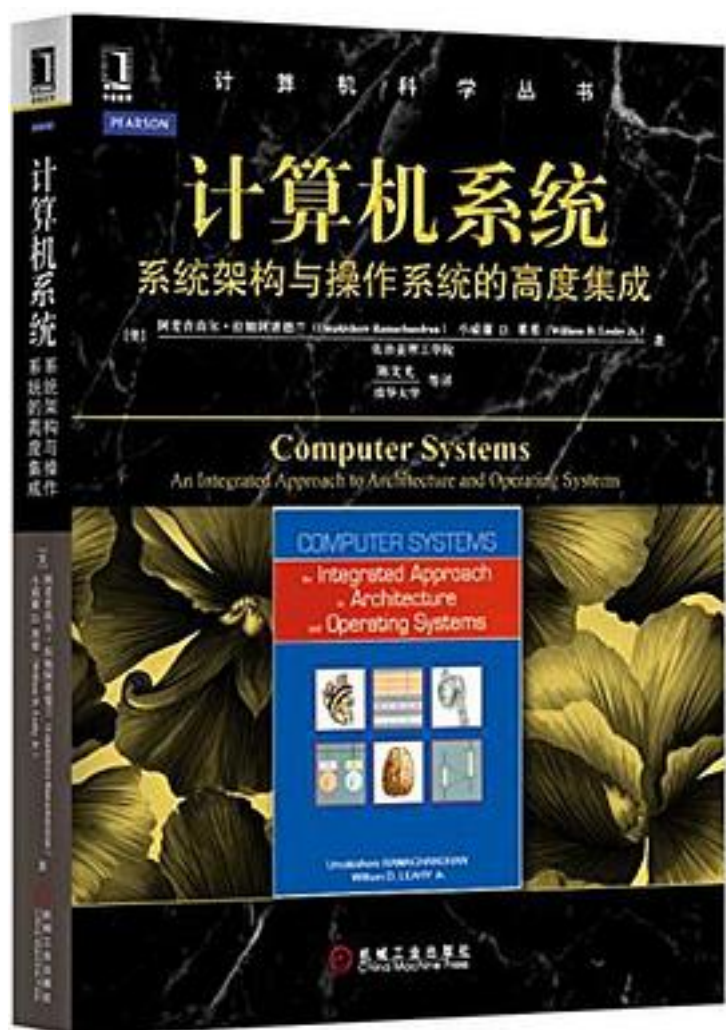


计算机系统：系统架构与操作系统的高度集成



计算机系统：系统架构与操作系统的高度集成_下载链接1

著者:

出版者:

出版时间:2015-7

装帧:

isbn:9787111506362

作者介绍:

Umakishore Ramachandran,

拥有威斯康星大学麦迪逊分校计算机专业博士学位。现在是佐治亚理工学院计算机系教授，STAR Center & Korean Programs中心主任。主要研究兴趣是体系结构设计、程序设计和并行分布式系统分析。曾获得NSF授予的美国总统青年研究人员奖、佐治亚理工学院优秀博士论文指导奖、杰出教师奖等。2014年，他被选为IEEE会士。

William D. Leahy Jr.

现为佐治亚理工学院计算机系讲师，讲授计算机科学的入门课程。他拥有佐治亚理工学院计算机科学硕士学位，在加入学术界之前，他有24年工业界的从业经验。

陈文光，清华大学计算机系教授。现为ACM中国理事会副主席；中国计算机学会杰出会员和杰出讲者，副秘书长。主要研究领域为操作系统、编译器与并行计算。讲授《计算机原理与系统结构》《计算机系统性能分析》等本科生、研究生课程。获国家科技进步二等奖一次，部级科技一等奖两次，部级科技进步二等奖两次。在计算机系统结构、程序设计语言与编译器、并行计算和软件的国际一流会议（如PLDI、PPOPP、PACT、ICSE、OOPSLA、Supercomputing CGO等）上连续发表多篇论文。PLDI 2012、PPOPP 2013、2014和CGO 2014程序委员会委员。任《Communication of ACM China Edition》主编，《Journal of Computer Science and Technology》《软件学报》等学术期刊编委。

