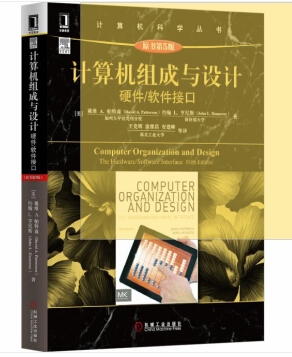
# 计算机组成与设计 硬件/软件接口

[美] [戴维A.帕特森](https://book.jd.com/writer/%E6%88%B4%E7%BB%B4A.%E5%B8%95%E7%89%B9%E6%A3%AE_1.html" \t "https://item.jd.com/_blank)，[美] [约翰 L. 亨尼斯](https://book.jd.com/writer/%E7%BA%A6%E7%BF%B0 L. %E4%BA%A8%E5%B0%BC%E6%96%AF_1.html" \t "https://item.jd.com/_blank) 著，[王党辉](https://book.jd.com/writer/%E7%8E%8B%E5%85%9A%E8%BE%89_1.html" \t "https://item.jd.com/_blank)，[康继昌](https://book.jd.com/writer/%E5%BA%B7%E7%BB%A7%E6%98%8C_1.html" \t "https://item.jd.com/_blank)，[安建峰](https://book.jd.com/writer/%E5%AE%89%E5%BB%BA%E5%B3%B0_1.html" \t "https://item.jd.com/_blank) 译

