

Broadview®
WWW.BROADVIEW.COM.CN

编程卓越之道

WRITE GREAT CODE

第一卷：

Volume 1:

深入理解计算机

Understanding the Machine



[美] Randall Hyde

韩东海

译 著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



NO STARCH
PRESS™

WRITE GREAT CODE

Volume 1: Understanding the Machine

编程卓越之道

第一卷：深入理解计算机

[美] Randall Hyde 著

韩东海 译

电子工业出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

各位程序员一定希望自己编写的代码是能让老板赞赏、满意的代码；是能让客户乐意掏钱购买的代码；是能让使用者顺利使用的代码；是能让同行欣赏赞誉的代码；是能让你自己引以为豪的卓越代码。

本书作者为希望能编写出卓越代码的人提供了自己积累的关于卓越编程的真知灼见。它弥补了计算机科学和工程课程中被忽略的一个部分——底层细节，而这正是构建卓越代码的基石。具体内容包括：计算机数据表示法，二进制数学运算与位运算，内存组织与内存访问，数据类型及其表示，布尔逻辑与数字设计，CPU 体系结构，CPU 指令集的体系结构，内存体系与内存组织，计算机系统如何与外界通信等。

Copyright©2004 by Randall Hyde. Title of English Language original: *Write Great Code*, ISBN 1-59327-003-8.

Simplified Chinese Language edition copyright©2006 by Publishing House of Electronics Industry.

本书中文简体专有翻译出版权由美国 No Starch Press, Inc. 授予电子工业出版社。该专有出版权受法律保护。版权贸易合同登记号：图字：01-2004-2712

图书在版编目 (CIP) 数据

编程卓越之道. 第一卷, 深入理解计算机 / (美) 海德 (Hyde,R.) 著; 韩东海译. —北京: 电子工业出版社, 2006.4

书名原文: WRITE GREAT CODE:Volume 1:Understanding the Machine

ISBN 7-121-02404-7

I. 编... II. ①海...②韩... III. 程序设计 IV. TP311.1

中国版本图书馆 CIP 数据核字 (2006) 第 022128 号

责任编辑: 周 筠

印 刷: 北京智力达印刷有限公司

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

经 销: 各地新华书店

开 本: 787×980 1/16 印张: 29.25 字数: 550 千字

印 次: 2006 年 4 月第 1 次印刷

定 价: 49.80 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077。质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

译者序

探究事物的本质是人类的天性，小时候可能是拆卸玩具、钟表，长大了可能是分析某个工具，那种了解了“幕后”原理的感觉实在是令人非常激动的。计算机差不多可以算是人类最重要的发明了，了解计算机的工作原理，了解那些有趣的应用背后的秘密，应该是每个用户、每个程序员都会感兴趣的。尤其对于程序员们来说，计算机就是我们最有趣的玩具、也是最有用的工具，但是，与很多工具不同的是，即使你拆开计算机的机箱，看到的也是现代集成电路技术的成果，想看到并了解真正的电路？嗯……，可能性是有的，但是，很难。

不过没有关系，我们有人类进步的阶梯：书籍，本书就是为了让读者不仅要知其然（程序做了什么），还要知其所以然（计算机做了什么）而写的。其实，这些内容在计算机体系结构相关的书籍中都有介绍，经典也已经不少，但是本书的视角比较独特，它将计算机科学的几门课程的内容加以融合，从程序员的角度进行阐述。本书作者在汇编领域精研多年，他将自己丰富的实际编程经验融入本书，对于广大读者来说，应该是颇有些益处的。本书可能还算不上经典，但是它是如此独特，同类的书籍不能说没有，但是实在是不多。对于程序员来说，阅读本书将有助于写出更高质量的代码（作者已经对此进行了充分的阐述，我就不再重复了）；即使你不是程序员，阅读本书也有助于你从一个更有趣的角度了解计算机。

翻译本书于我而言也是一次有益有趣的经历，有益在于温故知新；有趣呢，就是在翻译过程中有些术语的选择曾让我颇费踌躇，唉，憋出了不少白头发。作为一本技术书籍，在翻译中不敢追求雅，但求信、达，暂时我还相信我做到了这一点，如果哪位读者觉得有些翻译值得商榷，请不吝赐教，我的邮件地址是：handonghai@yahoo.com.cn。