目录: 出版者的话

译者序

前言

第1章 概述 1

1.1 盒子里有什么 1

1.2 计算机系统抽象层次 1

1.3 操作系统的作用 3

1.4 盒子里正在发生什么事 5

1.4.1 在计算机上启动应用程序 7

1.5 计算机硬件的演化 7

1.6 操作系统的演化 9

1.7 本书导读 9

练习题 10

参考文献注释和扩展阅读 10

第2章 处理器体系结构 12

2.1 处理器设计涉及什么 12

2.2 如何设计指令集 13

2.3 常见的高级语言功能集 13

2.4 表达式和赋值语句 14

2.4.1 操作数放在哪里 14

2.4.2 在指令中如何指定内存地址 17

2.4.3 每个操作数应该有多宽 18

2.4.4 字节序 9

2.4.5 操作数打包以及字操作数的对齐 21

2.5 高级数据抽象 22

2.5.1 结构 23

2.5.2 数组 23

2.6 条件语句和循环 24

2.6.1 if-then-else语句	25
2.6.2 switch语句	26
2.6.3 循环语句	27
2.7 检查点	27
2.8 编译函数调用	27
2.8.1 调用者的状态	28
2.8.2 过程调用剩余的工作	30
2.8.3 软件惯例	31
2.8.4 活动记录	35
2.8.5 递归	36
2.8.6 帧指针	36
2.9 指令集体系结构选择	38
2.9.1 额外的指令	38
2.9.2 额外的寻址模式	39
2.9.3 体系结构类型	39
2.9.4 指令格式	39
2.10 LC-2200指令集	42
2.10.1 指令格式	42
2.10.2 LC-2200寄存器组	43
2.11 影响处理器设计的问题	44
2.11.1 指令集	44
2.11.2 应用程序对指令集设计的影响	45
2.11.3 其他驱动处理器设计的问题	46
小结	47
练习题	47
参考文献注释和扩展阅读	49
第3章 处理器实现	51
3.1 体系结构与实现	51
3.2 处理器实现涉及什么	51
3.3 重要的硬件概念	52
3.3.1 电路	52
3.3.2 数据通路的硬件资源	52
3.3.3 边沿触发逻辑	53
3.3.4 连接数据通路元件	54
3.3.5 基于总线的设计	57
3.3.6 有限状态机	59
3.4 数据通路设计	60
3.4.1 ISA与数据通路宽度	61
3.4.2 时钟脉冲宽度	62
3.4.3 检查点	62
3.5 控制单元设计	62
3.5.1 ROM加状态寄存器	63
3.5.2 FETCH宏状态	65
3.5.3 DECODE宏状态	68
3.5.4 EXECUTE宏状态: ADD指令 (R型指令部分)	68
3.5.5 EXECUTE宏状态: NAND指令 (R型指令部分)	71
3.5.6 EXECUTE宏状态: JALR指令 (J型指令部分)	71
3.5.7 EXECUTE宏状态: LW指令 (I型指令部分)	72
3.5.8 EXECUTE宏状态: SW和ADDI指令 (I型指令部分)	75
3.5.9 EXECUTE宏状态: BEQ指令 (I型指令部分)	75
3.5.10 设计微程序中的条件分支	78
3.5.11 再谈DECODE宏状态	79
3.6 控制单元设计的另一种选择	80
3.6.1 微程序控制	80

- 3.6.2 硬连线控制 81
- 3.6.3 在两种控制设计风格中选择 82
- 小结 82
- 历史回顾 83
- 练习题 84
- 参考文献注释和扩展阅读 86
- 第4章 中断、陷入及异常 87
- 4.1 程序执行中的不连续性 88
- 4.2 处理程序不连续性 89
- 4.3 处理程序不连续性的体系结构改进 91
- 4.3.1 修改FSM 91
- 4.3.2 一个简单的中断处理过程 92
- 4.3.3 处理级联中断 92
- 4.3.4 从处理过程中返回 95
- 4.3.5 检查点 95
- 4.4 处理程序不连续性的硬件细节 96
- 4.4.1 中断的数据通路细节 96
- 4.4.2 获得处理过程地址的细节 97
- 4.4.3 保存/恢复栈 99
- 4.5 信息汇总 100
- 4.5.1 体系结构和硬件改进总结 100
- 4.5.2 工作中的中断机制 100
- 小结 102
- 练习题 103
- 参考文献注释和扩展阅读 104
- 第5章 处理器性能与流水线处理器的设计 105
- 5.1 时间和空间性能指标 105
- 5.2 指令频率 107
- 5.3 基准测试程序 108
- 5.4 提升处理器的性能 111
- 5.5 加速比 112
- 5.6 提升处理器的吞吐量 114
- 5.7 流水线简介 115
- 5.8 指令处理流水线 115
- 5.9 简单指令流水线的问题 117
- 5.10 修正指令流水线里的问题 118
- 5.11 指令流水线的通路元件 120
- 5.12 针对流水线的体系结构与实现 121
- 5.12.1 指令穿过流水线的过程详解 122
- 5.12.2 流水线寄存器的设计 124
- 5.12.3 各个阶段的实现 125
- 5.13 冒险 125
- 5.13.1 结构性冒险 126
- 5.13.2 数据冒险 126
- 5.13.3 控制冒险 135
- 5.13.4 冒险总结 141
- 5.14 在流水线处理器里处理程序不连续性 142
- 5.15 处理器设计的高级话题 144
- 5.15.1 指令级并行 144
- 5.15.2 更深的流水线 145
- 5.15.3 在乱序执行下再次讨论程序不连续性 147
- 5.15.4 管理共享资源 148
- 5.15.5 功耗 149
- 5.15.6 多核处理器设计 149