<https://item.jd.com/11729917.html>



出版者的话

本书赞誉

译者序

前言

作者简介

第1章　计算机概要与技术1

1.1　引言1

1.1.1　计算应用的分类及其特性2

1.1.2　欢迎来到后PC时代3

1.1.3　你能从本书学到什么4

1.2　计算机系统结构中的8个伟大思想6

1.2.1　面向摩尔定律的设计6

1.2.2　使用抽象简化设计6

1.2.3　加速大概率事件6

1.2.4　通过并行提高性能7

1.2.5　通过流水线提高性能7

1.2.6　通过预测提高性能7

1.2.7　存储器层次7

1.2.8　通过冗余提高可靠性7

1.3　程序概念入门7

1.4　硬件概念入门10

1.4.1　显示器11

1.4.2　触摸屏12

1.4.3　打开机箱12

1.4.4　数据安全15

1.4.5　与其他计算机通信16

1.5　处理器和存储器制造技术17

1.6　性能20

1.6.1　性能的定义20

1.6.2　性能的度量22

1.6.3　CPU性能及其因素23

1.6.4　指令的性能24

1.6.5　经典的CPU性能公式25

1.7　功耗墙27

1.8　沧海巨变：从单处理器向多处理器转变29

1.9　实例：Intel Core i7基准31

1.9.1　SPEC CPU基准测试程序31

1.9.2　SPEC功耗基准测试程序32

1.10　谬误与陷阱33

1.11　本章小结35

1.12　历史观点和拓展阅读36

1.13　练习题36

第2章　指令：计算机的语言40

2.1　引言40

2.2　计算机硬件的操作43

2.3　计算机硬件的操作数44

2.3.1　存储器操作数45

2.3.2　常数或立即数操作数47

2.4　有符号数和无符号数48

2.5　计算机中指令的表示53

2.6　逻辑操作58

2.7　决策指令60

2.7.1　循环61

2.7.2　case/switch语句63

2.8　计算机硬件对过程的支持64

2.8.1　使用更多的寄存器66

2.8.2　嵌套过程67

2.8.3　在栈中为新数据分配空间69

2.8.4　在堆中为新数据分配空间70

2.9　人机交互72

2.10　MIPS中32位立即数和寻址75

2.10.1　32位立即数75

2.10.2　分支和跳转中的寻址76

2.10.3　MIPS寻址模式总结78

2.10.4　机器语言解码79

2.11　并行与指令：同步81

2.12　翻译并执行程序83

2.12.1　编译器83

2.12.2　汇编器84

2.12.3　链接器85

2.12.4　加载器87

2.12.5　动态链接库87

2.12.6　启动一个Java程序89

2.13　以一个C排序程序作为完整的例子90

2.13.1　swap过程90

2.13.2　sort过程91

2.14　数组与指针96

2.14.1　用数组实现clear96

2.14.2　用指针实现clear97

2.14.3　比较两个版本的clear97

2.15　高级内容：编译C语言和解释Java语言98

2.16　实例：ARMv7(32位)指令集98

2.16.1　寻址模式99

2.16.2　比较和条件分支100

2.16.3　ARM的特色100

2.17　实例：x86指令集102

2.17.1　Intel x86的改进102

2.17.2　x86寄存器和数据寻址模式103

2.17.3　x86整数操作105

2.17.4　x86指令编码107

2.17.5　x86总结108

2.18　实例：ARMv8（64位）指令集108

2.19　谬误与陷阱109

2.20　本章小结110

2.21　历史观点和拓展阅读111

2.22　练习题112

第3章　计算机的算术运算117

3.1　引言117

3.2　加法和减法117

3.3　乘法121

3.3.1　顺序的乘法算法和硬件121

3.3.2　有符号乘法124

3.3.3　更快速的乘法124

