

深入理解计算机系统（原书第3版）



[深入理解计算机系统（原书第3版）_下载链接1](#)

著者:Randal E.Bryant

出版者:机械工业出版社

出版时间:2016-11

装帧:平装

isbn:9787111544937

和第2版相比，本版内容上*大的变化是，从以IA32和x86-64为基础转变为完全以x86-64为基础。主要更新如下：

基于x86-64，大量地重写代码，首次介绍对处理浮点数据的程序的机器级支持。

处理器体系结构修改为支持64位字和操作的设计。

引入更多的功能单元和更复杂的控制逻辑，使基于程序数据流表示的程序性能模型预测更加可靠。

扩充关于用GOT和PLT创建与位置无关代码的讨论，描述了更加强大的链接技术（比如库打桩）。

增加了对信号处理程序更细致的描述，包括异步信号安全的函数等。

采用新函数，更新了与协议无关和线程安全的网络编程。

作者介绍:

Randal E.

Bryant, 1981年于麻省理工学院获得计算机博士学位，1984年至今一直任教于卡内基-梅隆大学。现任卡内基-梅隆大学计算机科学学院院长、教授，同时还受邀任教于电子和计算机工程系。他从事本科生和研究生计算机系统方面课程的教学近40年。他和O’Hallaron教授一起在卡内基梅隆大学开设了15-213课程“计算机系统导论”，那便是本书的基础。他还是ACM院士、IEEE院士、美国国家工程院院士和美国人文与科学研究院院士。其研究成果被Intel、IBM、Fujitsu和Microsoft等主要计算机制造商使用，他还因研究获得过Semiconductor Research Corporation、ACM、IEEE颁发的多项大奖。

David R.

O’Hallaron卡内基梅隆大学电子和计算机工程系教授。在弗吉尼亚大学（University of Virginia）获得计算机科学的博士学位，2007年-2010年为Intel匹兹堡实验室主任。他教授本科生和研究生的计算机系统方面的课程已有20余年，并和Bryant教授一起开设了“计算机系统导论”课程。曾获得CMU计算机学院颁发的Herbert Simon杰出教学奖。他主要从事计算机系统领域的研究，与Quake项目成员一起获得过高性能计算领域中的*高国际奖项——Gordon Bell奖。他目前的工作重点是研究自动分级（autograding）概念，即评价其他程序质量的程序。

目录: 出版者的话

中文版序一

中文版序二

译者序

前言

关于作者

第1章 计算机系统漫游1

1.1 信息就是位+上下文1

1.2 程序被其他程序翻译成不同的格式3

1.3 了解编译系统如何工作是大有益处的4

1.4 处理器读并解释储存在内存中的指令5

1.4.1 系统的硬件组成5

1.4.2 运行hello程序7

1.5 高速缓存至关重要9

1.6 存储设备形成层次结构9

1.7 操作系统管理硬件10

1.7.1 进程11

1.7.2 线程	12
1.7.3 虚拟内存	12
1.7.4 文件	14
1.8 系统之间利用网络通信	14
1.9 重要主题	16
1.9.1 Amdahl定律	16
1.9.2 并发和并行	17
1.9.3 计算机系统中抽象的重要性	19
1.10 小结	20
参考文献说明	20
练习题答案	20
第一部分	
程序结构和执行	
第2章 信息的表示和处理	22
2.1 信息存储	24
2.1.1 十六进制表示法	25
2.1.2 字数据大小	27
2.1.3 寻址和字节顺序	29
2.1.4 表示字符串	34
2.1.5 表示代码	34
2.1.6 布尔代数简介	35
2.1.7 C语言中的位级运算	37
2.1.8 C语言中的逻辑运算	39
2.1.9 C语言中的移位运算	40
2.2 整数表示	41
2.2.1 整型数据类型	42
2.2.2 无符号数的编码	43
2.2.3 补码编码	44
2.2.4 有符号数和无符号数之间的转换	49
2.2.5 C语言中的有符号数与无符号数	52
2.2.6 扩展一个数字的位表示	54
2.2.7 截断数字	56
2.2.8 关于有符号数与无符号数的建议	58
2.3 整数运算	60
2.3.1 无符号加法	60
2.3.2 补码加法	62
2.3.3 补码的非	66
2.3.4 无符号乘法	67
2.3.5 补码乘法	67
2.3.6 乘以常数	70
2.3.7 除以2的幂	71
2.3.8 关于整数运算的最后思考	74
2.4 浮点数	75
2.4.1 二进制小数	76
2.4.2 IEEE浮点表示	78
2.4.3 数字示例	79
2.4.4 舍入	83
2.4.5 浮点运算	85
2.4.6 C语言中的浮点数	86
2.5 小结	87
参考文献说明	88
家庭作业	88
练习题答案	97
第3章 程序的机器级表示	109

