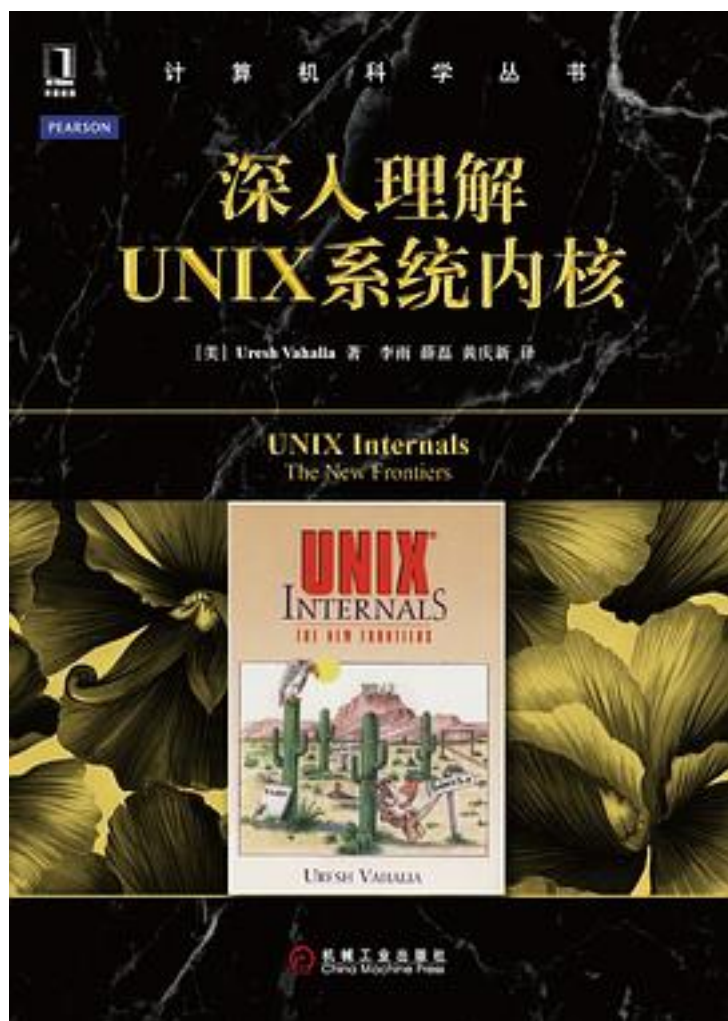


# 深入理解UNIX系统内核



[深入理解UNIX系统内核\\_下载链接1](#)

著者:Uresh Vahalia

出版者:机械工业出版社

出版时间:2015-5-1

装帧:平装

isbn:9787111491453

本书由国际资深UNIX专家撰写，深入剖析UNIX操作系统的内核技术，包含丰富的图示

与细节展示。作者从操作系统设计的角度来审视UNIX系统内核，针对内核中的每个模块，深入探讨其结构和设计，详细阐释主流UNIX系统如何选择具体模块的实现方法，以及每种方法的优缺点，为读者更好地理解操作系统内核知识、提升操作系统开发能力提供翔实指导。

全书共17章：第1章追溯UNIX系统的演变并分析影响系统主要变化的因素；第2~7章介绍进程子系统，包括线程及其在内核和用户库中的实现，以及在现代单处理器和多处理器系统中应用的同步框架；第8~11章介绍文件系统，内容涵盖知名系统的具体实现细节，以及一些使用日志提供更高可靠性和性能的高级文件系统；第12~15章介绍内存管理，涉及内核内存分配、虚拟内存等概念，以及转换后援缓冲和虚拟地址高速缓存；第16~17章主要介绍I/O子系统，内容包括设备驱动程序框架、内核与I/O子系统的交互等细节。

本书包含大量示例和案例分析，章后附有练习和参考文献，启发读者进一步思考与研究，可作为高等院校相关专业本科生或研究生的教材，也可供各类开发人员参考。

作者介绍:

