~1308 页) 上论文“The protection of information in computer systems”获取了大量内容。Ronald Rivest, David Mazieres 以及 Robert T.Morris 对本章内容做出了重要贡献。Brad Chen, Michael Ernst, Kevin Fu, Charles Leiserson, Susan Ruff 以及 Seth Teller 对如何改进文字提出了许多建议。

## 7. 建议的外部读物

建议阅读的材料是从多个渠道获得的。特别感谢 Michael D.Schroeder，他发现了计算机科学外的若干经典系统论文，如果不是他的意见，不会有人想到这些论文。感谢 Edward D. Lazowska 提供了在华盛顿大学使用的阅读列表，感谢 Bulter W.Lampson 对列表进行了细心的审阅。

## 8. 练习和问题集

每章末的练习和本书末的问题集主要来自 40 年来本课程的考试和测验题, 经过了许多教师、助教以及学生的收集、建议、试用、查错以及修订。

某些长的练习题和大部分问题集(由背景故事和若干相关问题组成)是由某个人完成的。如果这些问题集不是由作者之一撰写的, 我们会在问题集的开始页用脚注的形式标明其作者。

每个问题或问题集后都有形如“1978-3-14”这样的标识, 说明了该问题首次用于考试时的年份、考试号以及问题序号。

Jerome H. Saltzer

M.Frans Kaashoek

# 目 录

第 1 章 系统 .....	1	2.1 3 种基本抽象 .....	26
1.0 概述 .....	1	2.1.1 存储器 .....	27
1.1 系统和复杂性 .....	2	2.1.2 解释器 .....	32
1.1.1 不同领域中系统的共同 问题 .....	2	2.1.3 通信链路 .....	36
1.1.2 系统、组件、接口和 环境 .....	5	2.2 计算机系统命名 .....	37
1.1.3 复杂性 .....	6	2.2.1 命名模型 .....	38
1.2 复杂性的来源 .....	7	2.2.2 默认上下文引用和 显式上下文引用 .....	41
1.2.1 相互影响的需求 .....	8	2.2.3 路径名、命名网络和 递归名字解析 .....	44
1.2.2 保证高利用率 .....	10	2.2.4 多重查找：在分层的 上下文中搜索 .....	46
1.3 处理复杂性 I .....	11	2.2.5 名字比较 .....	47
1.3.1 模块化 .....	11	2.2.6 名字发现 .....	48
1.3.2 抽象化 .....	13	2.3 用名字和层次结构组织计算机 系统 .....	49
1.3.3 层次化 .....	15	2.3.1 硬件层：总线 .....	51
1.3.4 分级化 .....	15	2.3.2 软件层：文件抽象 .....	55
1.3.5 组合起来：用名字 关联 .....	16	2.4 总结经验，面向未来 .....	58
1.4 求同存异：计算机系统与 其他系统的比较 .....	16	2.5 案例分析：UNIX 文件系统中 的层次和命名 .....	58
1.4.1 计算机系统组合没有 限制 .....	17	2.5.1 UNIX 文件系统应用 程序编程接口 .....	59
1.4.2 d（技术）/dt 是前所 未有的 .....	18	2.5.2 块层 .....	60
1.5 处理复杂性 II .....	21	2.5.3 文件层 .....	61
1.5.1 为什么模块化、 抽象化、层次化和 分级化还不够 .....	21	2.5.4 inode 编号层 .....	62
1.5.2 迭代法 .....	21	2.5.5 文件名层 .....	62
1.5.3 保持简单 .....	23	2.5.6 路径名层 .....	64
1.6 本书其他内容 .....	24	2.5.7 连接 .....	65
习题 .....	25	2.5.8 重命名 .....	66
第 2 章 计算机系统的组成部分 .....	26	2.5.9 绝对路径层 .....	67
2.0 概述 .....	26	2.5.10 符号连接层 .....	68
		2.5.11 实现文件系统 API .....	70
		2.5.12 Shell，隐式上下文，	



搜索路径, 名字发现.... 73	<b>第 4 章 使用客户及服务增强模块化 ... 102</b>
2.5.13 进一步阅读推荐..... 74	4.0 概述 ..... 102
习题 ..... 75	4.1 客户/服务组织方式 ..... 103
<b>第 3 章 命名方案的设计 ..... 77</b>	4.1.1 从软模块化到强制 模块化..... 103
3.0 概述 ..... 77	4.1.2 客户/服务的组织 方式..... 108
3.1 命名方案设计中的考虑因素 .... 77	4.1.3 多客户端和 多服务器..... 114
3.1.1 模块化共享..... 77	4.1.4 可信中间方 ..... 114
3.1.2 元数据与名字重载..... 80	4.1.5 一个简单的例子 服务..... 116
3.1.3 地址: 定位对象的 名字 ..... 82	4.2 客户端和服务端之间的 通信 ..... 117
3.1.4 生成唯一的名字 ..... 83	4.2.1 远程过程调用 (RPC) ..... 117
3.1.5 预期用户与用户友好 的名字 ..... 86	4.2.2 RPC 不等于过程 调用..... 119
3.1.6 名字、值和绑定的 相对寿命 ..... 87	4.2.3 通过中间方的通信 ..... 121
3.1.7 回顾和展望: 名字是 基本的系统组件 ..... 89	4.3 总结及前景 ..... 122
3.2 案例研究: 统一资源 定位器 (URL) ..... 90	4.4 案例研究: 因特网域名 系统 (DNS) ..... 123
3.2.1 网页浏览作为参考 经历; 名字发现 ..... 90	4.4.1 DNS 中的名字解析 ..... 124
3.2.2 URL 的解释..... 90	4.4.2 层次化的名字管理 ..... 128
3.2.3 URL 大小写敏感性..... 91	4.4.3 DNS 的其他特点 ..... 128
3.2.4 部分 URL 的错误 上下文引用 ..... 92	4.4.4 DNS 中的名字发现 ..... 129
3.2.5 URL 中的名字重载..... 94	4.4.5 DNS 响应的可信性 ..... 130
3.3 战争故事: 名字使用中的 病症..... 95	4.5 案例研究: 网络文件 系统 (NFS) ..... 131
3.3.1 名字冲突赶走了 笑容 ..... 95	4.5.1 命名远程的文件和 目录..... 131
3.3.2 来自重载的脆弱名字, 以及市场对策 ..... 95	4.5.2 NFS 的远程过程 调用..... 133
3.3.3 来自重载的更加脆弱 的名字, 伴随市场 崩溃 ..... 96	4.5.3 扩展 UNIX 文件系统 来支持 NFS ..... 136
3.3.4 用户友好的名字中的 大小写敏感性 ..... 97	4.5.4 一致性 ..... 137
3.3.5 电话号码的用尽..... 98	4.5.5 NFS 版本 3 及后续 版本..... 139
习题 ..... 100	习题 ..... 139