- 3.1 历史观点110
- 3.2 程序编码113
 - 3.2.1 机器级代码113
 - 3.2.2 代码示例114
 - 3.2.3 关于格式的注解117
- 3.3 数据格式119
- 3.4 访问信息119
 - 3.4.1 操作数指示符121
 - 3.4.2 数据传送指令122
 - 3.4.3 数据传送示例125
 - 3.4.4 压入和弹出栈数据127
- 3.5 算术和逻辑操作128
 - 3.5.1 加载有效地址129
 - 3.5.2 一元和二元操作130
 - 3.5.3 移位操作131
 - 3.5.4 讨论131
 - 3.5.5 特殊的算术操作133
- 3.6 控制135
 - 3.6.1 条件码135
 - 3.6.2 访问条件码136
 - 3.6.3 跳转指令138
 - 3.6.4 跳转指令的编码139
 - 3.6.5 用条件控制来实现条件分支…141
 - 3.6.6 用条件传送来实现条件分支…145
 - 3.6.7 循环149
 - 3.6.8 switch语句159
- 3.7 过程164
 - 3.7.1 运行时栈164
 - 3.7.2 转移控制165
 - 3.7.3 数据传送168
 - 3.7.4 栈上的局部存储170
 - 3.7.5 寄存器中的局部存储空间172
 - 3.7.6 递归过程174
- 3.8 数组分配和访问176
 - 3.8.1 基本原则176
 - 3.8.2 指针运算177
 - 3.8.3 嵌套的数组178
 - 3.8.4 定长数组179
 - 3.8.5 变长数组181
- 3.9 异质的数据结构183
 - 3.9.1 结构183
 - 3.9.2 联合186
 - 3.9.3 数据对齐189
- 3.10 在机器级程序中将控制与数据结合起来192
 - 3.10.1 理解指针192
 - 3.10.2 应用：使用GDB调试器193
 - 3.10.3 内存越界引用和缓冲区溢出194
 - 3.10.4 对抗缓冲区溢出攻击198
 - 3.10.5 支持变长栈帧201
- 3.11 浮点代码204
 - 3.11.1 浮点传送和转换操作205
 - 3.11.2 过程中的浮点代码209
 - 3.11.3 浮点运算操作210
 - 3.11.4 定义和使用浮点常数212