目录: 目录

出版者的话

译者序

序言

前言

第1章 从头说起1

1.1 简介1

1.1.1 UNIX简史1

1.1.2 起源1

1.1.3 扩散2

1.1.4 BSD3

1.1.5 System V4

1.1.6 商业化5

1.1.7 Mach5

1.1.8 标准5

1.1.9 OSF和UI6

1.1.10 SVR4及其之后7

1.2 变革使命8

1.2.1 功能8

1.2.2 网络8

1.2.3 性能9

1.2.4 硬件变化9

1.2.5 质量提升10

1.2.6 变革10

1.2.7 其他应用程序领域11

1.2.8 小即是美11

1.2.9 灵活性12

1.3 回顾过去，展望未来13

1.3.1 UNIX系统的优点是什么13

1.3.2 UNIX系统的缺点是什么14

1.4 本书内容说明15

参考文献15

第2章 进程与内核17

2.1 简介17

|                       |    |
|-----------------------|----|
| 2.2 模式、空间和上下文         | 19 |
| 2.3 进程抽象              | 21 |
| 2.3.1 进程状态            | 21 |
| 2.3.2 进程上下文           | 23 |
| 2.3.3 用户凭据            | 23 |
| 2.3.4 u区和proc结构       | 24 |
| 2.4 执行在内核态中           | 26 |
| 2.4.1 系统调用接口          | 26 |
| 2.4.2 中断处理            | 27 |
| 2.5 同步                | 29 |
| 2.5.1 阻塞操作            | 29 |
| 2.5.2 中断              | 30 |
| 2.5.3 多处理器            | 31 |
| 2.6 进程调度              | 31 |
| 2.7 信号                | 32 |
| 2.8 新的进程和程序           | 33 |
| 2.8.1 fork和exec       | 33 |
| 2.8.2 进程的创建           | 34 |
| 2.8.3 fork的优化         | 35 |
| 2.8.4 调用新的程序          | 35 |
| 2.8.5 进程终止            | 37 |
| 2.8.6 等待进程终止          | 37 |
| 2.8.7 僵死进程            | 38 |
| 2.9 小结                | 39 |
| 2.10 练习题              | 39 |
| 参考文献                  | 39 |
| 第3章 线程和轻量级进程          | 41 |
| 3.1 简介                | 41 |
| 3.1.1 动机              | 41 |
| 3.1.2 多线程和多处理器        | 42 |
| 3.1.3 并发和并行           | 43 |
| 3.2 基本抽象              | 44 |
| 3.2.1 内核线程            | 45 |
| 3.2.2 轻量级进程           | 45 |
| 3.2.3 用户线程            | 46 |
| 3.3 轻量级线程设计时要考虑的问题    | 49 |
| 3.3.1 fork的语义         | 49 |
| 3.3.2 其他系统调用          | 50 |
| 3.3.3 信号传递和处理         | 50 |
| 3.3.4 可见性             | 51 |
| 3.3.5 栈增长             | 51 |
| 3.4 用户级别的线程库          | 51 |
| 3.4.1 编程接口            | 52 |
| 3.4.2 线程库的实现          | 52 |
| 3.5 调度器激活             | 53 |
| 3.6 Solaris和SVR4上的多线程 | 54 |
| 3.6.1 内核线程            | 54 |
| 3.6.2 轻量级进程的实现        | 55 |
| 3.6.3 用户线程            | 56 |
| 3.6.4 用户线程的实现         | 56 |
| 3.6.5 中断处理            | 57 |
| 3.6.6 系统调用处理          | 58 |
| 3.7 Mach的线程           | 58 |
| 3.7.1 Mach抽象：任务和线程    | 58 |

