

# 《深入理解计算机系统（原书第3版）》

## 图书基本信息

书名：《深入理解计算机系统（原书第3版）》

13位ISBN编号：9787111544935

出版时间：2016-11

作者：Randal E.Bryant,David O'Hallaron

译者：龚奕利,贺莲

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu000.com](http://www.tushu000.com)

# 《深入理解计算机系统（原书第3版）》

## 内容概要

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。

# 《深入理解计算机系统（原书第3版）》

## 作者简介

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。

## 书籍目录

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

# 《深入理解计算机系统（原书第3版）》

## 精彩短评

- 1、第三版真的加了很多新东西，非常赞！
- 2、挺不错的一本书，还是双色的，不过配套视频还没看，找时间下载来看看
- 3、相见恨晚
- 4、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
- 5、错别字好像有点多，我看了十分之一就发现了三处错字了，感觉校对的人不认真啊。
- 6、读完这本书你将成为大牛。。我读完基本写完练习，实验做了三个，做到缓冲区溢出一半，就没做了。读了一个月吧，目前还差两章，确实懂了很多以前困惑的东西。
- 7、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
- 8、我看这书时我已经修过数电和微机原理，然后觉得这书还是适合给非EE的学生拿来入门
- 9、中文版翻译的错别字较多，而且感觉完全是从第二版移植过来的，校对也不认真，比起英文版的质量差不止一点！希望后面再印刷时能够修正！
- 10、恕我直言，说这本入门的，肯定是很笼统地扫过一遍的人，看完外加做题的，总能找到很多不明之处。
- 11、确实不愧是经典，内容讲的详细，例如：虚拟存储器这一章，就详细的从虚拟存储器的实现和使用两个方面进行了讲解。

### 1、《深入理解计算机系统（原书第3版）》的笔记-第1页

我个人的经验是，有许多系统设计和概念，看似简单或不理解，可一旦自己动手做同样的试验，才明白当初设计者为什么要如此设计。计算机系统就像自然界的生态环境，对每一个部件的设计都要求它能够与系统其它部件和平相处，我们不能站在一个微观的视角去看待系统部件的设计是否最优，而应该从宏观来观察和思考确实是这样的，在我自己动手写web server、网络库之前总是觉得这些东西很高达上，不知道该如何下手，可是等到自己一行一行的把这些轮子造出来之后，回过头来想，觉得他们的原理竟是如此的简单。

# 《深入理解计算机系统（原书第3版）》

## 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:[www.tushu000.com](http://www.tushu000.com)