

# 《Unix内核源码剖析》

## 图书基本信息

书名：《Unix内核源码剖析》

13位ISBN编号：9787115345219

10位ISBN编号：711534521X

出版时间：2014-3-1

出版社：人民邮电出版社

作者：青柳隆宏

页数：342

译者：殷中翔

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)

# 《Unix内核源码剖析》

## 内容概要

### 为什么要阅读内核源代码

- 对计算机系统的全貌有更深入的了解
- 对学到的算法和思路举一反三
- 加深对操作系统的理解
- 提升自身的技术水平

### 为什么选择UNIX V6？

- 代码行数约为1 万行
- 有充实的资料可参考
- 网罗了操作系统的基本功能
- 简化的设计
- 便于读者对系统有完整的了解
- 有模拟器可供参考

本书是一本Unix内核源代码的阅读指南。作者结合UNIX V6已公开的相关文档，对其内核源码进行详细剖析，旨在让读者更深入地理解进程、中断、块I/O系统、文件系统、字符I/O系统、启动系统等操作系统的基本原理。

本书适合操作系统的初中级学习者阅读，特别适合通过大学课程和其他入门书对操作系统有所了解，但是对具体细节缺乏深入理解的读者，以及那些对操作系统的具体实现有兴趣的读者。

# 《Unix内核源码剖析》

## 作者简介

作者简介：

青柳隆宏

日本资深程序员，现在硅谷从事计算机处理芯片的设计工作。曾设计了超级计算机“京”的L1缓存，该计算机在2011年排名世界Top 500超级计算机的第1位，也是人类历史上第一台计算能力跨越1亿亿次每秒的计算系统。作者Email：v6@gachapin.jp。

译者简介：

殷中翔

日本国立九州大学系统信息专业硕士学位。具有十余年软件开发经验，现主要从事企业Web系统服务器端的开发及维护工作。

黄炎

同济大学本科毕业，现供职于上海爱可生，从事数据库高可用软件开发。大学有幸师从陈闾中教授，一窥UNIX V6代码的门径。

周金杰

同济大学硕士，法国国立高等电信学院（TELECOM PARIS）硕士。曾通读并研究UNIX V6源代码，同时参与了面向对象操作系统的移植以及操作系统调试器的开发工作，熟悉系统底层相关的技术知识。

崇尚设计，享受编程给生活带来的乐趣。

## 书籍目录

### 第I部分

什么是UNIX V6	1
第1章 UNIX V6的全貌	2
1.1 什么是UNIX V6	2
1.2 UNIX的历史	2
1.3 UNIX V6内核	4
1.4 构成UNIX V6运行环境的硬件	4
PDP-11	5
1.5 代码	9
1.6 手册	9
1.7 小结	10

### 第II部分

进程	11
第2章 进程	12
2.1 进程的概要	12
什么是进程	12
进程的并行执行	12
进程的执行状态	14
用户模式和内核模式	14
交换处理	15
2.2 proc结构体和user结构体	15
proc结构体	16
user结构体	17
2.3 为进程分配的内存	21
代码段	21
数据段	21
虚拟地址空间	22
变换地址	24
2.4 小结	26
第3章 进程的管理I	27
3.1 进程的生命周期	27
3.2 创建进程	28
进程的复制	28
父进程和子进程	29
系统调用fork	29
newproc()	32
panic()	36
3.3 切换执行进程	37
中断执行进程	37
进程的执行状态	37
选择执行进程的算法	38
上下文切换	39
系统调用wait	39
sleep()	39
swtch()	41
swtch() 的返回位置	44
setpri()	51