3.3.4　MIPS中的乘法124

3.3.5　小结125

3.4　除法125

3.4.1　除法算法及其硬件结构125

3.4.2　有符号除法128

3.4.3　更快速的除法128

3.4.4　MIPS中的除法129

3.4.5　小结129

3.5　浮点运算130

3.5.1　浮点表示131

3.5.2　浮点加法135

3.5.3　浮点乘法138

3.5.4　MIPS中的浮点指令139

3.5.5　算术精确性145

3.5.6　小结146

3.6　并行性和计算机算术：子字并行148

3.7　实例：x86中流处理SIMD扩展和高级向量扩展149

3.8　加速：子字并行和矩阵乘法150

3.9　谬误与陷阱153

3.10　本章小结155

3.11　历史观点和拓展阅读158

3.12　练习题159

第4章　处理器162

4.1　引言162

4.2　逻辑设计的一般方法165

4.3　建立数据通路167

4.4　一个简单的实现机制173

4.4.1　ALU控制173

4.4.2　主控制单元的设计175

4.4.3　为什么不使用单周期实现方式181

4.5　流水线概述182

4.5.1　面向流水线的指令集设计186

4.5.2　流水线冒险186

4.5.3　对流水线概述的小结191

4.6　流水线数据通路及其控制192

4.6.1　图形化表示的流水线200

4.6.2　流水线控制203

4.7　数据冒险：旁路与阻塞206

4.8　控制冒险214

4.8.1　假定分支不发生215

4.8.2　缩短分支的延迟215

4.8.3　动态分支预测216

4.8.4　流水线小结220

4.9　异常221

4.9.1　MIPS体系结构中的异常处理221

4.9.2　在流水线实现中的异常222

4.10　指令级并行226

4.10.1　推测的概念227

4.10.2　静态多发射处理器227

4.10.3　动态多发射处理器231

4.10.4　能耗效率与高级流水线233

4.11　实例：ARM Cortex-A8和Intel Core i7流水线234

4.11.1　ARM Cortex-A8235

4.11.2　Intel Core i7 920236

4.11.3　Intel Core i7 920的性能238

4.12　运行更快：指令级并行和矩阵乘法240

4.13　高级主题：通过硬件设计语言描述和建模流水线来介绍数字设计以及更多流水线示例242

4.14　谬误与陷阱242

4.15　本章小结243

4.16　历史观点和拓展阅读243

4.17　练习题243

第5章　大容量和高速度：开发存储器层次结构252

5.1　引言252

5.2　存储器技术255

5.2.1　SRAM技术256

5.2.2　DRAM技术256

5.2.3　闪存258

5.2.4　磁盘存储器258

5.3　cache的基本原理259

5.3.1　cache访问261

5.3.2　cache缺失处理265

5.3.3　写操作处理266

5.3.4　一个cache的例子:内置FastMATH处理器267

5.3.5　小结269

5.4　cache性能的评估和改进270

5.4.1　通过更灵活地放置块来减少cache缺失272

5.4.2　在cache中查找一个块275

5.4.3　替换块的选择276

5.4.4　使用多级cache结构减少缺失代价277

5.4.5　通过分块进行软件优化280

5.4.6　小结283

5.5　可信存储器层次283

5.5.1　失效的定义283

5.5.2　纠正一位错、检测两位错的汉明编码（SEC/DED）284

5.6　虚拟机287

5.6.1　虚拟机监视器的必备条件289

5.6.2　指令集系统结构（缺乏）对虚拟机的支持289

5.6.3　保护和指令集系统结构289

5.7　虚拟存储器290

5.7.1　页的存放和查找293

5.7.2　缺页故障294

5.7.3　关于写297

5.7.4　加快地址转换：TLB297