- 3.11.5 在浮点代码中使用位级操作212
- 3.11.6 浮点比较操作213
- 3.11.7 对浮点代码的观察结论215
- 3.12 小结216
- 参考文献说明216
- 家庭作业216
- 练习题答案226
- 第4章 处理器体系结构243
 - 4.1 Y86-64指令集体系结构245
 - 4.1.1 程序员可见的状态245
 - 4.1.2 Y86-64指令245
 - 4.1.3 指令编码246
 - 4.1.4 Y86-64异常250
 - 4.1.5 Y86-64程序251
 - 4.1.6 一些Y86-64指令的详情255
 - 4.2 逻辑设计和硬件控制语言HCL256
 - 4.2.1 逻辑门257
 - 4.2.2 组合电路和HCL布尔表达式257
 - 4.2.3 字级的组合电路和HCL整数表达式258
 - 4.2.4 集合关系261
 - 4.2.5 存储器和时钟262
 - 4.3 Y86-64的顺序实现264
 - 4.3.1 将处理组织成阶段264
 - 4.3.2 SEQ硬件结构272
 - 4.3.3 SEQ的时序274
 - 4.3.4 SEQ阶段的实现277
 - 4.4 流水线的通用原理282
 - 4.4.1 计算流水线282
 - 4.4.2 流水线操作的详细说明284
 - 4.4.3 流水线的局限性284
 - 4.4.4 带反馈的流水线系统287
 - 4.5 Y86-64的流水线实现288
 - 4.5.1 SEQ+：重新安排计算阶段288
 - 4.5.2 插入流水线寄存器289
 - 4.5.3 对信号进行重新排列和标号292
 - 4.5.4 预测下一个PC293
 - 4.5.5 流水线冒险295
 - 4.5.6 异常处理306
 - 4.5.7 PIPE各阶段的实现308
 - 4.5.8 流水线控制逻辑314
 - 4.5.9 性能分析322
 - 4.5.10 未完成的工作323
 - 4.6 小结325
 - 参考文献说明326
 - 家庭作业327
 - 练习题答案331
- 第5章 优化程序性能341
 - 5.1 优化编译器的能力和局限性342
 - 5.2 表示程序性能345
 - 5.3 程序示例347
 - 5.4 消除循环的低效率350
 - 5.5 减少过程调用353
 - 5.6 消除不必要的内存引用354
 - 5.7 理解现代处理器357

- 5.7.1 整体操作357
- 5.7.2 功能单元的性能361
- 5.7.3 处理器操作的抽象模型362
- 5.8 循环展开366
- 5.9 提高并行性369
 - 5.9.1 多个累积变量370
 - 5.9.2 重新结合变换373
- 5.10 优化合并代码的结果小结377
- 5.11 一些限制因素378
 - 5.11.1 寄存器溢出378
 - 5.11.2 分支预测和预测错误处罚379
- 5.12 理解内存性能382
 - 5.12.1 加载的性能382
 - 5.12.2 存储的性能383
- 5.13 应用：性能提高技术387
- 5.14 确认和消除性能瓶颈388
 - 5.14.1 程序剖析388
 - 5.14.2 使用剖析程序来指导优化390
- 5.15 小结392
- 参考文献说明393
- 家庭作业393
- 练习题答案395
- 第6章 存储器层次结构399
 - 6.1 存储技术399
 - 6.1.1 随机访问存储器400
 - 6.1.2 磁盘存储406
 - 6.1.3 固态硬盘414
 - 6.1.4 存储技术趋势415
 - 6.2 局部性418
 - 6.2.1 对程序数据引用的局部性418
 - 6.2.2 取指令的局部性419
 - 6.2.3 局部性小结420
 - 6.3 存储器层次结构421
 - 6.3.1 存储器层次结构中的缓存422
 - 6.3.2 存储器层次结构概念小结424
 - 6.4 高速缓存存储器425
 - 6.4.1 通用的高速缓存存储器组织结构425
 - 6.4.2 直接映射高速缓存427
 - 6.4.3 组相联高速缓存433
 - 6.4.4 全相联高速缓存434
 - 6.4.5 有关写的问题437
 - 6.4.6 一个真实的高速缓存层次结构的解剖438
 - 6.4.7 高速缓存参数的性能影响439
 - 6.5 编写高速缓存友好的代码440
 - 6.6 综合：高速缓存对程序性能的影响444
 - 6.6.1 存储器山444
 - 6.6.2 重新排列循环以提高空间局部性447
 - 6.6.3 在程序中利用局部性450
 - 6.7 小结450
 - 参考文献说明451
 - 家庭作业451
 - 练习题答案459
- 第二部分
- 在系统上运行程序