5.15.7 Intel Core微架构：一个流水线 150

小结 151

历史回顾 152

练习题 152

参考文献注释和扩展阅读 156

第6章 处理器调度 157

6.1 引言 157

6.2 程序和进程 158

6.3 调度环境 161

6.4 调度基础 162

6.5 性能指标 165

6.6 非抢占式调度算法 167

6.6.1 先到先服务 167

6.6.2 最短作业优先 170

6.6.3 优先级 171

6.7 抢占式调度算法 172

6.7.1 轮转调度器 175

6.8 结合优先级和抢占 178

6.9 元调度器 178

6.10 评价 179

6.11 调度对处理器体系结构的影响 180

小结和展望 181

Linux调度器——一个案例研究 181

历史回顾 183

练习题 185

参考文献注释和扩展阅读 186

第7章 内存管理技术 187

7.1 内存管理器提供的功能 187

7.2 内存管理的简单方案 189

7.3 内存分配方案 192

7.3.1 固定尺寸分区 192

7.3.2 变长分区 193

7.3.3 缩并 195

7.4 分页虚拟内存 196

7.4.1 页表 197

7.4.2 支持分页的硬件 199

7.4.3 页表的建立 199

7.4.4 虚拟和物理内存的相对大小 200

7.5 分段虚拟内存 200

7.5.1 支持分段的硬件 204

7.6 分页和分段的比较 204

7.6.1 解读CPU生成的地址 206

小结 207

历史回顾 208

MULTICS 209

Intel的内存体系结构 210

练习题 211

参考文献注释和扩展阅读 212

第8章 页式内存管理 213

8.1 按需分页 213

8.1.1 按需分页的硬件 213

8.1.2 页错误处理程序 214

8.1.3 按需分页内存管理的数据结构 214

8.1.4 页错误解析 215

- 8.2 进程调度器和内存管理器间交互 217
- 8.3 页替换策略 218
 - 8.3.1 Belady的Min算法 219
 - 8.3.2 随机替换 219
 - 8.3.3 先进先出策略 219
 - 8.3.4 最近最少使用策略 221
 - 8.3.5 第二次机会页替换算法 223
 - 8.3.6 页替换算法回顾 225
- 8.4 优化内存管理 225
 - 8.4.1 空闲页帧池 225
 - 8.4.2 颠簸 226
 - 8.4.3 工作集 228
 - 8.4.4 颠簸控制 229
- 8.5 其他考虑 229
- 8.6 旁路转换缓存 230
 - 8.6.1 TLB的地址转换 231
- 8.7 内存管理的高级话题 232
 - 8.7.1 多级页表 232
 - 8.7.2 局部页表项的访问权限 234
 - 8.7.3 反向页表 234
- 小结 234
- 练习题 234
- 参考文献注释和扩展阅读 236
- 第9章 分级存储体系 237
 - 9.1 缓存的概念 238
 - 9.2 局部性原理 238
 - 9.3 基本术语 238
 - 9.4 多级存储层次 239
 - 9.5 缓存结构 241
 - 9.6 直接映射缓存结构 241
 - 9.6.1 缓存查找 243
 - 9.6.2 缓存项中的字段 244
 - 9.6.3 用于直接映射缓存的硬件 245
 - 9.7 流水线处理器设计的影响 247
 - 9.8 缓存读/写算法 247
 - 9.8.1 CPU对缓存的读访问 248
 - 9.8.2 CPU对缓存的写访问 248
 - 9.9 处理器流水线中的缓存缺失处理 251
 - 9.9.1 在流水线性能上缓存缺失对内存延迟的影响 252
 - 9.10 利用空间局部性提高缓存性能 253
 - 9.10.1 增加块大小对性能的影响 256
 - 9.11 灵活的布局策略 257
 - 9.11.1 全相关缓存 258
 - 9.11.2 组相关缓存 259
 - 9.11.3 组相关的极端情况 261
 - 9.12 指令和数据缓存 263
 - 9.13 降低缺失损失 264
 - 9.14 缓存替换策略 264
 - 9.15 缺失类型简要说明 266
 - 9.16 TLB和缓存整合 268
 - 9.17 缓存控制器 269
 - 9.18 虚拟索引物理标记的缓存 270
 - 9.19 缓存设计因素概述 271
 - 9.20 主存的设计因素 272