- 3.7.2 Mach的C-threads59
- 3.8 Digital UNIX60
  - 3.8.1 UNIX接口60
  - 3.8.2 系统调用和信号62
  - 3.8.3 pthreads库62
- 3.9 Mach 3.0的continuation63
  - 3.9.1 编程模型63
  - 3.9.2 使用continuation63
  - 3.9.3 优化65
  - 3.9.4 分析65
- 3.10 小结65
- 3.11 练习题66
- 参考文献66
- 第4章 信号和会话管理68
  - 4.1 简介68
  - 4.2 信号生成和处理68
    - 4.2.1 信号处理69
    - 4.2.2 信号的生成71
    - 4.2.3 典型场景72
    - 4.2.4 睡眠与信号72
  - 4.3 不可靠的信号73
  - 4.4 可靠的信号74
    - 4.4.1 主要特性74
    - 4.4.2 SVR3实现75
    - 4.4.3 BSD信号管理75
  - 4.5 SVR4上的信号76
  - 4.6 信号的实现77
    - 4.6.1 信号生成78
    - 4.6.2 交付和处理78
  - 4.7 异常79
  - 4.8 Mach的异常处理79
    - 4.8.1 异常端口80
    - 4.8.2 错误处理80
    - 4.8.3 调试器交互81
    - 4.8.4 分析81
  - 4.9 进程组和终端管理82
    - 4.9.1 基本概念82
    - 4.9.2 SVR3模型83
    - 4.9.3 限制84
    - 4.9.4 4.3BSD的进程组和终端84
    - 4.9.5 缺点86
  - 4.10 SVR4的会话体系结构86
    - 4.10.1 动机87
    - 4.10.2 会话和进程组87
    - 4.10.3 数据结构88
    - 4.10.4 控制终端89
    - 4.10.5 4.4BSD的会话实现机制89
  - 4.11 小结90
  - 4.12 练习题90
  - 参考文献91
- 第5章 进程调度92
  - 5.1 简介92
  - 5.2 时钟中断处理93
    - 5.2.1 callout93

- 5.2.2 告警95
- 5.3 调度器目标95
- 5.4 传统的UNIX调度96
  - 5.4.1 进程优先级97
  - 5.4.2 调度器的实现98
  - 5.4.3 运行队列的操作99
  - 5.4.4 分析99
- 5.5 SVR4调度器100
  - 5.5.1 类无关层100
  - 5.5.2 调度类的接口101
  - 5.5.3 分时类103
  - 5.5.4 实时类104
  - 5.5.5 priocntl系统调用105
  - 5.5.6 分析106
- 5.6 Solaris 2.x调度的改善107
  - 5.6.1 可抢占的内核107
  - 5.6.2 多处理器的支持107
  - 5.6.3 隐式调度108
  - 5.6.4 优先级反转109
  - 5.6.5 优先级继承的实现110
  - 5.6.6 优先级继承的局限性112
  - 5.6.7 turnstile113
  - 5.6.8 分析113
- 5.7 Mach上的调度113
  - 多处理器支持114
- 5.8 Digital UNIX的实时调度116
  - 多处理器支持116
- 5.9 其他调度实现117
  - 5.9.1 公平调度方法117
  - 5.9.2 最终期限驱动调度方法117
  - 5.9.3 三级调度器118
- 5.10 小结119
- 5.11 练习题119
- 参考文献120
- 第6章 进程间通信121
  - 6.1 简介121
  - 6.2 通用的IPC方法121
    - 6.2.1 信号122
    - 6.2.2 管道122
    - 6.2.3 SVR4管道124
    - 6.2.4 进程跟踪124
  - 6.3 System V IPC126
    - 6.3.1 公共元素126
    - 6.3.2 信号量127
    - 6.3.3 消息队列130
    - 6.3.4 共享内存131
    - 6.3.5 讨论133
  - 6.4 Mach IPC133
  - 基本概念134
  - 6.5 消息135
    - 6.5.1 消息数据结构135
    - 6.5.2 消息传递接口136
  - 6.6 端口137
    - 6.6.1 端口命名空间137