- 第7章 链接464
 - 7.1 编译器驱动程序465
 - 7.2 静态链接466
 - 7.3 目标文件466
 - 7.4 可重定位目标文件467
 - 7.5 符号和符号表468
 - 7.6 符号解析470
 - 7.6.1 链接器如何解析多重定义的全局符号471
 - 7.6.2 与静态库链接475
 - 7.6.3 链接器如何使用静态库来解析引用477
 - 7.7 重定位478
 - 7.7.1 重定位条目479
 - 7.7.2 重定位符号引用479
 - 7.8 可执行目标文件483
 - 7.9 加载可执行目标文件484
 - 7.10 动态链接共享库485
 - 7.11 从应用程序中加载和链接共享库487
 - 7.12 位置无关代码489
 - 7.13 库打桩机制492
 - 7.13.1 编译时打桩492
 - 7.13.2 链接时打桩492
 - 7.13.3 运行时打桩494
 - 7.14 处理目标文件的工具496
 - 7.15 小结496
- 参考文献说明497
- 家庭作业497
- 练习题答案499
- 第8章 异常控制流501
 - 8.1 异常502
 - 8.1.1 异常处理503
 - 8.1.2 异常的类别504
 - 8.1.3 Linux/x86-64系统中的异常505
 - 8.2 进程508
 - 8.2.1 逻辑控制流508
 - 8.2.2 并发流509
 - 8.2.3 私有地址空间509
 - 8.2.4 用户模式和内核模式510
 - 8.2.5 上下文切换511
 - 8.3 系统调用错误处理512
 - 8.4 进程控制513
 - 8.4.1 获取进程ID513
 - 8.4.2 创建和终止进程513
 - 8.4.3 回收子进程516
 - 8.4.4 让进程休眠521
 - 8.4.5 加载并运行程序521
 - 8.4.6 利用fork和execve运行程序524
 - 8.5 信号526
 - 8.5.1 信号术语527
 - 8.5.2 发送信号528
 - 8.5.3 接收信号531
 - 8.5.4 阻塞和解除阻塞信号532
 - 8.5.5 编写信号处理程序533
 - 8.5.6 同步流以避免讨厌的并发错误540
 - 8.5.7 显式地等待信号543

- 8.6 非本地跳转546
- 8.7 操作进程的工具550
- 8.8 小结550
- 参考文献说明550
- 家庭作业550
- 练习题答案556
- 第9章 虚拟内存559
 - 9.1 物理和虚拟寻址560
 - 9.2 地址空间560
 - 9.3 虚拟内存作为缓存的工具561
 - 9.3.1 DRAM缓存的组织结构562
 - 9.3.2 页表562
 - 9.3.3 页命中563
 - 9.3.4 缺页564
 - 9.3.5 分配页面565
 - 9.3.6 又是局部性救了我们565
 - 9.4 虚拟内存作为内存管理的工具565
 - 9.5 虚拟内存作为内存保护的工具567
 - 9.6 地址翻译567
 - 9.6.1 结合高速缓存和虚拟内存570
 - 9.6.2 利用TLB加速地址翻译570
 - 9.6.3 多级页表571
 - 9.6.4 综合：端到端的地址翻译573
 - 9.7 案例研究：Intel Core i7/Linux内存系统576
 - 9.7.1 Core i7地址翻译576
 - 9.7.2 Linux虚拟内存系统580
 - 9.8 内存映射582
 - 9.8.1 再看共享对象583
 - 9.8.2 再看fork函数584
 - 9.8.3 再看execve函数584
 - 9.8.4 使用mmap函数的用户级内存映射585
 - 9.9 动态内存分配587
 - 9.9.1 malloc和free函数587
 - 9.9.2 为什么要使用动态内存分配589
 - 9.9.3 分配器的要求和目标590
 - 9.9.4 碎片591
 - 9.9.5 实现问题592
 - 9.9.6 隐式空闲链表592
 - 9.9.7 放置已分配的块593
 - 9.9.8 分割空闲块594
 - 9.9.9 获取额外的堆内存594
 - 9.9.10 合并空闲块594
 - 9.9.11 带边界标记的合并595
 - 9.9.12 综合：实现一个简单的分配器597
 - 9.9.13 显式空闲链表603
 - 9.9.14 分离的空闲链表604
 - 9.10 垃圾收集605
 - 9.10.1 垃圾收集器的基本知识606
 - 9.10.2 Mark&Sweep垃圾收集器607
 - 9.10.3 C程序的保守Mark&Sweep608
 - 9.11 C程序中常见的与内存有关的错误609
 - 9.11.1 间接引用坏指针609
 - 9.11.2 读未初始化的内存609
 - 9.11.3 允许栈缓冲区溢出610