9.20.1 简单的主存	272
9.20.2 与缓存块大小相匹配的主存和总线	273
9.20.3 交错式内存	273
9.21 现代主存系统分析	274
9.21.1 页式DRAM	278
9.22 分级存储体系的性能影响	279
小结	280
现代处理器的分级存储体系（一个例子）	281
练习题	281
参考文献注释和扩展阅读	283
第10章 输入/输出和稳定性存储	284
10.1 CPU和I/O设备间的通信	284
10.1.1 设备控制器	284
10.1.2 内存映射I/O	285
10.2 程控I/O	287
10.3 DMA	288
10.4 总线	290
10.5 I/O处理器	291
10.6 设备驱动	292
10.6.1 例子	293
10.7 外围设备	295
10.8 磁盘存储器	296
10.8.1 磁盘技术的传奇故事	302
10.9 磁盘调度算法	304
10.9.1 先到先服务	305
10.9.2 最短寻道时间优先	305
10.9.3 SCAN	305
10.9.4 C-SCAN	306
10.9.5 LOOK和C-LOOK	307
10.9.6 磁盘调度总结	307
10.9.7 算法比较	308
10.10 固态硬盘	309
10.11 I/O总线和设备驱动的演化	310
10.11.1 设备驱动的动态负载	311
10.11.2 信息汇总	312
小结	314
练习题	314
参考文献注释和扩展阅读	315
第11章 文件系统	317
11.1 属性	317
11.2 在磁盘子系统上实现文件系统的设计选择	321
11.2.1 连续分配	322
11.2.2 带有溢出区域的连续分配	324
11.2.3 链接分配	324
11.2.4 文件分配表	325
11.2.5 索引分配	327
11.2.6 多级索引分配	328
11.2.7 混合索引分配	328
11.2.8 不同分配策略的比较	331
11.3 信息汇总	331
11.3.1 索引节点	336
11.4 文件系统的组件	336
11.4.1 创建、写入文件的剖析	337
11.5 各种子系统的交互	337

11.6 文件系统在物理媒介上的布局	340
11.6.1 内存中的数据结构	342
11.7 处理系统崩溃	343
11.8 其他物理媒介上的文件系统	343
11.9 现代文件系统一览	344
11.9.1 Linux	344
11.9.2 Microsoft Windows	348
小结	349
练习题	350
参考文献注释和扩展阅读	352
第12章 多线程编程与多处理器	353
12.1 为什么需要多线程	353
12.2 线程所需的编程支持	354
12.2.1 线程创建和终止	354
12.2.2 线程之间的通信	356
12.2.3 读/写冲突、竞争条件及不确定性	357
12.2.4 线程之间的同步	361
12.2.5 线程库中数据类型的内部表示	365
12.2.6 简单的编程示例	366
12.2.7 死锁和活锁	369
12.2.8 条件变量	370
12.2.9 视频处理示例的完整解决方案	373
12.2.10 解决方案的讨论	374
12.2.11 重新检查条件	375
12.3 线程函数调用和多线程编程概念总结	377
12.4 线程编程的一些注意事项	379
12.5 使用线程作为软件结构抽象	379
12.6 POSIX pthread库调用总结	379
12.7 操作系统对线程的支持	382
12.7.1 用户级线程	383
12.7.2 内核级线程	385
12.7.3 Solaris线程：一个内核级线程例子	386
12.7.4 线程和库	387
12.8 在单处理器上的多线程的硬件支持	388
12.8.1 线程创建、终止以及线程间的通信	388
12.8.2 线程之间的同步	388
12.8.3 原子的Test-and-Set指令	388
12.8.4 使用Test-and-Set指令的Lock算法	390
12.9 多处理器	391
12.9.1 页表	391
12.9.2 分级存储体系	391
12.9.3 保证原子性	393
12.10 高级话题	393
12.10.1 操作系统话题	393
12.10.2 架构话题	403
12.10.3 未来之路：多核与众核架构	412
小结	413
历史回顾	414
练习题	415
参考文献注释和扩展阅读	417
第13章 网络与网络协议基础知识	419
13.1 预备知识	419
13.2 基本术语	419
13.3 网络软件	423

