

代码揭秘

从C/C++的角度探秘计算机系统

左飞 著

- ◎ 一线专家倾心巨献，内外兼修，步步为营。
- ◎ 剥茧抽丝，揭开代码背后鲜为人知的秘密。
- ◎ 化繁为简，还原表象掩盖下的计算机本质。



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

8800 8500 8200 7900 7600 7300 7000 6700 6400 6100 5800

代码揭秘

从C/C++的角度探秘计算机系统

左飞 著

電子工業出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书从程序设计角度出发,以 C/C++为描述语言,以 Visual C++为形式工具,将隐藏在代码背后的关于计算机组成原理、计算机操作系统等方面的机制和知识娓娓道来,不仅让读者知其然,更要让读者知其所以然。并让这些知识再反作用于编程实践,从而帮助读者写出更适合机器优化的高质量代码。揭开代码背后鲜为人知的秘密,具体说来,全书主要讨论了包括计算机底层编码、内存与指针、计算机指令与代码系统、函数调用的机制、多级存储系统、线程与进程的概念以及代码优化等多个方面的话题。

本书既可作为大专院校相关专业师生的教学参考书,也可供计算机及其相关领域的工程技术人员查阅之用,对于普通计算机爱好者,本书也不失为帮助他们理解计算机底层机制的一本深入浅出的计算机读物。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

代码揭秘:从 C/C++的角度探秘计算机系统 / 左飞著. 北京:电子工业出版社, 2009.9
ISBN 978-7-121-09310-4

I. 代… II. 左… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2009) 第 125580 号

责任编辑:葛娜

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:27 字数:544 千字

印 次:2009 年 9 月第 1 次印刷

印 数:4000 册 定 价:56.00 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,联系及邮购电话:(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。

这可不是个序

至少绝不是普通意义上的序，那么普通意义上的序是什么样的呢？遮遮掩掩、欲盖弥彰的说这本书是多么多么的好，内容是多么多么的精彩，作者又都干过哪些多么多么英勇的事情。

这种普普通通的序，毫无疑问，都是很招读者烦的。每个看到这些序的读者的第一反应就是赶紧往后翻。受不了。我记得我还专门写过一篇博客就是来抨击请人写序和给人写序这件事的。真是受不了。

所以，你现在看的这个东西绝对不是序，那它是什么呢？它是一个“推荐”，那么有人会问，“推荐”和“序”到底有什么区别呢？

区别大了。推荐在两个极其重要的方面超越了序：

首先，推荐是赤裸裸的夸奖，绝对没有丝毫的遮遮掩掩；

其次，推荐都比较短，节省时间。

说起来，这部书稿，作者最早是交到我这里的（当时还未全部完成）。我第一眼就喜欢上这本书，心里的真实想法是——“太好了，一本好书又到手了，嘿”。但是，什么事坏就坏在但是上，完全是因为外在的客观因素，导致我只能极为不舍的、满含深情地握着作者的手说“兄弟，没法子了，拿到博文去出吧，那儿还行”。

现在，这本书终于出版了，我的心情是一半欢喜一半遗憾，欢喜的是好书终于出版了，遗憾的是不是我出的。

没留神已经写了这么多了，再多就不符合第二条了。

我用一句话来做个赤裸裸的总结夸奖吧：

这本书用通俗易懂的语言对计算机系统的原理和机制作了明明白白的讲解，并通过逆向思维沿着与通常的编程过程完全相反的方向大踏步向前走，最终走到了一切的起点——你所写下的代码。其目的只有一个，就是让你掌握这样一个本领——写出高效的代码。

陈冰 高级程序员

2008 年度 IT 图书最佳策划编辑奖获得者

《大话设计模式》策划编辑

《电脑使用说明书》作者

《Flash 第一步》作者

谈谈“如何选书”——代序

现在文化产业蓬勃发展，图书出版业也呈现百家争鸣、百花齐放的态势。写书的人多了，书店里的书也多了。读者可选择的空间更大了，但读者淘到一本好书的难度也大了。作者抱怨着写本好书真难啊，读者也对歌似的叫嚷道找到一本好书真不容易啊。在读者正式下决心去买这本书之前，我想从一个读者的角度来和大家谈谈“如何选择一本好书”这个问题。