- 6.6.2 端口数据结构138
- 6.6.3 端口转换138
- 6.7 消息传递139
  - 6.7.1 转换端口权利140
  - 6.7.2 out-of-line内存141
  - 6.7.3 控制流142
  - 6.7.4 通知143
- 6.8 端口操作143
  - 6.8.1 销毁端口143
  - 6.8.2 备份端口144
  - 6.8.3 端口集合144
  - 6.8.4 端口插补145
- 6.9 扩展性145
- 6.10 Mach 3.0的增强146
  - 6.10.1 一次性的发送权利147
  - 6.10.2 Mach 3.0的通知147
  - 6.10.3 发送权利的用户引用计数148
- 6.11 讨论148
- 6.12 小结149
- 6.13 练习题149
- 参考文献149
- 第7章 同步和多处理器151
  - 7.1 简介151
  - 7.2 传统UNIX内核里的同步机制152
    - 7.2.1 中断屏蔽152
    - 7.2.2 睡眠和唤醒152
    - 7.2.3 传统方法的局限性153
  - 7.3 多处理器系统154
    - 7.3.1 内存模型154
    - 7.3.2 同步支持155
    - 7.3.3 软件体系架构156
  - 7.4 多处理器的同步问题157
    - 7.4.1 唤醒丢失问题157
    - 7.4.2 惊群问题158
  - 7.5 信号量158
    - 7.5.1 信号量提供互斥操作159
    - 7.5.2 使用信号量提供事件等待159
    - 7.5.3 使用信号量来控制可计数的资源160
    - 7.5.4 信号量的缺点160
    - 7.5.5 Convoy161
  - 7.6 自旋锁162
  - 7.7 条件变量163
    - 7.7.1 实现问题164
    - 7.7.2 事件165
    - 7.7.3 阻塞锁165
  - 7.8 读写锁165
    - 7.8.1 设计考虑165
    - 7.8.2 实现166
  - 7.9 引用计数167
  - 7.10 其他考虑168
    - 7.10.1 死锁避免168
    - 7.10.2 递归锁169
    - 7.10.3 阻塞还是自旋170
    - 7.10.4 锁什么170

|                     |     |
|---------------------|-----|
| 7.10.5 粒度和持续时间      | 170 |
| 7.11 案例研究           | 171 |
| 7.11.1 SVR4.2/MP    | 171 |
| 7.11.2 Digital UNIX | 172 |
| 7.11.3 其他实现         | 173 |
| 7.12 小结             | 174 |
| 7.13 练习题            | 174 |
| 参考文献                | 175 |
| 第8章 文件系统接口和框架       | 176 |
| 8.1 简介              | 176 |
| 8.2 文件的用户接口         | 176 |
| 8.2.1 文件和目录         | 177 |
| 8.2.2 文件属性          | 178 |
| 8.2.3 文件描述符         | 179 |
| 8.2.4 文件I/O         | 181 |
| 8.2.5 分散-聚集I/O      | 182 |
| 8.2.6 文件锁机制         | 183 |
| 8.3 文件系统            | 183 |
| 8.4 特殊文件            | 184 |
| 8.4.1 符号链接          | 184 |
| 8.4.2 管道和FIFO       | 185 |
| 8.5 文件系统框架          | 186 |
| 8.6 Vnode/Vfs架构     | 187 |
| 8.6.1 目标            | 187 |
| 8.6.2 从设备I/O得到的注解   | 187 |
| 8.6.3 vnode/vfs接口概览 | 190 |
| 8.7 实现概览            | 191 |
| 8.7.1 目标            | 191 |
| 8.7.2 Vnodes以及打开文件  | 191 |
| 8.7.3 Vnode         | 192 |
| 8.7.4 Vnode引用计数     | 193 |
| 8.7.5 Vfs对象         | 194 |
| 8.8 文件系统相关对象        | 194 |
| 8.8.1 每个文件的私有数据     | 194 |
| 8.8.2 vnodeops结构    | 195 |
| 8.8.3 vfs层中文件系统相关部分 | 196 |
| 8.9 挂载文件系统          | 196 |
| 8.9.1 虚拟文件系统转换表     | 196 |
| 8.9.2 mount函数实现     | 197 |
| 8.9.3 VFS_MOUNT过程   | 197 |
| 8.10 文件操作           | 197 |
| 8.10.1 路径遍历         | 198 |
| 8.10.2 目录名查找缓存      | 199 |
| 8.10.3 VOP_LOOKUP操作 | 199 |
| 8.10.4 打开文件         | 200 |
| 8.10.5 文件I/O        | 201 |
| 8.10.6 文件属性         | 201 |
| 8.10.7 用户凭据         | 201 |
| 8.11 分析             | 202 |
| 8.11.1 SVR4系统实现的缺点  | 202 |
| 8.11.2 4.4BSD模型     | 203 |
| 8.11.3 OSF/1方法      | 204 |
| 8.12 小结             | 205 |
| 8.13 练习题            | 205 |