最后，请允许我在此表达对几位编辑的谢意：陈元玉，陈兴璐，魏泉（排名不分先后左右^_^），辛苦了！

韩东海

2006年3月15日

致 谢

尽管只有作者的名字被印在了本书的封面上，但是您手上的这本书可不是一个人的工作成果。本书的创作是团队协作的成果，在此，我要感谢那些为保证本书的品质做出贡献的人。

Mary Philips，我的好朋友，她帮助我校对了初稿的一些章节。

Bill Pollock，他通读了本书的第 1 到第 6 章，并给出了很多建议。

Karol Jurado，我的编辑，她引导了整个项目从概念到成品的整个过程。

Hillel Heinstein，策划编辑，他保证了本书一直保持正确的主线和简洁的文字。

Andy Carroll，文字编辑，他帮助我改进了本书的文字。

Mark de Wever，技术审阅，他发现了大量的输入错误和技术问题，保证了本书素材的准确性。

Riley Hoffman，他负责排版等繁杂事务，正是他的工作保证了本书（及其目录）的可读性。

Stephanie Provines，她在校对过程中发现了一些印刷及排版错误。

Leigh Sacks，她为本书以及我的前一本书《The Art of Assembly Language》（译注：中文影印版为《汇编语言艺术》，中文版为《汇编语言编程艺术》），做的营销工作真是棒了，当然还要感谢 No Starch 出版所的所有人，他们从一开始就给予这个项目巨大的支持。

最后，重要的是，我要感谢我的妻子，**Mandy**，她恩准我从家务中甩手而出，正因此我才能完成这本书。

多谢各位

Randall Hyde

编程卓越之道 第一卷：深入理解计算机

目录一览

第 1 章 编写卓越代码须知 1	第 2 章 数值表示 9
第 3 章 二进制算术与位运算 39	第 4 章 浮点表示法 65
第 5 章 字符表示法 103	第 6 章 内存组织与访问 133
第 7 章 复合数据类型与内存对象 161	第 8 章 布尔逻辑与数字设计 191
第 9 章 CPU 体系结构 225	第 10 章 指令集体系结构 259

第 11 章
内存体系结构与组织
295

第 12 章
输入与输出 (I/O)
329

运用底层语言思想
编写高级语言代码
405

附录 A
ASCII 字符集
407

索引
411

目 录

第 1 章 编写卓越代码须知	1
1.1 编程卓越之道系列.....	1
1.2 本卷内容.....	3
1.3 本卷所做的假设.....	5
1.4 卓越代码的各项特征.....	6
1.5 本卷涉及的环境.....	7
1.6 获取更多信息.....	8
第 2 章 数值表示	9
2.1 什么是数.....	10
2.2 计数系统 (Numbering System)	11
2.2.1 十进制位值计数系统.....	11
2.2.2 进制 (基数)	12
2.2.3 二进制计数系统.....	13
2.2.4 十六进制计数系统.....	15
2.2.5 八进制 (基数为 8) 计数系统.....	18
2.3 数/字符串转换	19
2.4 数的内部表示.....	21
2.4.1 位 (bits)	21
2.4.2 位串.....	22
2.5 有符号数与无符号数.....	24
2.6 二进制数一些有用的特性.....	25
2.7 符号扩展, 零扩展, 以及缩减.....	27
2.8 饱和操作 (saturation)	30
2.9 二进制编码的十进制 (BCD) 表示法	31
2.10 定点表示法.....	33
2.11 比例数格式 (scaled numeric formats)	35
2.12 有理数表示法.....	38

2.13 获取更多信息.....	38
第 3 章 二进制算术与位运算.....	39
3.1 二进制数与十六进制数的算术运算.....	39
3.1.1 二进制加法.....	40
3.1.2 二进制减法.....	41
3.1.3 二进制乘法.....	42
3.1.4 二进制除法.....	43
3.2 位逻辑运算.....	46
3.3 二进制数和位串 (bit string) 的逻辑运算.....	47
3.4 有用的位运算.....	48
3.4.1 使用与运算检测位串的各个位.....	48
3.4.2 使用与运算来检测一组位是零/非零.....	49
3.4.3 比较一个位串中的一组位.....	49
3.4.4 使用逻辑与创建模-n 计数器 (Modulo-n Counters).....	51
3.5 移位 (Shift) 与循环移位 (Rotate).....	52
3.6 位域与打包 (packed) 数据.....	55
3.7 打包与解包数据.....	60
3.8 获取更多信息.....	64
第 4 章 浮点表示法.....	65
4.1 浮点运算简介.....	66
4.2 IEEE 浮点数格式.....	71
4.2.1 单精度浮点格式.....	72
4.2.2 双精度浮点格式.....	74
4.2.3 扩展精度浮点格式.....	74
4.3 规格化 (normalization) 与反向规格化 (denormalized) 数.....	75
4.4 舍入 (rounding).....	77
4.5 特殊的浮点数.....	78
4.6 浮点异常.....	79
4.7 浮点运算.....	80
4.7.1 浮点表示.....	80
4.7.2 浮点加法与减法.....	81
4.7.3 浮点乘法与除法.....	92
4.8 获取更多信息.....	100
第 5 章 字符表示法.....	103
5.1 字符数据.....	104