首先读者应该明确的一个问题是：一本技术类或者科普类的书籍之所以存在，必然有它的道理，毕竟知识本身没有优劣好坏之分，只是看这些知识适不适合你。别人认为好的书对于你来说未必好，这个阶段对你作用大的书未必对你永远都有那么大的作用。本文之前曾在笔者博客上发表过，很多网友都觉得此文之言较为中肯，很具参考意义。唯独这句貌似表述“书籍没有好坏之分，只有适不适合你”之意的句子稍稍引起了一些争议。对此我想说的是，一方面，我的原句是“知识没有优劣好坏”而非“书籍没有好坏”，所以理解上还应对我的话做进一步的思辨。另一方面，这也是一个态度问题。尽管目前图书市场的确存在良莠不齐的现象，但如果你最开始就以一种批判的态度去审视手中的书，那试问即使书中有黄金，恐怕你也都会视而不见了。况且买书的人是你自己，你也当然应该勇于为自己的行为承担责任啊。所以我希望下面的文字能够对正在阅读本书的读者有所启迪，帮助他们擦亮眼睛，选出真正适合自己的好书。

选书还是得先看看书名、前言和目录之类的东西。我曾经写过一本名叫《Visual C++ 数字图像处理开发入门与编程实践》的书，有的读者写信夸奖说“这书挺好，内容通俗易懂，实例丰富详尽”；而有的读者则批评说这书“太令人失望了，内容太简单，没深度”。书名已经点明“开发入门”，这表明这本书就是面向初学者的一本入门书，如果名叫“入门”，内容却讳莫如深，恐怕有点挂羊头卖狗肉的嫌疑了吧。所以买书还是要先看好书名，看看适不适合自己。当然，目前有些书的名字起得就有点让人误解。比如去年有本书叫《梦断代码》，这确实是一本不错的书，不过这名字起得就太有蛊惑力了，乍看上去总以为是讲程序编码的，但其实这本书是讲历史的！所以这时，读者就必须翻看书籍的前言和目录了。很多读者容易忽略前言和目录，这显然是不明智的。作者往往会在前言中对书籍的主要内容、谋篇布局以及读者对象等信息进

行交代，这些信息能够进一步帮助你确定这本书到底适不适合你。

看前言还能获得的一个信息是这本书的参与者有多少。这个非常重要。我就曾在书店翻到一本讲编程的书，书大约有四五百页，但是参与编写书籍的人多达四十余位，真是不禁让人心底发凉。连贯性对于一本书是很重要的，前后风格不一，内容杂糅，这种书想来就让人头疼。四十个厨师炒的一盘菜，这盘菜还能吃不？

另外，如果你想踏踏实实地学好编程，最好不要买国内高校的教材。原因有三。首先，国内高校的教材往往是注重理论，里面的程序最长都不超过 100 行，这种书学来只能用来去考“计算机二级”之类的考试，实际的编程能力很难以这些书为基础来获得提升。其次，国内高校的计算机教授很多都长时间不编代码，实战经验匮乏，难写出紧跟技术发展又兼具实践意义的书籍。最后，大学教授挂名编书，学生实际操刀的例子屡见不鲜，这种书难保质量。又有可能出现相互抄袭的情况。例如，前不久爆出的中国工程院某院士被六名学者联名检举学术腐败的事件，其中一个事项就是院士主编的书涉嫌抄袭等问题。后来院士出面解释说被投诉的部分都是由其他作者执笔撰写的，毕竟一本书涉及的内容和方向较多，不可能所有专业都由一个人搞。可见在“一人主编，多人参编”的机制之下，书籍质量的控制并不那么容易，院士尚且若此，其他人更可见一斑。我在写《Visual C++ 数字图像处理开发入门与编程实践》一书时，起初翻阅了来自国内好几所大学不同的教授写的书，结果发现好多章节严重雷同，某些段落竟然一致到标点符号，真是雷死人了！

除了看书名、前言和目录，还要看看出版社和作者。这是非常重要的，有时一字之差，往往谬之千里。原来听过一则趣闻是这样说的，武侠小说大家金庸封笔已久，很多忠实读者都渴望他老人家能有再次出山的那一天。后来市场上出现了一本赫然印着“金庸新著”的书，读者买回去看完后不禁大呼上当，仔细琢磨才发现竟是作者名叫“金庸新”，而非金庸老爷爷重出江湖之“新”作。作者无疑是质量保证的最根本要素。在 IT 领域，国内也有很多非常优秀的作者，通常的认识是如果一个作者最开始就很负责任，那么他一般都会负责到底；但是如果一个作者最开始就不要脸，那他继续无耻下去也是很正常的。此外，一些大牌出版社还是很注重自身声誉的。出版社在对选题的把握、版面的设计以及印刷质量控制等方面都起着至关重要的作用，所以好的出版社也是优质图书的一项重要保证。