- 9.11.4 假设指针和它们指向的对象是相同大小的610
- 9.11.5 造成错位错误611
- 9.11.6 引用指针，而不是它所指向的对象611
- 9.11.7 误解指针运算611
- 9.11.8 引用不存在的变量612
- 9.11.9 引用空闲堆块中的数据612
- 9.11.10 引起内存泄漏613
- 9.12 小结613
- 参考文献说明613
- 家庭作业614
- 练习题答案617
- 第三部分
- 程序间的交互和通信
- 第10章 系统级I/O622 10.1 Unix I/O622
- 10.2 文件623
- 10.3 打开和关闭文件624
- 10.4 读和写文件625
- 10.5 用RIO包健壮地读写626
- 10.5.1 RIO的无缓冲的输入输出函数627
- 10.5.2 RIO的带缓冲的输入函数627
- 10.6 读取文件元数据632
- 10.7 读取目录内容633
- 10.8 共享文件634
- 10.9 I/O重定向637
- 10.10 标准I/O638
- 10.11 综合：我该使用哪些I/O函数？ 638
- 10.12 小结640
- 参考文献说明640
- 家庭作业640
- 练习题答案641
- 第11章 网络编程642
- 11.1 客户端服务器编程模型642
- 11.2 网络643
- 11.3 全球IP因特网646
- 11.3.1 IP地址647
- 11.3.2 因特网域名649
- 11.3.3 因特网连接651
- 11.4 套接字接口652
- 11.4.1 套接字地址结构653
- 11.4.2 socket函数654
- 11.4.3 connect函数654
- 11.4.4 bind函数654
- 11.4.5 listen函数655
- 11.4.6 accept函数655
- 11.4.7 主机和服务的转换656
- 11.4.8 套接字接口的辅助函数660
- 11.4.9 echo客户端和服务器的示例662
- 11.5 Web服务器665
- 11.5.1 Web基础665
- 11.5.2 Web内容666
- 11.5.3 HTTP事务667
- 11.5.4 服务动态内容669
- 11.6 综合：TINY Web服务器671
- 11.7 小结678

参考文献说明678
家庭作业678
练习题答案679
第12章 并发编程681
12.1 基于进程的并发编程682
12.2 基于I/O多路复用的并发编程684
12.3 基于线程的并发编程691
12.4 多线程程序中的共享变量696
12.5 用信号量同步线程698
12.6 使用线程提高并行性710
12.7 其他并发问题716
12.8 小结722
参考文献说明723
家庭作业723
练习题答案726
附录A 错误处理729
参考文献733
• • • • • [\(收起\)](#)

[深入理解计算机系统（原书第3版）_下载链接1_](#)

标签

计算机

计算机科学

计算机系统

计算机体系结构

操作系统

CSAPP

计算机底层

编程

评论

等这本书很久啦，其实第三版的CSAPP早就翻译完了，出版社联合上交做了视频课程，内容不必多说，北大上交用了十多年的教材，想特别表扬下机工，纸张再也不是那种吹弹可破的了，还是双色版。

编程人应人手一本的五星好书，习题非常值得一做

看完会对计算机底层机制有一个overview式的理解,清楚地知道一个"hello world"程序从编写,编译,链接最后实际运行的全过程.作者在开篇提到,"如果你全心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理解,你会步上成为为数不多的大牛之路".没错,是"步上","逐渐成为"大牛.CSAPP是一本很基础的书.糅合了计算机组成原理,操作系统,体系结构,网络编程,并行程序设计原理等课程的基础知识.除第一章外,每一章甚至是一小节都可以扩展成一本书来讲授.对于刚接触编程,或者像我这种非CS专业出身的人来说,这是一本很深入和方向性的书,它会告诉你,要想成为一个优秀的程序员,你应当重点理解哪些计算机底层原理,告诉你应该在以后的自学过程中,应该重点学习哪些课程,比如体系结构等

还能说什么呢? 能把硬件、系统、软件系统地结合起来,构成一整个框架,这对于一名非科班的programmer了解计算机运作是一件多么美妙的事情!

