

计算机系统漫游

计算机系统是由硬件和系统软件组成的，它们共同工作来运行应用程序。虽然系统的具体实现方式随着时间不断变化，但是系统内在的概念却没有改变。所有计算机系统都有相似的硬件和软件组件，它们又执行着相似的功能。一些程序员希望深入了解这些组件是如何工作的以及这些组件是如何影响程序的正确性和性能的，以此来提高自身的技能。本书便是为这些读者而写的。

现在就要开始一次有趣的漫游历程了。如果你全力投身学习本书中的概念，完全理解底层计算机系统以及它对应用程序的影响，那么你会步上成为为数不多的“大牛”的道路。

你将会学习一些实践技巧，比如如何避免由计算机表示数字的方式引起的奇怪的数字错误。你将学会怎样通过一些小窍门来优化自己的 C 代码，以充分利用现代处理器和存储器系统的设计。你将了解编译器是如何实现过程调用的，以及如何利用这些知识来避免缓冲区溢出错误带来的安全漏洞，这些弱点给网络和因特网软件带来了巨大的麻烦。你将学会如何识别和避免链接时那些令人讨厌的错误，它们困扰着普通的程序员。你将学会如何编写自己的 Unix shell、自己的动态存储分配包，甚至于自己的 Web 服务器。你会认识并带来的希望和陷阱，这个主题随着单个芯片上集成了多个处理器核变得越来越重要。

在 Kernighan 和 Ritchie 的关于 C 编程语言的经典教材[61]中，他们通过图 1-1 中所示的 hello 程序来向读者介绍 C。尽管 hello 程序非常简单，但是为了让它实现运行，系统的每个主要组成部分都需要协调工作。从某种意义上来说，本书的目的就是要帮助你了解当你在系统上执行 hello 程序时，系统发生了什么以及为什么会这样。

```
code/intro/hello.c
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("hello, world\n");
6      return 0;
7  }
code/intro/hello.c
```

图 1-1 hello 程序(来源: [60])

我们通过跟踪 hello 程序的生命周期来开始对系统的学习——从它被程序员创建开始，到在系统上运行，输出简单的消息，然后终止。我们将沿着这个程序的生命周期，简要地介绍一些逐步出现的关键概念、专业术语和组成部分。后面的章节将围绕这些内容展开。

1.1 信息就是位 + 上下文

hello 程序的生命周期是从一个源程序(或者说源文件)开始的，即程序员通过编辑器创建并保存的文本文件，文件名是 hello.c。源程序实际上就是一个由值 0 和 1 组成的位(又称为比特)序列，8 个位被组织成一组，称为字节。每个字节表示程序中的某些文本字符。