上述原则仅仅是我站在一位普通读者的角度所谈的关于如何选书的一些浅见，如果能对其他读者有所帮助和启发，我将不胜欣喜。

前言

算下来这本书应该是我奉献给读者的第三本书了。一路写下来，自己也感觉非常庆幸，庆幸有许多读者如此地厚爱于我！更高兴地看到广大读者能够从我的书里汲取知识，获得启迪。在计算机图书的创作过程中，我不禁感慨：写一本好书不容易！更何况千里马常有，而伯乐不常有，能够让一本书找到它真正的读者同样有难度。因此我写下了后面的文字，希望这些介绍性的东西能够帮助读者理解本书要旨，明确所述内容。

本书缘起

我想有一个问题，很多读者都非常关心，那就是如何成为一名编程高手。这是以往很多读者向我写信讨教的话题。关于方法学上的问题本书附录中谈了很多，这里不再赘述。但是这里我想告诉读者的是本书能够帮助你做些什么。

通常认为一个计算机程序设计高手应当具备的条件是熟练掌握至少一门计算机程序设计语言，然后有比较扎实的数据结构与算法功底。这样，基本上他已经可以从一种比较高的视角来抽象现实问题并运用计算机来进行模拟和求解了。但是，这其实还不够，一个高效的计算机程序需要“内外兼修”。内功就是程序所使用的数据结构和算法，这是决定程序效率的根本因素；而外功就是程序编码是否符合计算机系统的口味，是否能够最大程度地调动和运用系统的资源。

这一点的作用是不容忽视的。但是由于目前很多程序员都是半路出家，没有接受过系统的计算机科学理论教育，因为他们不知道有这么回事，所以无法给予足够的重视就无可厚非了。就像古代人们不知道有细菌和病毒的存在，所以那时也就没有消毒的概念一样。

另外，我需要指出的是，很多计算机专业科班出身的学生也未必能够领悟这项“外功”的奥义。就目前中国的计算机教育来说，学校的课程设置仅仅是将各项知识独立地对待，这样对于悟性不是非常高的学生来说，在没有被点化的情况下就没有办法有机地将这么多课程串接起来。一个不能形成完整系统性的知识结构是空洞和脆弱的结构。

国外先进的计算机教育已经注意到了这一点，很多国外的大学都开设了这样一门从程序设计角度来让学生真切感受计算机组织机制和原理的课程。庆幸的是，国内很多人也已经意识到

了这个问题，所以相关的课程和有关的书籍也在被逐步引入到国内。但是目前存在的一个问题是，原封不动地将国外的课程和教材搬到国内明显让这洋学问显得有点“水土不服”。

这种水土不服主要表现在三个方面。首先，这些书籍往往都是国外大学的教材，这些教材面向专业学生，这无疑将广大非专业学生和读者拒之门外了。其次，这些书是以国外大学的情况为参照而写作的，没有充分考虑中国的情况——当然，人家干嘛要考虑呀？这就让中国读者学起来非常不顺手。比如，某些例子可能是以 UNIX 或者 Linux 下的编程为基础设计的，中国绝大多数使用 Windows 的读者就很难完成这些实验。这极大地打击了他们继续学习的积极性和后劲。最后，引进版的书明显要比实际技术的发展老几代。现在都是多核时代了，书里可能还在讲奔腾 2，这就让中国的读者只能在后面跟着人家跑，却永远没办法超越。

核心内容

基于上面的考虑我写了这样一本《代码揭秘》。或许很多人对这个名字感到困惑并充满疑问，因为直观上好像不能确定这本书的具体内容是什么。接下来我就要告诉读者这本书到底是写什么的。本书从程序设计角度出发（因为这是广大读者最熟悉、最容易接受的出发点），以 C/C++ 为描述语言（因为这是目前最广为使用的计算机语言，另外，C/C++ 中像指针这样的底层特性也非常适合用来揭示系统深处的东西），以 Visual C++ 为形式工具（因为它也是中国程序员所习惯使用的开发环境），将隐藏在代码背后的关于计算机组成原理、计算机操作系统等方面的原理和知识娓娓道来，不仅让读者知其然，更要让读者知其所以然。并让这些知识再反作用于编程实践，从而帮助读者写出更适合机器优化的高质量代码。揭开代码背后鲜为人知的秘密，让代码开口说话，告诉你一个真实的计算机，从而让你能够写出适合与计算机交流的优秀代码，这就是本书所能为你做到的。