第三版的印刷质量良心哭了。。。

挺难的。有些地方没有学习的必要，但还是硬着头皮都啃下来了。后期的温故复习的工作量怕是比读书本身的工作量还要大。

这是一本多么神奇的书呢? 无论我怎么看都像是新的

本身是习惯了Java这种高级语言，猛然从底层看，有一些不太适应。我主要是想看小数位经常在系统丢失的问题，找到了想要的答案。

就想问能不能请个懂的到来校对这书。。这次只看了点汇编，完成了bomb实验。

- 1.大二看的，这本书应该叫计算机系统导论，一点都不深入，就是入门级别的
- 2.程序员了解计算机系统的最佳书，串联了多门专业课，以组原和OS为主
- 3.配套实验很值得做

神书；然，借了一学期，看了不到两章。

如果厨二经常说什么神作的话，这本书就是当之无愧的神作

有第二版，又买了第三版。读起来如此酣畅淋漓的技术书籍真的不多。

非常好的书。以后有些经验之后再读会有更多收获吧。
20200117：打算转行的时候一位大神让我先读这本书的前6章。在入行接近2年，写了一定的C++之后，现在开始看这本书的7章之后的内容了。有了一些实际的编译性语言的经验之后看觉得太爽了。

粗略的读了一遍。很多不懂的。

看得比较粗，就是个全家桶，并没有特别深入吧，算是概论型，几乎每章都可以抽出来又是一本书

[查阅]

过瘾。第三版中文版和英文版的印刷质量很好，彩色印刷。

看了几章感兴趣的，写得确实通俗易懂，内容没得挑，不过书的纸质有点差……

这本书十分易读，脉络很清晰，同时知识点也不会缺乏深度，整个阅读过程真是太舒畅了。学生时代如果能好好花时间读一读，不知道可以少走多少弯路。
家庭作业未做，答案：<http://t.cn/R0q1UH2>。

[深入理解计算机系统（原书第3版）_下载链接1](#)

书评

CMU是全美以至全球公认的CS最猛的大学之一，没办法，作为CS的发源地，再加上三位神一样的人先后在此任教：Alan Perlis（CS它祖宗+第一届Turing奖获得者）、Allen Newell（AI缔造者+Turing奖获得者）和Herbert Simon（AI缔造者+Turing奖获得者+Nobel经济学奖获得者，当代的L...

最新课程：2014年6月30日这门课在Coursera开始第二季，地址<https://www.coursera.org/course/hwswinterface>，请大家及时关注
如果你觉得这本书过于厚重担心看不下来的话，不妨跟着coursera的Hardware/Software Interface这门课程去听一听。这本书虽然是这门课的超集，但是其中...

Chapter 1 A Tour of Computer System 一个对计算机系统总体的介绍，简单明了。应试★★★：可能在笔试中会有一些整体上的概念题。修炼★：属于计算机最基本的概念。

上个星期终于把csapp看完了。
我买的是中文版的，因为除了貌似评价不错以外，由于涉及到些自己不了解的底层东西，怕是看英文会云里雾里。现在看来，大概不能算是个优点，但是的确能够加快我的看书速度，否则一星期还真不大可能把这书搞定。
对csapp慕名已久，主要在于据说这本...

第一次在豆瓣上发东西，呵呵～

本学期选了汇编程序设计，实际上就是用这本CSAPP当教材了。作者是CMU计算机系主任，该门课的经典地位应该和MIT的SICP差不多吧（也在这学期选了～）。最初只是想混学分，结果上了之后才发觉这门课是如此强大。我们老师也将CMU原配的几个Lab作业...

这本书的中译名为“深入理解计算机系统”，有一定的问题。如果直译原书名，应该是类似于“以程序员的视角理解计算机系统”。可能在国内看来，这是讲系统的，用C和汇编语言的，因此很“深入”。事实上，这是一本入门级别的书，这本书其实并不“深入”，它谈论的内容还是相对比较...