13.4 协议栈	424
13.4.1 因特网协议栈	424
13.4.2 OSI模型	426
13.4.3 分层的实际问题	427
13.5 应用层	427
13.6 传输层	428
13.6.1 停止并等待协议	429
13.6.2 流水线协议	431
13.6.3 可靠的流水线协议	432
13.6.4 处理传输错误	436
13.6.5 因特网上的传输协议	437
13.6.6 传输层总结	438
13.7 网络层	439
13.7.1 路由算法	439
13.7.2 因特网寻址	444
13.7.3 网络服务模式	446
13.7.4 网络路由与转发	449
13.7.5 网络层总结	450
13.8 链路层和局域网	450
13.8.1 以太网	451
13.8.2 CSMA/CD	451
13.8.3 IEEE 802.3	453
13.8.4 无线局域网与IEEE 802.11	453
13.8.5 令牌环	454
13.8.6 其他链路层协议	456
13.9 网络硬件	456
13.10 协议栈各层之间的关系	460
13.11 用于数据包传输的数据结构	460
13.11.1 TCP/IP包头	461
13.12 消息传输时间	462
13.13 协议层功能总结	466
13.14 网络软件与操作系统	466
13.14.1 套接字库	467
13.14.2 在操作系统中实现协议栈	468
13.14.3 网络设备驱动程序	468
13.15 使用UNIX套接字进行网络编程	469
13.16 网络服务与高层协议	474
小结	475
历史回顾	475
练习题	480
参考文献注释和扩展阅读	482
第14章 尾声：旅途回顾	483
14.1 处理器设计	483
14.2 进程	483
14.3 虚拟内存系统和内存管理	483
14.4 分级存储体系	484
14.5 并行系统	484
14.6 输入/输出系统	484
14.7 永久性存储	484
14.8 网络	485
结束语	485
附录A 使用UNIX套接字进行网络编程	486
参考文献	495
索引	500

• • • • • [\(收起\)](#)

[计算机系统：系统架构与操作系统的高度集成 下载链接1](#)

标签

计算机

计算机系统

操作系统

计算机科学

体系结构

计算科学

高级语言与硬件

計算機

评论

现代操作系统的微软的Windows和苹果的Mac最终进化的目标都是在核心抽象层面的基本概念与Unix操作系统中难以区分。奠定每个程序设计语言基础的是其程序控制计算系统的模型。某些方法是纯粹的抽象，某些通过硬件来表现，其他则通过编译或解释程序。编译器基于高级编程语言的原则，目标机器的指令集和操作系统遵循的规则，经过一系列阶段产生机器代码。

对计算机软硬件的体系结构做了一个很好的入门介绍，有一定的技术细节，但不过分深入，对于非科班出身的来说，可以大致了解整个计算机的运作原理，可以花一点时间用来入门。想更深入的话，可以再找一些各主题更深入的教材来进一步学习

[计算机系统：系统架构与操作系统的高度集成_下载链接1](#)

书评

非常适合入门。如果你对程序设计有一定的了解，这本书会通过高级语言（如C）的语言特性，如循环、分支、赋值等，引出处理器的指令集如何设计，从一个非常精简的指令集出发，对指令集的二进制布局进行了介绍，引出了处理器的结构、寄存器布局、微控制体系如何根据指令集来进行...

[计算机系统：系统架构与操作系统的高度集成_下载链接1](#)