读者对象

接下来我要回答的问题是本书为谁而写。

如果你是一名渴望在编程技艺上有所精进的程序设计爱好者，那么这本书就是助你成为编程高手的制胜法宝；

如果你是一名正苦于无法突破编程瓶颈的程序员，那么这本书就是帮你打通任督二脉的武林秘籍；

如果你是一名感觉书本知识仍然无法内化的计算机专业学生，那么本书就是替你拨云开雾、指点迷津的通天灯塔。

阅读建议

最后，我还希望和读者谈谈阅读及学习本书的建议。

第一，从整体上看全书共分9章，章节之间相互关联、层层递进，本着循序渐进的原则逐渐展开，因此阅读本书时切不可跳跃式地选读，这样非但不能领悟整个体系间的精髓，更会为进一步的阅读带来困难。特别是第2~7章尤不可拆分阅读，务必按其顺序来学习。

第二，为了帮助读者理解，书中绘制了大量图表。这些图表的作用不可小视，如果在文字理解上遇到困难，可以参照图表来学习。为了帮助读者加深理解，书中还设计了大量的实验，尽管书中给出了实验结果，但也请读者务必亲自动手实践一下这些例子，这样才能让知识凝固在你的脑中。

第三，本书所涉及的知识面比较广，在某些时候限于篇幅的考虑而未能把所有问题都展开来深入探究。对于那些并不是非常深入的知识点，如果读者有兴趣，建议读者多多查阅有关方面的资料，这样学习才能更加博文通识。

关于本书

排除前期构思和素材准备的时间，本书的写作时长也将近有一年之久。在这个过程中，笔者力求精益求精，对很多知识点再三推敲，并翻阅了大量资料和文献。这些工作和努力无非是希望能够写一本对得起读者的书。

在早先的两本书出版之后，我也收到了许多热心读者的来信，一部分人表达了我作品的充分肯定，这当然是我所乐于见到的，能够获得来自读者的认可对于一个负责任的作者来说无疑是莫大的荣耀。而更多的人则向我询问了书中的一些问题，主要是他们在阅读过程中遇到的困难。这也是我非常高兴见到的事情，因为这让我真切地感知到确实有很多人在读我写的书。很多读者同问的一个问题是如何能够学好编程。为了回答这个问题，我特别撰写了一篇题为《浅谈编程能力的培养与提高》的文章附在本书正文之后，希望对那些仍处在迷茫之中的读者能够起到一定的帮助作用。

写一本好书真的很难，写一本没有错误的好书更是难如登天。我闻听计算机科学大师 Knuth 在《计算机程序设计艺术》丛书出版后也提出如果谁能够从他书中找到一个错误，就能够获得256美分的奖励，事实证明获得这项奖励的人还是大有人在的。这也客观地说明了即使作者很牛，即使作者很用心，书中出现纰漏和欠缺也的确是在所难免的事情，所以我也真心地希望广大读者能够不吝赐教和批评。不过以往的经验着实有点让我失望。给我写信的读者大有人在，但是指出我书中错误的读者却寥寥无几。我想这其中的一个非常重要的原因在于中国的教育更多的是让我们学会一味接受，而非大胆地思辨！我们太过盲从而迷信权威，只会聆听高人教诲，却不想自己思考。这对于学习来说不是一个好现象。古人云：尽信书则不如无书。我也希望广大读者能够以思辨的态度来看书，这样你将会获得更多。如果在这个过程中，读者有什么感想或者问题希望和作者交流的，欢迎给我写信，联系信箱：fzuo@yahoo.cn。

最后诚挚地祝愿每位读者都能够真正感受到编程的快乐，并在编程的路上行得更稳健，走得更长远！

目 录

第 1 章 绪论	1
1.1 计算机系统初探	2
1.1.1 换个角度看计算机	2
1.1.2 CPU 很好很强大	4
1.2 计算机语言与编译技术	8
1.2.1 如何让机器理解你	8
1.2.2 编译技术与开发环境	14
1.2.3 程序开发流程	20
1.3 C 语言神话	22
1.3.1 C 语言的历史	22
1.3.2 简单说说 C 语言的特点	23
1.4 奇迹的延续	25
1.4.1 C++ 的产生与发展	25
1.4.2 C++ 与面向对象思想	27
1.5 本章小结	31
第 2 章 揭开数据表示的面纱	32
2.1 进制系统	33
2.1.1 最简单的计数方式	33
2.1.2 计算机里只有黑白	34
2.1.3 压缩表示的二进制	36