5.7.5　集成虚拟存储器、TLB和cache 300

5.7.6　虚拟存储器中的保护302

5.7.7　处理TLB缺失和缺页303

5.7.8　小结307

5.8　存储器层次结构的一般框架309

5.8.1　问题1：一个块可以被放在何处309

5.8.2　问题2：如何找到一个块310

5.8.3　问题3：当cache缺失时替换哪一块311

5.8.4　问题4：写操作如何处理311

5.8.5　3C：一种理解存储器层次结构行为的直观模型312

5.9　使用有限状态机来控制简单的cache314

5.9.1　一个简单的cache314

5.9.2　有限状态机315

5.9.3　一个简单的cache控制器的有限状态机316

5.10　并行与存储器层次结构：cache一致性317

5.10.1　实现一致性的基本方案318

5.10.2　监听协议319

5.11　并行与存储器层次结构：冗余廉价磁盘阵列320

5.12　高级内容：实现cache控制器320

5.13　实例：ARM Cortex-A8和Intel Core i7的存储器层次结构320

5.14　运行更快:cache分块和矩阵乘法324

5.15　谬误和陷阱326

5.16　本章小结329

5.17　历史观点和拓展阅读329

5.18　练习题329

第6章　从客户端到云的并行处理器340

6.1　引言340

6.2　创建并行处理程序的难点342

6.3　SISD、MIMD、SIMD、SPMD和向量机345

6.3.1　在x86中的SIMD：多媒体扩展346

6.3.2　向量机346

6.3.3　向量与标量的对比347

6.3.4　向量与多媒体扩展的对比348

6.4　硬件多线程350

6.5　多核和其他共享内存多处理器352

6.6　图形处理单元简介355

6.6.1　NVIDIA GPU体系结构简介356

6.6.2　NVIDIA GPU存储结构357

6.6.3　GPU展望358

6.7　集群、仓储级计算机和其他消息传递多处理器360

6.8　多处理器网络拓扑简介363

6.9　与外界通信：集群网络366

6.10　多处理器测试集程序和性能模型366

6.10.1　性能模型368

6.10.2　Roofline模型369

6.10.3　两代Opteron的比较370

6.11　实例：评测Intel Core i7 960和NVIDIA Tesla GPU的Roofline模型373

6.12　运行更快：多处理器和矩阵乘法376

6.13　谬误与陷阱378

6.14　本章小结379

6.15　历史观点和拓展阅读381

6.16　练习题382

附录A　汇编器、链接器和SPIM仿真器389

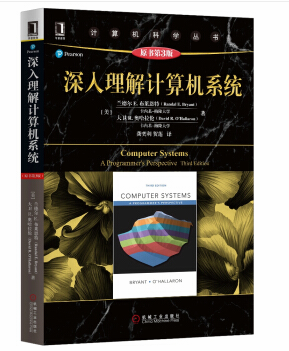
附录B　逻辑设计基础437

索引494

# [深入理解计算机系统](https://item.jd.com/12006637.html)

[美] [兰德尔 E.布莱恩特](https://book.jd.com/writer/%E5%85%B0%E5%BE%B7%E5%B0%94 E.%E5%B8%83%E8%8E%B1%E6%81%A9%E7%89%B9_1.html" \t "https://item.jd.com/_blank)（[Randal E.·Bryant](https://book.jd.com/writer/Randal E.%C2%B7Bryant_1.html" \t "https://item.jd.com/_blank)） 著，[龚奕利](https://book.jd.com/writer/%E9%BE%9A%E5%A5%95%E5%88%A9_1.html" \t "https://item.jd.com/_blank)，[贺莲](https://book.jd.com/writer/%E8%B4%BA%E8%8E%B2_1.html" \t "https://item.jd.com/_blank) 译

