

## 本书概述

本书由 12 章组成，旨在阐述计算机系统的核心概念。内容概述如下：

- 第 1 章：计算机系统漫游。这一章通过研究“hello, world”这个简单程序的生命周期，介绍计算机系统的主要概念和主题。
- 第 2 章：信息的表示和处理。我们讲述了计算机的算术运算，重点描述了会对程序员有影响的无符号数和数的补码表示的特性。我们考虑数字是如何表示的，以及由此确定对于一个给定的字长，其可能编码值的范围。我们探讨有符号和无符号数字之间类型转换的效果，还阐述算术运算的数学特性。菜鸟级程序员经常很惊奇地了解到(用补码表示的)两个正数的和或者积可能为负。另一方面，补码的算术运算满足很多整数运算的代数特性，因此，编译器可以很安全地把一个常量乘法转化为一系列的移位和加法。我们用 C 语言的位级操作来说明布尔代数的原理和应用。我们从两个方面讲述了 IEEE 标准的浮点格式：一是如何用它来表示数值，一是浮点运算的数学属性。

对计算机的算术运算有深刻的理解是写出可靠程序的关键。比如，程序员和编译器不能用表达式  $(x-y<0)$  来替代  $(x<y)$ ，因为前者可能会产生溢出。甚至也不能用表达式  $(-y<-x)$  来替代，因为在补码表示中负数和正数的范围是不对称的。算术溢出是造成程序错误和安全漏洞的一个常见根源，然而很少有书从程序员的角度来讲述计算机算术运算的特性。

- 第 3 章：程序的机器级表示。我们教读者如何阅读由 C 编译器生成的 x86-64 机器代码。我们说明为不同控制结构(比如条件、循环和开关语句)生成的基本指令模式。我们还讲述过程的实现，包括栈分配、寄存器使用惯例和参数传递。我们讨论不同数据结构(如结构、联合和数组)的分配和访问方式。我们还说明实现整数和浮点数算术运算的指令。我们还以分析程序在机器级的样子作为途径，来理解常见的代码安全漏洞(例如缓冲区溢出)，以及理解程序员、编译器和操作系统可以采取的减轻这些威胁的措施。学习本章的概念能够帮助读者成为更好的程序员，因为你们懂得程序在机器上是如何表示的。另外一个好处就在于读者会对指针有非常全面而具体的理解。
- 第 4 章：处理器体系结构。这一章讲述基本的组合和时序逻辑元素，并展示这些元素如何在数据通路中组合到一起，来执行 x86-64 指令集的一个称为“Y86-64”的简化子集。我们从设计单时钟周期数据通路开始。这个设计概念上非常简单，但是运行速度不会太快。然后我们引入流水线的思想，将处理一条指令所需要的不同步骤实现为独立的阶段。这个设计中，在任何时刻，每个阶段都可以处理不同的指令。我们的五阶段处理器流水线更加实用。本章中处理器设计的控制逻辑是用一种称为 HCL 的简单硬件描述语言来描述的。用 HCL 写的硬件设计能够编译和链接到本书提供的模拟器中，还可以根据这些设计