

图书基本信息

书名：<<深入理解计算机系统（原书第2版）>>

13位ISBN编号：9787111321330

10位ISBN编号：7111321332

出版时间：2010年

出版时间：机械工业出版社

作者：（美）Randal E.Bryant,David O'Hallaron

页数：702

译者：龚奕利,雷迎春

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## 前言

本书通过程序员的视角来介绍计算机系统，首先把高级语言转换成计算机所能理解的一种中间格式（如汇编语言），然后描述计算机如何解释和执行这些中间格式的程序，系统的哪一部分影响程序的执行效率。

在讲述计算机系统知识的同时，也给出了关于C语言和汇编语言的编程、阅读技巧以及基本的系统编程工具，还给出一些方法帮助程序员基于对计算机系统的理解来改善程序的性能等问题。

本书强调对计算机系统概念的理解，但并不意味着不动手。

如果按照本书的安排做每一章后面的习题，将有助于加深对正文所述概念和知识的理解，更可以从实际动手中学习到新的知识。

本书的主要内容是关于计算机体系结构与编译器和操作系统的交互，包括：数据表示，汇编语言和汇编级计算机体系结构，处理器设计，程序的性能度量和优化，程序的加载器、链接器和编译器，I/O和设备的存储器层次结构，虚拟存储器，外部存储管理，中断、信号和进程控制。

本书的最大优点是程序员描述计算机系统的实现细节，帮助其在大脑中构造一个层次型的计算机系统，从最底层的数据在内存中的表示（如大多数程序员一直陌生或疑惑的浮点数表示），到流水线指令的构成，到虚拟存储器，到编译系统，到动态加载库，到最后的用户态应用。

贯串本书的一条主线是使程序员在设计程序时，能充分意识到计算机系统的重要性，建立起所写程序可能被执行的数据或指令流图，明白执行程序时到底发生了什么事，从而能设计出高效、可移植、健壮的程序，并能够更快地对程序排错、改进程序性能等。

原书是卡内基-梅隆大学（CMU）的教材，现在很多国内外著名的大学也选用其作为教材或辅助性资料，因此，本书的读者不仅仅是那些因为工作和兴趣而关注本书的人，还包括一些在校的大学生。

我们认为，在校学生越早接触本书的内容，将越有利于他们学习计算机的相关课程，培养对计算机系统的研究兴趣。

总的来说，本书是一座桥梁，它帮助程序员衔接了计算机系统各个领域的知识，为程序员构造了一个概念性框架。

要想获取更多关于计算机系统结构、操作系统、编译器、网络、并发编程方面的知识，还需要进一步阅读相关书籍。

## 内容概要

本书从程序员的视角详细阐述计算机系统的本质概念，并展示这些概念如何实实在在地影响应用程序的正确性、性能和实用性。

全书共12章，主要内容包括信息的表示和处理、程序的机器级表示、处理器体系结构、优化程序性能、存储器层次结构、链接、异常控制流、虚拟存储器、系统级i/o、网络编程、并发编程等。

书中提供大量的例子和练习题，并给出部分答案，有助于读者加深对正文所述概念和知识的理解。

本书适合作为高等院校计算机及相关专业本科生、研究生的教材，也可供想要写出更快、更可靠程序的程序员及专业技术人员参考。

## 作者简介

Randal E. Bryant 1973年于密歇根大学（University of Michigan）获得学士学位，随即就读于麻省理工学院（Massachusetts Institute of Technology）的研究生院，并在1981年获计算机博士学位。他在加州理工学院（California Institute of Technology）做了三年助教，从1984年

书籍目录

第1章 计算机系统漫游

第一部分 程序结构和执行

第2章 信息的表示和处理

第3章 程序的机器级表示

第4章 处理器体系结构

第5章 优化程序性能

第6章 存储器层次结构

第二部分 在系统上运行程序

第7章 链接

第8章 异常控制流

第9章 虚拟存储器

第三部分 程序间的交互和通信

第10章 系统级i/o

第11章 网络编程

第12章 并发编程

附录a 错误处理

参考文献

章节摘录

插图：1.9 重要主题在此，总结一下我们旋风式的系统漫游。

这次讨论得出一个很重要的观点，那就是系统不仅仅只是硬件。

系统是硬件和系统软件互相交织的集合体，它们必须共同协作以达到运行应用程序的最终目的。

本书的余下部分会讲述硬件和软件的详细内容，通过了解这些详细内容，你可以写出更快速、更可靠和更安全的程序。

我们在此强调几个贯穿计算机系统所有方面的重要概念作为本章的结束。

我们还会在本书中的多处讨论这些概念的重要性。

1.9.1 并发和并行数字计算机的整个历史中，有两个需求是驱动进步的持续动力：一个是我们想要计算机做得更多，另一个是我们想要计算机运行得更快。

当处理器同时能够做更多事情时，这两个因素都会改进。

我们用的术语并发（concurrency）是一个通用的概念，指一个同时具有多个活动的系统；而术语并行（parallelism）指的是用并发使一个系统运行得更快。

并行可以在计算机系统的多个抽象层次上运用。

在此，我们按照系统层次结构中由高到低的顺序重点强调三个层次。

1.线程级并发构建进程这个抽象，我们能够设计出同时执行多个程序的系统，这就导致了并发。

使用线程，我们甚至能够在进程中执行多个控制流。

从20世纪60年代初期出现时间共享以来，计算机系统中就开始有了对并发执行的支持。

传统意义上，这种并发执行只是模拟出来的，是通过使一台计算机在它正在执行的进程间快速切换的方式实现的，就好像一个杂技演员保持多个球在空中飞舞。

这种并发形式允许多个用户同时与系统交互，例如，当许多人想要从一个Web服务器获取页面时。

它还允许一个用户同时从事多个任务，例如，在一个窗口中开启Web浏览器，在另一窗口中运行字处理器，同时又播放音乐。

在以前，即使处理器必须在多个任务间切换，大多数实际的计算也都是由一个处理器来完成的。

这种配置称为单处理器系统。

媒体关注与评论

“2005年，我开始采用Bryant和O’Hallaron的这本书作为本科生计算机系统课程的教材。现今，这本书仍然是我的计算机系统课程教科书的首选。

”——Mirela Damian，维拉诺瓦大学“本书表述清晰、恰到好处——举重若轻地呈现了那些非常复杂的内容。

”——Ibrahim Matta，波士顿大学“这是一本学习计算机硬件和软件如何‘真正’协同工作的好书，还教会你为什么了解这些知识会使你成为一个更有价值的程序员。

本书还帮你为学习像操作系统和编译器这样的高级课程做好准备。

在本书中，我最喜欢的章节是关于缓存的，当我第一次发现缓存有多重要时，真是难以置信！

”——Vishal Shah，Ask.com总架构师

编辑推荐

《深入理解计算机系统》：计算机科学丛书。



版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>