5.1.1	ASCII 字符集	104
5.1.2	EBCDIC 字符集	107
5.1.3	双字节字符集	108
5.1.4	Unicode 字符集	109
5.2	字符串	110
5.2.1	字符串格式	111
5.2.2	字符串类型：静态，伪动态，以及动态字符串	116
5.2.3	字符串引用计数	117
5.2.4	Delphi/Kylix 字符串	118
5.2.5	创建你自己的字符串格式	119
5.3	字符集合	119
5.3.1	字符集合的幂集表示法	120
5.3.2	字符集合的列表表示法	120
5.4	设计你自己的字符集	121
5.4.1	设计一种高效的字符集	122
5.4.2	为数字分组字符码	124
5.4.3	分组字母字符	124
5.4.4	比较字母字符	126
5.4.5	其他字符分组	128
5.5	获取更多信息	131
第 6 章	内存组织与访问	133
6.1	基本的系统组成部分	134
6.1.1	系统总线	134
6.1.2	地址总线	135
6.1.3	控制总线	136
6.2	内存物理组织	137
6.2.1	位地址总线	139
6.2.2	16 位数据总线	140
6.2.3	32 位数据总线	142
6.2.4	64 位总线	143
6.2.5	在非 80x86 处理器上访问小数据单位	143
6.3	大端组织与小端组织	144
6.4	系统时钟	149
6.4.1	内存访问与系统时钟	151
6.4.2	等待状态	152

6.4.3	高速缓存内存.....	153
6.5	CPU 内存访问.....	157
6.5.1	直接内存寻址模式.....	158
6.5.2	间接寻址模式.....	158
6.5.3	变址寻址模式.....	159
6.5.4	比例变址寻址模式.....	160
6.6	获取更多信息.....	160
第 7 章	复合数据类型与内存对象.....	161
7.1	指针类型.....	162
7.1.1	指针的实现.....	163
7.1.2	指针与动态内存分配.....	164
7.1.3	指针操作与指针运算.....	164
7.2	数组.....	169
7.2.1	数组声明.....	169
7.2.2	数组在内存中的表示.....	172
7.2.3	访问数组元素.....	173
7.2.4	多维数组.....	174
7.3	记录/结构.....	181
7.3.1	Pascal/Delphi 中的记录.....	181
7.3.2	C/C++中的记录.....	182
7.3.3	HLA 中的记录.....	182
7.3.4	记录的内存存储.....	183
7.4	判别式联合.....	185
7.4.1	C/C++中的联合.....	186
7.4.2	Pascal/Delphi/Kylix 中的联合.....	186
7.4.3	HLA 中的联合.....	187
7.4.4	联合的内存存储.....	187
7.4.5	联合的其他用途.....	188
7.5	获取更多信息.....	189
第 8 章	布尔逻辑与数字设计.....	191
8.1	布尔代数.....	192
8.1.1	布尔运算符.....	192
8.1.2	布尔代数的公理.....	192
8.1.3	布尔运算符优先级.....	194
8.2	布尔函数与真值表.....	194

8.3	函数号.....	197
8.4	布尔表达式的代数运算.....	198
8.5	标准型.....	199
8.5.1	最小项之和标准型与真值表.....	200
8.5.2	使用代数方法得到最小项之和标准型.....	202
8.5.3	最大项之积标准型.....	203
8.6	布尔函数化简.....	204
8.7	但是, 这些和计算机又有什么关系呢.....	212
8.7.1	电子线路与布尔函数的对应.....	213
8.7.2	组合电路.....	214
8.7.3	时序与钟控逻辑 (Sequential and Clocked Logic).....	220
8.8	获取更多信息.....	224
第 9 章	CPU 体系结构.....	225
9.1	CPU 设计基础.....	225
9.2	指令解码与执行: 随机逻辑与微码.....	228
9.3	指令执行详解.....	229
9.3.1	mov 指令.....	230
9.3.2	add 指令.....	232
9.3.3	jnz 指令.....	234
9.3.4	loop 指令.....	234
9.4	并行——提高处理速度的关键.....	235
9.4.1	预取队列.....	238
9.4.2	妨碍预取队列性能的情况.....	242
9.4.3	流水线操作——重叠执行多条指令.....	243
9.4.4	指令高速缓存——提供访问内存的多条通路.....	247
9.4.5	流水线相关 (pipeline hazards).....	249
9.4.6	超标量运算——并行执行指令.....	251
9.4.7	乱序执行 (Out-of-Order Execution).....	253
9.4.8	寄存器重命名.....	253
9.4.9	甚长指令字 (VLIW) 体系结构.....	255
9.4.10	并行处理.....	255
9.4.11	多处理.....	257
9.5	获取更多信息.....	258
第 10 章	指令集体系结构.....	259
10.1	指令集设计的重要性.....	260