<https://item.jd.com/12006637.html>



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

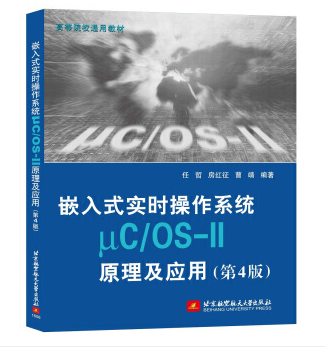
家庭作业723

练习题答案726

附录A　错误处理729

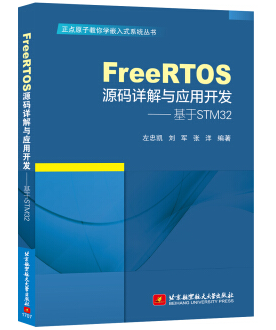
参考文献733

# 嵌入式实时操作系统μC/OS-II原理及应用(第4版 任哲,房红

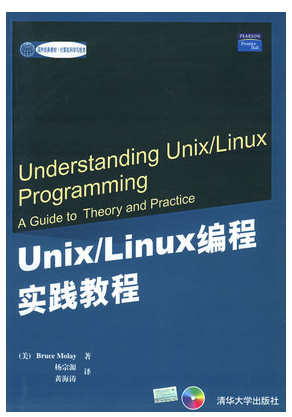


# FreeRTOS源码详解与应用开发—基于STM32

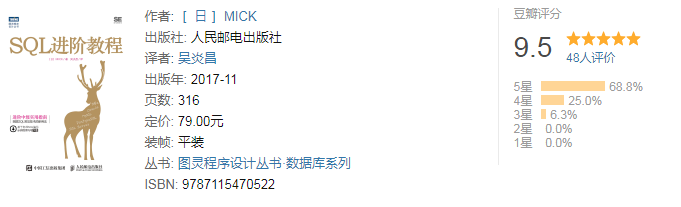
<https://item.jd.com/12121539.html>



# UNIX/Linux编程实践教程

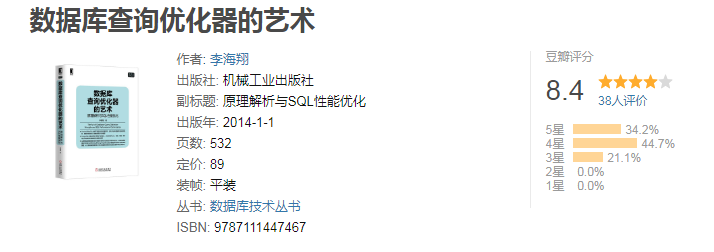


# **SQL进阶教程**



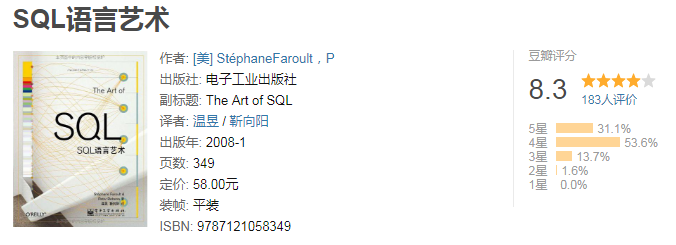
第1章 神奇的 SQL  
1-1　CASE表达式　　2  
▲ 在SQL里表达条件分支　　2  
练习题　　19  
1-2　自连接的用法　　21  
▲ 面向集合语言SQL　　21  
练习题　　35  
1-3　三值逻辑和NULL　　38  
▲ SQL的温柔陷阱　　38  
1-4　HAVING子句的力量　　55  
▲ 出彩的配角　　55  
练习题　　70  
1-5　外连接的用法　　72  
▲ SQL的弱点及其趋势和对策　　72  
练习题　　92  
1-6　用关联子查询比较行与行　　94  
▲ 用SQL进行行与行之间的比较　　94  
练习题　　110  
1-7　用SQL进行集合运算　　112  
▲ SQL和集合论　　112  
练习题　　128  
1-8　EXISTS谓词的用法　　　　130  
▲ SQL中的谓词逻辑　　130  
练习题　　146  
1-9　用SQL处理数列　　　　149  
▲ 灵活使用谓词逻辑　　149  
练习题　　165  
1-10　HAVING子句又回来了　　167  
▲ 再也不要叫它配角了！　　167  
练习题　　183  
1-11　让SQL飞起来　　　　186  
▲ 简单的性能优化　　186  
1-12　关系数据库的世界　　　　216  
▲ 确立SQL的编程风格　　201  
第2章 SQL 编程方法  
2-1　关系数据库的历史　　　　216  
▲ 1969年——一切从这里开始　　216  
2-2　为什么叫“关系”模型　　222  
▲ 为什么不叫“表”模型　　222  
2-3　开始于关系，结束于关系　　229  
▲ 关于封闭世界的幸福　　229  
2-4　地址这一巨大的怪物　　233  
▲ 为什么关系数据库里没有指针　　233  
2-5　GROUP BY和PARTITION BY　　238  
▲ 物以“类”聚　　238  
1-8　EXISTS谓词的用法  
1-9　用SQL处理数列  
1-10　HAVING子句又回来了  
1-11　让SQL飞起来  
1-12　关系数据库的世界  
2-1　关系数据库的历史  
2-2　为什么叫“关系”模型  
2-3　开始于关系，结束于关系　　229  
2-4　地址这一巨大的怪物  
2-5　GROUP BY和PARTITION BY　　238  
2-6　从面向过程思维向声明式思维、面向集合思维转变的7个关键点　　243  
▲ 画圆　　243  
2-7　SQL和递归集合　　　　250  
▲ SQL和集合论之间　　250  
2-8　人类的逻辑学　　　　256  
▲ 浅谈逻辑学的历史　　256  
2-9　消灭NULL委员会　　　　260  
▲ 全世界的数据库工程师团结起来！　　260  
2-10　SQL中的层级　　　　264  
▲ 严格的等级社会　　266  
第3章 附录  
3-1　习题解答　　　　274  
3-2　参考文献　　　　298  
后　记　　　　302