2.2 位与位操作	38
2.2.1 计算机存储的单位	38
2.2.2 位操作和位段	39
2.3 计算机中的数值	48
2.3.1 一种最简单的数	48
2.3.2 现实世界需要负数	49
2.3.3 只有整数还不够	51
2.4 让计算机学会写字	56
2.4.1 ASCII 码	57
2.4.2 汉字编码	59
2.4.3 更强大的编码	60
2.5 C 语言基本数据类型	62
2.5.1 整型	63
2.5.2 字符型	66
2.5.3 注意浮点数陷阱	67
2.6 本章小结	71

第 3 章 变量与地址

3.1 程序都在内存中	73
3.2 很多初学者都怕指针	77
3.3 睁大眼睛看内存	81
3.4 数组与指针是近亲	84
3.5 再谈指针	93
3.6 本章小结	97

第 4 章 动态内存管理

4.1 malloc 和 free	99
4.2 sizeof 并不复杂	101
4.3 内存操作函数	105
4.3.1 memset	105
4.3.2 memcpy 和 memmove	107

4.4 new 和 delete.....	109
4.5 内存错误面面观.....	113
4.5.1 最怕内存泄漏.....	113
4.5.2 小心重复释放.....	115
4.5.3 指针是个坏东西.....	117
4.5.4 超量写内存.....	118
4.6 使用 Visual C++ 检查内存泄漏.....	120
4.7 本章小结.....	124
第 5 章 代码与指令系统.....	125
5.1 还原代码的本来面目.....	126
5.1.1 内存中的代码.....	126
5.1.2 指向函数的指针.....	131
5.1.3 CPU 的存储器.....	136
5.1.4 寄存器变量.....	143
5.1.5 寄存器组举例.....	145
5.2 指令——简单 or 复杂.....	149
5.2.1 指令格式.....	149
5.2.2 操作类型.....	154
5.2.3 操作数类型.....	155
5.3 如何找到地址.....	156
5.3.1 下一步该做什么.....	156
5.3.2 计算机知道自己需要什么.....	165
5.4 本章小结.....	172
第 6 章 函数与函数调用.....	173
6.1 函数与参数.....	174
6.1.1 C/C++ 中的函数.....	174
6.1.2 参数传递.....	174
6.1.3 作用域.....	180
6.2 函数的递归调用.....	184

6.2.1 到处都是递归	184
6.2.2 小心使用递归	188
6.2.3 递归与非递归	194
6.2.4 内、外部变量分配原理	198
6.3 内存的使用	200
6.3.1 活动记录与栈	200
6.3.2 静态分配	203
6.3.3 有静就有动	205
6.4 程序在内存中的模样	217
6.5 本章小结	219
第 7 章 多级存储系统	221
7.1 存储系统及层级结构	222
7.1.1 存储器分类	222
7.1.2 存储器的层级结构	225
7.1.3 访问的局部性原理	227
7.1.4 再谈存储器的层级	231
7.2 高速缓存	235
7.2.1 缓存设计策略	235
7.2.2 多级缓存原理	245
7.2.3 实际编码指导	250
7.3 虚拟内存	258
7.3.1 何为虚拟内存	259
7.3.2 虚拟地址	262
7.3.3 页面请求与磁盘缓冲	269
7.3.4 工作集合与系统颠簸	274
7.3.5 虚拟内存与性能影响	276
7.4 本章小结	278
第 8 章 操作系统交互	279
8.1 多任务	280

8.1.1	串行与并行	280
8.1.2	多任务的实现	282
8.1.3	并发程序设计	283
8.2	进程	284
8.2.1	进程的概念	284
8.2.2	进程的状态	285
8.2.3	进程控制块	288
8.3	Win32 进程编程	289
8.3.1	创建进程	289
8.3.2	环境变量	299
8.4	线程	301
8.4.1	线程的概念	301
8.4.2	多线程	302
8.4.3	超线程	303
8.4.4	线程池	304
8.5	调度	305
8.5.1	处理器的调度	305
8.5.2	调度算法准则	306
8.5.3	常见的调度算法	308
8.6	Win32 线程编程	313
8.6.1	创建和退出线程	313
8.6.2	挂起和恢复线程	317
8.6.3	远程线程的注入	320
8.7	本章小结	329
第 9 章	瓶颈与优化	330
9.1	优化还是不优化	331
9.2	测量与分析的内容	333
9.3	测量与分析的方法	334
9.3.1	使用计时器	334
9.3.2	使用 Profile	341