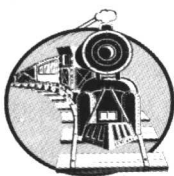
10.2 指令设计基本目标.....	261
10.2.1 选择指令长度.....	263
10.2.2 规划未来.....	265
10.2.3 选择指令.....	266
10.2.4 给指令指派操作码.....	266
10.3 Y86 假想处理器.....	267
10.3.1 Y86 的限制.....	268
10.3.2 Y86 指令.....	268
10.3.3 Y86 的寻址模式.....	270
10.3.4 Y86 指令编码.....	271
10.3.5 Y86 指令编码举例.....	274
10.3.6 扩展 Y86 指令集.....	278
10.4 80x86 指令编码.....	279
10.4.1 编码指令操作码.....	281
10.4.2 add 指令编码的例子.....	287
10.4.3 编码立即操作数.....	291
10.4.4 8, 16 与 32 位操作数编码.....	292
10.4.5 指令的替代编码 (alternate encoding).....	292
10.5 指令集设计对程序员的意义.....	293
10.6 获取更多信息.....	293
第 11 章 内存体系结构与组织.....	295
11.1 内存层次结构.....	295
11.2 内存层次结构是如何工作的.....	298
11.3 内存子系统的相对性能.....	300
11.4 高速缓存体系结构.....	302
11.4.1 直接映射高速缓存.....	303
11.4.2 全相联高速缓存.....	304
11.4.3 n 路组相联高速缓存.....	304
11.4.4 高速缓存方案与数据访问类型的匹配.....	305
11.4.5 缓存线替换策略.....	306
11.4.6 写数据到内存中.....	307
11.4.7 高速缓存使用与软件.....	308
11.5 虚存, 保护, 以及页面调度.....	309
11.6 颠簸.....	312
11.7 NUMA 与外围设备.....	313

11.8	编写理解内存层次结构的软件.....	314
11.9	运行时内存组织.....	316
11.9.1	静态与动态对象, 绑定, 以及生命期.....	317
11.9.2	代码, 只读, 以及常量段.....	319
11.9.3	静态变量段.....	319
11.9.4	未初始化存储 (BSS) 段.....	319
11.9.5	栈段.....	320
11.9.6	堆段与动态内存分配.....	321
11.10	获取更多信息.....	328
第 12 章	输入与输出 (I/O)	329
12.1	将 CPU 与外界相连.....	330
12.2	将端口连接到系统的其他方式.....	333
12.3	I/O 机制.....	334
12.3.1	内存映射输入输出.....	334
12.3.2	输入输出与高速缓存.....	335
12.3.3	I/O 映射输入/输出.....	335
12.3.4	直接内存访问 (DMA)	336
12.4	输入输出速度等级.....	337
12.5	系统总线与数据传输率.....	338
12.5.1	PCI 总线的性能.....	339
12.5.2	ISA 总线的性能.....	340
12.5.3	AGP 总线.....	341
12.6	缓冲.....	341
12.7	握手.....	342
12.8	I/O 端口的超时.....	343
12.9	中断与轮询方式 I/O	344
12.10	保护模式操作与设备驱动程序.....	345
12.10.1	设备驱动程序 (Device Drivers)	346
12.10.2	与设备驱动程序以及“文件”通信.....	347
12.11	深入研究各种 PC 外设.....	347
12.12	键盘.....	348
12.13	标准 PC 并口.....	349
12.14	串口.....	351
12.15	磁盘驱动器.....	352
12.15.1	软盘驱动器.....	352