# **数据库查询优化器的艺术**



推荐序一  
推荐序二  
前言  
第一篇 查询优化技术  
第1章 数据管理系统的查询优化  
1.1 数据库调优  
1.2 查询优化技术  
1.2.1 查询重用  
1.2.2 查询重写规则  
1.2.3 查询算法优化  
1.2.4 并行查询优化  
1.2.5 分布式查询优化  
1.2.6 其他优化  
1.3 本章小结  
第2章 逻辑查询优化  
2.1 查询优化技术的理论基础  
2.1.1 关系代数  
2.1.2 关系代数等价变换规则对优化的指导意义  
2.2 查询重写规则  
2.2.1 子查询的优化  
2.2.2 视图重写  
2.2.3 等价谓词重写  
2.2.4 条件化简  
2.2.5 外连接消除  
2.2.6 嵌套连接消除  
2.2.7 连接消除  
2.2.8 语义优化  
2.2.9 针对非SPJ的优化  
2.3 启发式规则在逻辑优化阶段的应用  
2.4 本章小结  
第3章 物理查询优化  
3.1 查询代价估算  
3.1.1 代价模型  
3.1.2 选择率计算的常用方法  
3.2 单表扫描算法  
3.2.1 常用的单表扫描算法  
3.2.2 单表扫描代价计算  
3.3 索引  
3.3.1 如何利用索引  
3.3.2 索引列的位置对使用索引的影响  
3.3.3 联合索引对索引使用的影响  
3.3.4 多个索引对索引使用的影响  
3.4 两表连接算法  
3.4.1 基本的两表连接算法  
3.4.2 进一步认识两表连接算法  
3.4.3 连接操作代价计算  
3.5 多表连接算法  
3.5.1 多表连接顺序  
3.5.2 常用的多表连接算法  
3.5.3 多表连接算法的比较  
3.6 本章小结  
第4章 查询优化器与其他模块的关系  
4.1 查询优化器整体介绍  
4.2 查询优化器与其他模块的关系  
4.3 本章小结  
第二篇 PostgreSQL查询优化器原理解析  
第5章 PostgreSQL查询优化器概述  
5.1 PostgreSQL查询执行过程  
5.2 PostgreSQL查询优化器的架构和设计思想  
5.2.1 PostgreSQL查询优化器架构  
5.2.2 PostgreSQL查询优化器的层次  
5.2.3 PostgreSQL查询优化器设计思想  
5.3 主要概念  
5.4 代码层次结构  
5.5 本章小结  
第6章 PostgreSQL查询优化器相关数据结构  
6.1 主要数据结构  
6.1.1 基本数据结构  
6.1.2 查询树  
6.1.3 各种对象的结构  
6.1.4 连接操作相关的结构  
6.1.5 查询执行计划相关的结构  
6.2 各个结构之间的关系  
6.3 各个阶段间和主要结构体间的关系  
6.4 本章小结  
第7章 PostgreSQL查询优化器实现原理解析  
7.1 查询优化整体流程  
7.2 查询优化器实现原理解析  
7.2.1 planner--主入口函数  
7.2.2 standard\_planner--标准的查询优化器函数  
7.2.3 subquery\_planner--生成（子）查询执行计划函数  
7.2.4 grouping\_planner--生成查询执行计划并对非SPJ优化  
7.2.5 build\_minmax\_path--聚集函数MIN/MAX的优化函数  
7.2.6 query\_planner--生成最优的查询路径函数  
7.2.7 make\_one\_rel--构造多表连接路径并选出最优路径函数  
7.2.8 make\_rel\_from\_joinlist--生成多表连接路径函数  
7.2.9 optimize\_minmax\_aggregates--聚集操作MIN/MAX优化函数  
7.2.10 create\_plan--创建查询执行计划函数  
7.2.11 非SPJ处理--grouping\_planner的各个子模块  
7.2.12 其他重要的函数与操作  
7.3 代价估算实现原理解析  
7.3.1 查询代价估算  
7.3.2 单表扫描方式的代价估算  
7.3.3 两表连接的代价估算  
7.3.4 其他代价估算函数  
7.3.5 选择率的计算  
7.4 从目录结构和文件功能角度看查询优化器  
7.4.1 查询优化子模块与主要文件的关系  
7.4.2 查询优化器代码结构  
7.5 本章小结  
第8章 从功能的角度看PostgreSQL查询优化  
8.1 优化器之逻辑查询优化  
8.1.1 视图重写  
8.1.2 子查询优化  
8.1.3 等价谓词重写  
8.1.4 条件化简  
8.1.5 外连接消除  
8.1.6 嵌套连接消除  
8.1.7 连接的消除  
8.1.8 语义优化  
8.1.9 选择操作下推  
8.1.10 非SPJ优化  
8.2 优化器之物理查询优化  
8.2.1 PostgreSQL的物理优化主要完成的工作  
8.2.2 启发式规则在物理查询优化阶段的使用  
8.2.3 两表连接  
8.2.4 代价估算  
8.2.5 PostgreSQL的索引与查询优化  
8.3 其他  
8.3.1 grouping\_planner函数主干再分析  
8.3.2 用户指定的连接语义与PostgreSQL实现两表连接的函数及算法的关系  