|                         |     |
|-------------------------|-----|
| 参考文献                    | 206 |
| 第9章 文件系统的实现             | 207 |
| 9.1 简介                  | 207 |
| 9.2 System V文件系统 (s5fs) | 208 |
| 9.2.1 目录                | 208 |
| 9.2.2 inode             | 209 |
| 9.2.3 超级块               | 210 |
| 9.3 s5fs内核组织            | 211 |
| 9.3.1 内存inode           | 211 |
| 9.3.2 inode查找           | 212 |
| 9.3.3 文件I/O             | 213 |
| 9.3.4 inode的分配和回收       | 214 |
| 9.4 s5fs的分析             | 215 |
| 9.5 伯克利快速文件系统 (FFS)     | 216 |
| 9.6 硬盘结构                | 216 |
| 9.7 磁盘组织                | 216 |
| 9.7.1 块和片段              | 217 |
| 9.7.2 分配策略              | 218 |
| 9.8 FFS的增强功能            | 219 |
| 9.9 分析                  | 220 |
| 9.10 临时文件系统             | 221 |
| 9.10.1 内存文件系统           | 221 |
| 9.10.2 tmpfs文件系统        | 222 |
| 9.11 特殊用途文件系统           | 223 |
| 9.11.1 specfs文件系统       | 223 |
| 9.11.2 /proc文件系统        | 223 |
| 9.11.3 处理器文件系统          | 225 |
| 9.11.4 Trans lucent文件系统 | 225 |
| 9.12 旧的缓冲区缓存            | 226 |
| 9.12.1 基本操作             | 227 |
| 9.12.2 缓冲区头结构           | 228 |
| 9.12.3 优点               | 228 |
| 9.12.4 缺点               | 228 |
| 9.12.5 保证文件系统的一致性       | 229 |
| 9.13 小结                 | 230 |
| 9.14 练习题                | 230 |
| 参考文献                    | 231 |
| 第10章 分布式文件系统            | 232 |
| 10.1 简介                 | 232 |
| 10.2 分布式文件系统的一般特征       | 232 |
| 10.3 网络文件系统             | 233 |
| 10.3.1 用户视角             | 234 |
| 10.3.2 设计目标             | 235 |
| 10.3.3 NFS的组件           | 235 |
| 10.3.4 无状态设计            | 237 |
| 10.4 NFS协议集             | 238 |
| 10.4.1 外部数据表示           | 238 |
| 10.4.2 远程过程调用           | 239 |
| 10.5 NFS实现              | 240 |
| 10.5.1 控制流              | 240 |
| 10.5.2 文件句柄             | 241 |
| 10.5.3 挂载操作             | 241 |
| 10.5.4 路径名的查找           | 242 |
| 10.6 UNIX语义             | 243 |

|                                            |      |
|--------------------------------------------|------|
| 10.6.1 打开文件许可                              | 243  |
| 10.6.2 已打开文件的删除                            | 243  |
| 10.6.3 读写操作                                | 244  |
| 10.7 NFS性能                                 | 244  |
| 10.7.1 性能瓶颈                                | 244  |
| 10.7.2 客户端缓存                               | 244  |
| 10.7.3 延迟写                                 | 245  |
| 10.7.4 重传缓存                                | 246  |
| 10.8 专用NFS服务器                              | 247  |
| 10.8.1 Auspex的Functional Multiprocessor 架构 | 247  |
| 10.8.2 IBM的HA-NFS服务器                       | 248  |
| 10.9 NFS安全                                 | 249  |
| 10.9.1 NFS访问控制                             | 249  |
| 10.9.2 UID重映射                              | 250  |
| 10.9.3 根用户重映射                              | 250  |
| 10.10 NFS版本                                | 3251 |
| 10.11 远程文件共享                               | 252  |
| 10.12 RFS架构                                | 252  |
| 10.12.1 远程消息协议                             | 253  |
| 10.12.2 有状态操作                              | 253  |
| 10.13 RFS实现                                | 254  |
| 10.13.1 远程挂载                               | 254  |
| 10.13.2 RFS客户端和服务端                         | 255  |
| 10.13.3 崩溃恢复                               | 255  |
| 10.13.4 其他问题                               | 256  |
| 10.14 客户端缓存                                | 256  |
| 10.15 Andrew文件系统                           | 258  |
| 10.15.1 可伸缩架构                              | 258  |
| 10.15.2 存储和命名空间的组织                         | 259  |
| 10.15.3 会话级语义                              | 260  |
| 10.16 AFS实现                                | 260  |
| 10.16.1 缓存与一致性                             | 261  |
| 10.16.2 路径名查找                              | 261  |
| 10.16.3 安全性                                | 262  |
| 10.17 AFS的不足                               | 262  |
| 10.18 DCE的分布式文件系统                          | 263  |
| 10.18.1 DFS架构                              | 263  |
| 10.18.2 缓存一致性                              | 264  |
| 10.18.3 令牌管理器                              | 265  |
| 10.18.4 DFS的其他服务                           | 266  |
| 10.18.5 分析                                 | 266  |
| 10.19 小结                                   | 267  |
| 10.20 练习题                                  | 267  |
| 参考文献                                       | 268  |
| 第11章 高级文件系统                                | 271  |
| 11.1 简介                                    | 271  |
| 11.2 传统文件系统的局限                             | 271  |
| 11.2.1 FFS磁盘布局                             | 272  |
| 11.2.2 磁盘上写操作的主导                           | 273  |
| 11.2.3 元数据更新                               | 274  |
| 11.2.4 故障修复                                | 274  |
| 11.3 文件系统簇 (SUN-FFS)                       | 275  |
| 11.4 日志方法                                  | 276  |
| 11.5 日志结构文件系统                              | 277  |