9.3.3 使用性能监视器	349
9.4 基本规律	350
9.4.1 二八法则	350
9.4.2 安达尔定律	351
9.5 程序优化路线	354
9.5.1 优化实践的经典案例	354
9.5.2 优化案例的启示	355
9.6 编译器不是万能的	357
9.7 实际优化建议	360
9.7.1 循环条件中的低效	361
9.7.2 注意字符串的操作	363
9.7.3 权衡函数调用需求	364
9.7.4 转换指针形式代码	366
9.7.5 检查存储器的访问	367
9.7.6 使用循环展开技术	372
9.7.7 查表替换复杂运算	375
9.7.8 耗时计算移出循环	377
9.8 本章小结	378
附录 A 浅谈编程能力的培养与提高	379
附录 B 程序人生	397
参考文献	414

第1章 绪论

——知己知彼，百战不殆。

如何才能编写出高效的计算机程序呢？编写一个高效的计算机程序需要两个条件。首先，针对具体的问题，需要选择一组最好的算法和数据结构，好的算法与合适的数据结构能够保证程序本身采取最直接、最有效的方法去求得问题的答案。但仅有好的算法和数据结构还不足以使程序运行的效率发挥到极致！编写高效程序的另外一个条件就是编写出来的代码必须是适合于编译器进行有效优化的代码。要理解编译器的行为，并合乎它的胃口并非易事，但也绝非难事。如果对计算机系统原理和编译过程能做到胸有成竹、如数家珍，那么理解编译器的行为也就不是什么难事了，这样写出高效的代码自然也是水到渠成。本章作为全书的导引，首先给出一些基本的概念和原理，以为读者的后续学习奠定基础。

1.1 计算机系统初探

孙子兵法中言：知己知彼，百战不殆。为了写出高效的代码，最大程度地提高程序运行效率，我们必须更好地理解计算机这样一个看似复杂的系统。本节将给出一些关于计算机系统的基本的概念，其中提出的一些问题或者技术细节都是本书后续内容中要讨论的重点。

1.1.1 换个角度看计算机

之所以要探究计算机系统的内部结构和工作原理，是为了能够准确地把握代码指令的传递和处理流程，找出限制和降低程序运行效率的关键点所在，从而让程序员能够跟随计算机一同思考，并编写出高效、可靠的代码。要了解计算机系统的内部结构和工作原理并不容易，它至少涉及三门独立的科目：计算机组成原理、计算机操作系统和编译原理。这些都是计算机专业的学生必修的主要课程。但即使将它们逐个学完，若不能加以贯穿和联系，那么一个系统化的知识框架就没办法成功地搭建起来，于是学习成果和功效就会被打折！从程序设计和代码编写的角度将上述知识有机地串接起来是一个不错的主意。

作为一个程序员或者开发人员来说，完全掌握上述科目的知识也实在是一件劳心费神的事，何况似乎掌握了它们也不见对程序开发起到什么明显的推进作用。那这个矛盾该如何解决呢？很简单，就是以程序员最熟悉的方式来诠释这些基础的理论知识。如果能从简单的代码编写出发，思考编译器的工作机制，进而探究计算机系统的工作原理和组织结构，这是一件多么有趣的事情啊！当程序员通过代码编写这一主线将整个流程走完时，一方面，他们可以对计算机系统有一个深入的理解；另一方面，这种理解将反作用于程序设计实践，提高程序员的编码能力，使他们足以写出漂亮、高效的代码。这就是我们所希望做到的。帮助读者走完这一流程的第一步就是先向读者介绍一些基本的概念和必要的准备知识，其中最基础的就是对计算机系统有一个概括的认知。这当然不同于一般的计算机文化基础或者计算机入门知识，我们假设读者都不是“电脑盲”。所谓的对于计算机系统的概括的认知，更重要的是侧重于计算机中指令的传递过程，也就是跟程序运行直接相关的硬件设备，诸如外设之类的硬件则不在我们考虑的范围之内。

首先想请问大家一个问题，什么是电子计算机呢？大家一定认为这是一个非常简单而容易的问题，然而尽管大家经常使用计算机，但要给出电子计算机的准确定义也未必像想象中那么容易。电子计算机是根据程序化的指令来执行具体任务，集合输入、处理、存储和输出功能于一身的一种电子机器。