8.3.3 集合操作优化  
8.4 本章小结  
第9章 PostgreSQL查询优化的关键算法  
9.1 动态规划算法  
9.1.1 动态规划算法的处理流程  
9.1.2 紧密树处理流程  
9.2 遗传算法  
9.2.1 PostgreSQL遗传算法的处理流程  
9.2.2 主要的数据结构  
9.2.3 主要的函数和变量  
9.2.4 应用遗传算法实现表连接的语义  
9.2.5 应用遗传算法计算适应度  
9.2.6 进一步理解PostgreSQL的遗传算法  
9.3 动态规划算法与遗传算法对比  
9.4 本章小结  
第10章 PostgreSQL查询优化器与其他部分的关系  
10.1 查询优化器与语法分析器  
10.2 查询优化器与执行器  
10.3 查询优化器与缓冲区管理模块  
10.4 查询优化器与对象访问模块  
10.5 查询优化器与统计模块  
10.6 查询优化器与索引模块  
10.7 本章小结  
第三篇 MySQL查询优化器原理解析  
第11章 MySQL查询优化器概述  
11.1 MySQL查询执行过程  
11.2 MySQL查询优化器的架构和设计思想  
11.2.1 MySQL查询优化器架构  
11.2.2 MySQL查询优化器的层次  
11.2.3 MySQL查询优化器设计思想  
11.3 主要概念  
11.3.1 常量表  
11.3.2 表数据的访问方式  
11.4 代码层次结构  
11.5 本章小结  
第12章 MySQL查询优化器相关数据结构  
12.1 主要的类和数据结构  
12.1.1 查询树  
12.1.2 基本对象  
12.1.3 连接对象与执行计划  
12.1.4 代价估算类  
12.2 各个阶段主要结构体间的关系  
12.3 本章小结  
第13章 MySQL查询优化器的原理解析  
13.1 查询优化器整体流程  
13.2 优化器的代码详解  
13.2.1 JOIN.prepare--优化前的准备工作  
13.2.2 JOIN.optimize--优化器主入口方法  
13.2.3 make\_join\_statistics--计算最优的查询优化执行计划  
13.2.4 choose\_table\_order--求解多表连接最优连接路径  
13.2.5 make\_join\_statistics函数的其他子函数  
13.2.6 make\_join\_select--对条件求值、下推连接条件到表中  
13.2.7 test\_if\_skip\_sort\_order--排序操作的优化  
13.2.8 make\_join\_readinfo--为连接的每个表构造信息  
13.2.9 JOIN.exec--执行查询执行计划的函数  
13.3 代价估算  
13.3.1 查询代价估算模型  
13.3.2 查询代价估算过程  
13.3.3 其他的代价估算  
13.3.4 对存储引擎的调用接口  
13.3.5 统计信息  
13.4 本章小结  
第14章 从功能的角度看MySQL查询优化  
14.1 优化器之逻辑查询优化  
14.1.1 视图重写  
14.1.2 子查询优化  
14.1.3 等价谓词重写  
14.1.4 条件化简  
14.1.5 外连接消除  
14.1.6 嵌套连接消除  
14.1.7 连接的消除  
14.1.8 语义优化  
14.1.9 非SPJ优化  
14.2 优化器之物理查询优化  
14.2.1 MySQL的物理优化主要完成的工作  
14.2.2 启发式规则在物理查询优化阶段的使用  
14.2.3 MySQL的索引与查询优化  
14.2.4 用户指定的连接语义与MySQL实现两表连接的算法  
14.3 本章小结  
第15章 MySQL查询优化的关键算法  
15.1 深入理解MySQL的多表连接算法  
15.2 本章小结  
第16章 MySQL查询优化器与其他部分的关系  
16.1 查询优化器与语法分析器  
16.2 查询优化器与执行器  
16.3 查询优化器与缓冲区管理模块  
16.4 查询优化器与索引模块  
16.5 本章小结  
第四篇 PostgreSQL查询优化器VSMySQL查询优化器  
第17章 PostgreSQL和MySQL的逻辑查询优化技术  
17.1 查询重写  
17.1.1 子查询优化  
17.1.2 视图重写  
17.1.3 等价谓词重写  
17.1.4 条件化简  
17.1.5 外连接消除  
17.1.6 嵌套连接消除  
17.1.7 连接消除  
17.1.8 语义优化  
17.2 非SPJ的优化  
17.3 本章小结  
第18章 PostgreSQL和MySQL的物理查询优化技术  
18.1 查询代价估算模型比较  
18.2 单表扫描算法  
18.3 索引  
18.4 两表连接算法  
18.5 多表连接算法  
18.6 本章小结  
第19章 PostgreSQL和MySQL的其他异同  
19.1 启发式规则的使用比较  
19.2 综合比较  
19.2.1 基本概念的比较  
19.2.2 数据结构的比较  
19.2.3 设计思想的比较  
19.2.4 编码规范的比较  
19.3 本章小结  
附录A 如何掌握数据库内核  
附录B 如何阅读本书  
附录C 如何阅读查询执行计划  
附录D 如何跟踪查询执行计划