CMU和ICS的课号为213，然后他的courseweb在这：

<http://www.cs.cmu.edu/~213/index.html>

里面有CMU往年的CS213的所有exam资料和答案：

<http://www.cs.cmu.edu/~213/exams.html> 还有另外一个资源是：

<http://www.cs.cmu.edu/~213/lectures/> 这里是FTP服务器，提供所有CS213的slide...

如果计算机科学只能带走一本书的话，我选择这本。

这是本很有趣的书，了解计算机底层技术是件极富挑战同趣味的事情，这样的书籍给人带来极大乐趣。个人认为这样的书籍，计科专业的本科学生越早接触越好，它会让你比别人更早拥有一个系统的计算机知识，更能触类旁通。如果想从事...

这本书是很好的书，我认为只要是工作中涉及编程工作的同学都要必须要熟读的书。我之前看过第二版，没看完。后来发现有第三版了，我就从网上找来一本二手的，重新从头认真读，现在读完了前5章，简单说一下。

跟之前的中文版第二版相比，这本第三版的印刷质量和纸张都有很大进步...

这是一本多么伟大的书籍！我希望我曾去过卡耐基梅隆大学并参加这门课程。这本书是卡耐基梅隆大学的教授在讲授计算机系统课程后的几年时间里写的。从程序员的观点看（作为标题来说更为恰当），这本书涵盖了广泛的主题范围，包括操作系统，编译器，计算机系统结构，集级编程，内...

直到今天,大体上看完了这本书,空过了其中四章.因为它们说的是比较以Unix为基础的技术或者是网络技术或者比较深入地讨论了某些细节.它们分别是第八章,异常控制流,第九章,测量程序执行时间,第十二章,网络编程,第十三章,并发编程.花了大约四十天,除了其中有些日子,应该是实实...

注：图片无法显示，请参考：

<http://www.cnblogs.com/remlostime/archive/2011/04/10/2011914.html>

最近在上金博的《计算机原理》。为什么说是最值得上的课，原因有二。

一者，教材是CMU的人写的久负盛名的《Computer Systems:A Programmer's Perspective》（<http://book.do...>

这本书的中文版是从原书第二版翻译的，第二版是2001年出版的。

这本书是基于IA32架构的，而目前大部分的计算机都是基于x86_64的，如果你已经身经百战见得多了可能并不在意这些区别，但是要是初学的话发现自己电脑上跑出来的效果和书上相差比较大还是会很懵逼的。这本书第三...

越来越觉得，这本书的价值远远超过我网购总价 84 元。

翻译、印刷、排版、纸张等等都非常地好，远比第一版好多了。

它使我一点点地明白了以前未知的、理解模糊的重要知识点。这学期刚好在上“计算机系统结构”这门课，用的是张晨曦老师的教材，主任一直夸这本教材是目前国内...

刚读完这本书. 感觉很像是计算机导论

将计算机与硬件相关的入门知识都笼统的介绍了一下 讲解了他们怎么用

如果有数学基础 例如数学物理生物等立刻专业的同学, 看完这本书再看看组成原理

体系结构 编译原理 还有汇编语言和操作系统原理,

计算机底层的基础知识就已经非常足够了 ...

作为一名计算机程序员，如果缺乏对计算机的层次理解，那么其基本素养是值得怀疑的，其思考基础是有欠缺的。

当我们沉浸在这样或者那样的编程教程里，沉浸在大量的语法架构之中的时候，反而常常因为这样或者那样一门复杂的技术而忘记计算机系统的构成其本身。

也是一个偶然的机会才在别人的书桌上随便翻开看看的。结果一发不可收拾，现在自己花了RMB72购入囊中，列入珍藏的书目中了。正如英文的原名所叙述的，from a programmer's perspective, 故名思义，就是从程序员的视角来看待一个计算机系统。现有的一些计算机原理书，往往过于偏...

英文名：Computer Systems: A Programmer's Perspective 作者：【美】Randal E.Bryant、David O'Hallaron 序言 第1章 计算机系统漫游
计算机系统是由硬件和系统软件组成的，它们共同协作以运行应用程序。计算机内部的信息被表示为一组组的位，它们依据不同的上下文又有...

[深入理解计算机系统（原书第3版）_下载链接1_](#)