|                           |     |
|---------------------------|-----|
| 11.6 4.4BSD日志结构文件系统       | 277 |
| 11.6.1 日志写入               | 278 |
| 11.6.2 数据检索               | 279 |
| 11.6.3 崩溃恢复               | 279 |
| 11.6.4 cleaner进程          | 280 |
| 11.6.5 分析                 | 280 |
| 11.7 元数据日志                | 281 |
| 11.7.1 正常操作               | 282 |
| 11.7.2 日志的一致性             | 283 |
| 11.7.3 崩溃恢复               | 284 |
| 11.7.4 分析                 | 284 |
| 11.8 Episode文件系统          | 285 |
| 11.8.1 基本抽象               | 285 |
| 11.8.2 结构                 | 286 |
| 11.8.3 日志记录               | 287 |
| 11.8.4 其他特性               | 287 |
| 11.9 “看门狗”监视器             | 288 |
| 11.9.1 目录的“看门狗”进程         | 289 |
| 11.9.2 消息通道               | 289 |
| 11.9.3 “看门狗”进程的应用         | 290 |
| 11.10 4.4BSD的portal文件系统   | 290 |
| 11.11 可堆叠文件系统层次           | 291 |
| 11.11.1 框架和接口             | 292 |
| 11.11.2 SunSoft 原型        | 293 |
| 11.12 4.4BSD文件系统接口        | 294 |
| 11.13 小结                  | 295 |
| 11.14 练习题                 | 295 |
| 参考文献                      | 296 |
| 第12章 内核内存分配               | 298 |
| 12.1 简介                   | 298 |
| 12.2 功能需求                 | 299 |
| 12.3 资源映射分配器              | 301 |
| 12.4 简单的幂空闲链表分配器          | 303 |
| 12.5 McKusick-Karels分配器   | 305 |
| 12.6 伙伴系统                 | 307 |
| 12.7 SVR4的惰性伙伴算法          | 308 |
| 12.7.1 惰性合并               | 309 |
| 12.7.2 SVR4的实现细节          | 310 |
| 12.8 Mach和OSF/1的区块分配器     | 310 |
| 12.8.1 垃圾回收               | 311 |
| 12.8.2 分析                 | 312 |
| 12.9 一种针对多处理器系统的分层式分配器    | 312 |
| 12.10 Solaris 2.4的Slab分配器 | 314 |
| 12.10.1 复用对象              | 314 |
| 12.10.2 利用硬件缓存            | 315 |
| 12.10.3 分配器足迹             | 315 |
| 12.10.4 设计与接口             | 316 |
| 12.10.5 实现细节              | 317 |
| 12.10.6 分析                | 318 |
| 12.11 小结                  | 318 |
| 12.12 练习题                 | 319 |
| 参考文献                      | 320 |
| 第13章 虚拟内存                 | 321 |
| 13.1 简介                   | 321 |