# **SQL语言艺术**



前言  
1 制定计划：为性能而设计  
数据的关系视图  
规范化的重要性  
有值、无值、空值  
限用Boolean型字段  
理解子类型（Subtype）  
约束应明确声明  
过于灵活的危险性  
历史数据的难题  
设计与性能  
处理流程  
数据集中化（Centralizing）  
系统复杂性  
小结  
2 发动战争：高效访问数据库  
查询的识别  
保持数据库连接稳定  
战略优先于战术  
先定义问题，再解决问题  
保持数据库Schema稳定  
直接操作实际数据  
用SQL处理集合  
动作丰富的SQL语句  
充分利用每次数据库访问  
接近DBMS核心  
只做必须做的  
SQL语句反映业务逻辑  
把逻辑放到查询中  
一次完成多个更新  
慎用自定义函数  
简洁的SQL  
SQL的进攻式编程  
精明地使用异常（Exceptions）  
3 战术部署：建立索引  
找到“切入点”  
索引与目录  
让索引发挥作用  
函数和类型转换对索引的影响  
索引与外键  
同一字段，多个索引  
系统生成键  
索引访问的不同特点  
4 机动灵活：思考SQL语句  
SQL的本质  
掌握SQL艺术的五大要素  
过滤  
5 了如指掌：理解物理实现  
物理结构的类型  
冲突的目标  
把索引当成数据仓库  
记录强制排序  
数据自动分组（Grouping）  
分区是双刃剑  
分区与数据分布  
数据分区的最佳方法  
预连接表  
神圣的简单性  
6 锦囊妙计：认识经典SQL模式  
小结果集，直接条件  
小结果集，间接条件  
多个宽泛条件的交集  
多个间接宽泛条件的交集  
大结果集  
基于一个表的自连接  
通过聚合获得结果集  
基于日期的简单搜索或范围搜索  
结果集和别的数据存在与否有关  
7 变换战术：处理层次结构  
小结果集，直接条件  
小结果集，间接条件  
多个宽泛条件的交集  
多个间接宽泛条件的交集  
大结果集  
基于一个表的自连接  
通过聚合获得结果集  
基于日期的简单搜索或范围搜索  
结果集和别的数据存在与否有关  
8 孰优孰劣：认识困难，处理困难  
看似高效的查询条件  
抽象层  
分布式系统  
动态定义的搜索条件  
9 多条战线：处理并发  
数据库引擎作为服务提供者  
并发修改数据  
10 集中兵力：应付大数据量  
增长的数据量  
数据仓库  
11 精于计谋：挽救响应时间  
数据的行列转换  
基于变量列表的查询  
基于范围的聚合  
一般规则，最后使用  
查询与列表中多个项目相符的记录  
最佳匹配查询  
优化器指令  
12 明察秋毫：监控性能  
数据库速度缓慢  
服务器负载因素  
何谓“性能优良”  
从业务任务角度思考  
执行计划  
合理运用执行计划  
总结：影响性能的重要因素  
Photo Credits  
索引