12.15.2	硬盘驱动器.....	352
12.15.3	RAID 系统.....	358
12.15.4	Zip 与其他光读软盘驱动器.....	359
12.15.5	光驱 (optical drive)	359
12.15.6	CD-ROM, CD-R, CD-R/W, DVD, DVD-R, DVD-RAM 与 DVD-R/W 驱动器.....	360
12.16	磁带驱动器.....	362
12.17	闪存.....	363
12.18	RAM 盘与半导体盘.....	365
12.19	SCSI 设备与控制器.....	367
12.20	IDE/ATA 接口.....	372
12.21	大容量存储设备上的文件系统.....	374
12.21.1	使用空闲空间位图 (bitmap) 管理文件.....	377
12.21.2	文件分配表.....	378
12.21.3	块表文件组织.....	381
12.22	编写处理大容量存储设备上的数据的软件.....	385
12.22.1	文件访问性能.....	386
12.22.2	同步与异步 I/O.....	387
12.22.3	I/O 类型的影响.....	388
12.22.4	内存映射文件.....	389
12.23	通用串行总线 (USB)	390
12.23.1	USB 的设计.....	390
12.23.2	USB 的性能.....	392
12.23.3	USB 传输的类型.....	393
12.23.4	USB 设备驱动程序.....	395
12.24	鼠标, 触控板与其他指点设备.....	396
12.25	操纵杆与游戏控制器.....	397
12.26	声卡.....	399
12.26.1	音频接口外设如何产生声音.....	400
12.26.2	音频与 MIDI 文件格式.....	401
12.26.3	编程处理音频设备.....	403
12.27	获取更多信息.....	403
	运用底层语言思想编写高级语言代码.....	405
	附录 A ASCII 字符集.....	407
	索引.....	411

1

WHAT YOU NEED TO KNOW TO WRITE GREAT CODE 编写卓越代码须知



《编程卓越之道》将教你写出让你引以为豪的代码；能够给同行留下深刻印象的代码；能够让客户满意、乐于使用的代码；能够让人们（客户，你的老板，等等）不惜花大价钱来获得的代码。简而言之，《编程卓越之道》系列将论述如何编写出卓越超凡，让其他程序员敬畏的软件。

1.1 编程卓越之道系列

《编程卓越之道：深入理解计算机》是《编程卓越之道》系列四卷本中的第一本。要编写卓越的代码，需要综合知识、经验以及程序员在多年的求索和从错误中学习之后才能

获得的技能。编写这个系列的目的是为了与各位新老程序员们分享几十年来有价值的看法和经验。我希望这几本书能够对缩短学习时间有所帮助，并且能够减少“刻苦”学习所带来的挫折感。

本书是系列中的第1卷，名为《深入理解计算机》，其目的在于弥补典型的计算机科学或者计算机工程的课程语焉不详的一些底层细节。本卷的内容是构筑卓越软件的基石，不了解这些内容是无法编写高效代码的，而很多问题的解决也需要对这个主题有着深入的掌握。虽然我试图让各卷之间尽量独立，但掌握《深入理解计算机》仍可被认为是阅读后续各卷的先决条件。

第2卷名为《运用底层语言思想编写高级语言代码（Thinking Low-Level, Writing High-Level）》会将第1卷中介绍的知识立即应用到实践中去，《运用底层语言思想编写高级语言代码》将会讲述如何分析用高级程序设计语言编写的代码，来判断编译器为之产生的机器码的质量。经过上述知识的武装，你就可以用高级语言编写效率与直接手写汇编程序非常接近的程序。高级语言程序员常有的错觉是认为不管程序员写出怎样的程序，优化编译器总能产生最优化的机器码，这是完全错误的，你在源文件中选用的语句和数据结构都会对编译器最终产生的机器码的效率有很大影响。通过讲述如何分析编译器产生的机器码，《运用底层语言思想编写高级语言代码》将教会你写出高效的代码而不必求诸汇编语言。

除了高效，卓越代码还有其他很多特征，本系列的第3卷《Engineering Software》，将覆盖其中的一部分。《Engineering Software》将讨论如何编写让别人更加容易理解和维护的代码，以及如何提高程序员的生产率，以及如何在不承担诸多软件工程书籍所提到的“忙碌的工作”的情况下，提高程序的生产率。《Engineering Software》将讨论如何编写能让你与你共事的程序员会乐于看到的代码，而不是那种会让别人在你背后用挑剔的话语来讨论你能力的代码。

卓越的代码是要工作的，因此，如果不加上一卷来讨论测试、调试以及质保的话，就是我的疏忽了。不管你对软件测试的态度是害怕还是讨厌，或者你认为那不过是只有初级程序员才需要做的事情，一个非常普遍的事实是：没有多少程序员正确地测试了他们的代码。一般来说这不是因为程序员真的认为测试很烦人或者不值得做，而是因为他们完全不知道如何去测试他们的程序，去根除错误，去保证代码的质量。其结果就是，很少有应用软件被高质量地测试过，这导致了大家普遍对软件工业评价不高。为了解决这些问题，