|                     |     |
|---------------------|-----|
| 13.2 按需分页           | 324 |
| 13.2.1 功能需求         | 324 |
| 13.2.2 虚拟地址空间       | 325 |
| 13.2.3 页面的首次访问      | 326 |
| 13.2.4 交换区          | 326 |
| 13.2.5 转换映射         | 327 |
| 13.2.6 页面替换策略       | 328 |
| 13.3 对硬件的需求         | 328 |
| 13.3.1 MMU缓存        | 330 |
| 13.3.2 Intel 80x86  | 331 |
| 13.3.3 IBM RS/6000  | 333 |
| 13.3.4 MIPS R3000   | 335 |
| 13.4 4.3BSD—案例研究    | 336 |
| 13.4.1 物理内存         | 337 |
| 13.4.2 地址空间         | 338 |
| 13.4.3 页在哪里         | 339 |
| 13.4.4 交换空间         | 340 |
| 13.5 4.3 BSD 内存管理操作 | 341 |
| 13.5.1 创建进程         | 341 |
| 13.5.2 缺页异常处理       | 342 |
| 13.5.3 空闲页面链表       | 344 |
| 13.5.4 交换           | 345 |
| 13.6 分析             | 346 |
| 13.7 练习题            | 347 |
| 参考文献                | 348 |
| 第14章 SVR4 VM架构      | 349 |
| 14.1 简介             | 349 |
| 14.2 内存映射文件         | 349 |
| 14.3 VM的设计理念        | 351 |
| 14.4 基础抽象           | 352 |
| 14.4.1 物理内存         | 353 |
| 14.4.2 地址空间         | 353 |
| 14.4.3 地址映射         | 354 |
| 14.4.4 匿名页          | 355 |
| 14.4.5 硬件地址转换       | 356 |
| 14.5 段驱动程序          | 357 |
| 14.5.1 seg_vn       | 357 |
| 14.5.2 seg_map      | 358 |
| 14.5.3 seg_dev      | 359 |
| 14.5.4 seg_kmem     | 359 |
| 14.5.5 seg_kp       | 359 |
| 14.6 交换层            | 359 |
| 14.7 VM操作           | 361 |
| 14.7.1 创建新映射        | 361 |
| 14.7.2 匿名页处理        | 361 |
| 14.7.3 创建进程         | 363 |
| 14.7.4 共享匿名页        | 364 |
| 14.7.5 处理缺页异常       | 364 |
| 14.7.6 共享内存         | 365 |
| 14.7.7 其他组件         | 366 |
| 14.8 与vnode子系统的交互   | 367 |
| 14.8.1 对vnode接口的修改  | 367 |
| 14.8.2 统一文件访问机制     | 368 |
| 14.8.3 其他细节         | 370 |

- 14.9 Solaris的虚拟交换空间370
  - 14.9.1 交换空间扩展370
  - 14.9.2 虚拟交换管理371
  - 14.9.3 讨论372
- 14.10 分析372
- 14.11 性能改进374
  - 14.11.1 缺页异常率偏高的原因374
  - 14.11.2 SVR4对SunOS VM实现的改进375
  - 14.11.3 结果与讨论375
- 14.12 小结376
- 14.13 练习题376
- 参考文献377
- 第15章 其他内存管理技术378
  - 15.1 简介378
  - 15.2 Mach的内存管理设计378
    - 15.2.1 设计目标378
    - 15.2.2 对外接口379
    - 15.2.3 基础抽象380
  - 15.3 内存共享机制381
    - 15.3.1 写时复制共享382
    - 15.3.2 读写共享 383
  - 15.4 内存对象与Pager384
    - 15.4.1 初始化内存对象384
    - 15.4.2 内核与pager的接口384
    - 15.4.3 内核与Pager的交互385
  - 15.5 外部pager和内部pager386
  - 15.6 页面替换388
  - 15.7 分析389
  - 15.8 4.4BSD的内存管理390
  - 15.9 旁路转换缓冲区的一致性391
    - 15.9.1 单处理器上的TLB一致性392
    - 15.9.2 多处理器问题393
  - 15.10 Mach中的TLB击落算法394
    - 15.10.1 同步与死锁的避免395
    - 15.10.2 讨论395
  - 15.11 SVR4和SVR4.2 UNIX中的TLB一致性396
    - 15.11.1 SVR4/MP396
    - 15.11.2 SVR4.2/MP397
    - 15.11.3 惰性击落算法397
    - 15.11.4 立即击落算法398
    - 15.11.5 讨论399
  - 15.12 其他TLB一致性算法399
  - 15.13 虚拟地址缓存400
    - 15.13.1 修改映射401
    - 15.13.2 地址别名402
    - 15.13.3 DMA操作402
    - 15.13.4 维护缓存一致性403
    - 15.13.5 分析404
  - 15.14 练习题404
  - 参考文献405
- 第16章 设备驱动和I/O407
  - 16.1 简介407
  - 16.2 概述407
    - 16.2.1 硬件配置408

- 16.2.2 设备中断409
- 16.3 设备驱动程序框架410
  - 16.3.1 设备和驱动的分类410
  - 16.3.2 调用驱动程序代码411
  - 16.3.3 设备转换表412
  - 16.3.4 驱动入口点函数413
- 16.4 输入输出 (I/O) 子系统414
  - 16.4.1 主设备号和从设备号415
  - 16.4.2 设备文件416
  - 16.4.3 specfs文件系统417
  - 16.4.4 通用snode结构418
  - 16.4.5 设备克隆419
  - 16.4.6 针对字符设备的I/O操作420
- 16.5 poll系统调用420
  - 16.5.1 poll系统调用的实现421
  - 16.5.2 BSD 4.3系统select系统调用422
- 16.6 块设备I/O操作423
  - 16.6.1 buf数据结构424
  - 16.6.2 与vnode结构的交互425
  - 16.6.3 设备访问方法425
  - 16.6.4 块设备的裸I/O427
- 16.7 DDI/DKI规范428
  - 16.7.1 一般建议429
  - 16.7.2 Section 3函数429
  - 16.7.3 其他种类接口430
- 16.8 更新的SVR4发行版431
  - 16.8.1 多处理器安全的驱动431
  - 16.8.2 SVR4.1/ES的变化432
  - 16.8.3 动态加载和卸载432
- 16.9 未来方向434
- 16.10 小结435
- 16.11 练习题435
- 参考文献436
- 第17章 STREAMS437
  - 17.1 动机437
  - 17.2 概述438
  - 17.3 消息和队列439
    - 17.3.1 消息440
    - 17.3.2 虚拟复制440
    - 17.3.3 消息类型441
    - 17.3.4 队列和模块442
  - 17.4 流I/O443
    - 17.4.1 STREAMS调度程序444
    - 17.4.2 优先级组445
    - 17.4.3 流量控制445
    - 17.4.4 驱动程序尾部446
    - 17.4.5 流头447
  - 17.5 配置和设置448
    - 17.5.1 配置一个STREAMS模块或驱动程序448
    - 17.5.2 打开流449
    - 17.5.3 推送一个模块到流上451
    - 17.5.4 克隆设备451
  - 17.6 STREAMS的ioctl命令452
    - 17.6.1 带有I\_STR的ioctl命令处理452

17.6.2 透明的ioctl命令453  
17.7 内存分配453  
17.8 多路复用455  
17.8.1 上部多路复用器455  
17.8.2 下部多路复用器456  
17.8.3 链接流456  
17.8.4 数据流458  
17.8.5 普通链接和持久链接458  
17.9 FiFO和管道459  
17.9.1 FiFO中STREAMS的应用459  
17.9.2 使用STREAMS的管道460  
17.10 网络接口461  
17.10.1 传输提供者接口461  
17.10.2 传输层接口462  
17.10.3 socket463  
17.10.4 SVR4的socket实现464  
17.11 小结464  
17.12 练习题465  
参考文献466  
• • • • • ([收起](#))

[深入理解UNIX系统内核\\_下载链接1](#)

标签

UNIX

操作系统

内核

经典

linux

计算机系统

网络编程

计算机-操作系统

# 评论

-----  
[深入理解UNIX系统内核\\_下载链接1](#)

# 书评

这本书讲解了所有常见Unix系统内核的设计，其内容极为广泛，包括但不限于进程/线程模型与调度，IPC，文件系统，内存管理等等各种在OS设计时遇到的问题。由于涵盖的主题比较多，深度上就很难兼顾，但这只是相对于其广度来说的，客观的说，虽然缺乏系统的具体细节但是对整体结构...

-----  
[深入理解UNIX系统内核\\_下载链接1](#)