wakeup()	51
setrun()	52
3.4 执行程序	53
程序执行文件的格式	53
系统调用exec	54
estabur()	62
sureg()	65
expand()	66
3.5 进程的终止	68
系统调用exit	69
系统调用wait	71
3.6 数据区域的扩展	73
系统调用break	73
3.7 管理内存和交换空间	76
map结构体	76
获取未使用区域	77
释放区域	79
3.8 小结	81
第4章 交换处理	82
4.1 什么是交换处理	82
代码段和数据段	82
sched()	83
xswap()	87
4.2 共享代码段的处理	88
xalloc()	90
xfree()	93
xccdec()	93
4.3 小结	94
第III 部分	
中断	95
第5章 中断与陷入	96
5.1 什么是中断与陷入	96
什么是中断	96
什么是陷入	97
5.2 优先级与向量 ( Vector )	98
中断优先级和处理器优先级	98
中断和陷入向量	100
5.3 中断和陷入的处理流程	100
发生中断或陷入	101
执行call和trap	104
5.4 时钟中断处理函数	107
时钟设备的规格	107
时钟中断处理函数的内容	108
clock()	113
5.5 陷入处理函数	117
trap()	117
grow()	122
5.6 系统调用的处理流程	123
传递参数的方法	123

sysent结构体	124
trap()	126
5.7 小结	128
第6章 信号	129
6.1 什么是信号	129
信号的发送方法	129
确认接收信号	129
信号的种类	130
ssig()	131
kill()	132
signal()	133
psignal()	133
issig()	134
psig()	134
core()	136
在系统调用处理中处理信号	136
6.2 跟踪功能	137
什么是跟踪	137
ipc结构体	138
跟踪的处理流程	138
stop()	139
ptrace()	140
procxmt()	142
wait()	143
6.3 小结	144
第IV 部分	
块I/O 系统	145
第7章 块设备子系统	146
7.1 设备的基础	146
设备的种类	146
设备驱动	146
类别和设备编号	147
特殊文件	147
7.2 块设备子系统	148
缓冲区	148
b-list和av-list	150
RAW 输入输出	151
7.3 缓冲区的初始化	152
binit()	152
clrbuf()	153
7.4 缓冲区的获取和释放	154
getblk()	154
notavail()	156
brelse()	157
7.5 读取	158
读取的种类	158
bread()	159
iowait()	160
iodone()	160

geterror()	161
breada()	161
incore()	162
7.6 写入	163
写入的种类	163
bwrite()	164
bawrite()	165
bdwrite()	165
bflush()	166
7.7 RAW输入输出	167
physio()	167
swap()	169
7.8 小结	171
第8章 块设备驱动	172
8.1 什么是块设备驱动	172
块设备驱动表	172
设备处理队列	173
处理流程	173
8.2 RK-11磁盘驱动	174
RK11-D	175
特殊文件	175
设定bdevsw[]	176
中断处理函数	176
RK11-D 的寄存器	177
rkstrategy()	179
rkstart()	180
rkaddr()	180
devstart()	181
rkintr()	182
RAW 输入输出	184
8.3 小结	184
第V 部分	
文件系统	185
第9章 文件系统	186
9.1 什么是文件系统	186
inode	186
树状结构的命名空间	187
挂载	187
访问权限	188
根磁盘	189
9.2 块设备的区域	189
用于启动的区域	190
超级块	190
inode区域	191
存储区域	193
9.3 挂载	193
mount结构体	193
系统调用mount	194
getmdev()	196

系统调用umount	197
9.4 inode的获取和释放	198
inode[]	198
iget()	201
iput()	203
iupdat()	204
9.5 inode与存储区域的对应关系	205
bmap()	208
itrunc()	211
9.6 分配块设备中的块	213
ialloc()	213
ifree()	216
alloc()	217
free()	220
getfs()	222
badblock()	222
9.7 将路径变为inode	223
目录的内容	223
namei()	224
access()	230
9.8 初始化与同步	232
iinit()	232
update()	232
9.9 小结	234
第10章 文件处理	235
10.1 用户程序对文件的处理	235
10.2 3 个结构体	235
标准输入输出	237
10.3 文件的生成和打开处理	237
系统调用creat	238
maknode()	238
wdir()	239
系统调用open	240
open1()	241
falloc()	242
ufalloc()	243
openi()	243
10.4 文件的读取和写入	244
系统调用read、write	244
rdwr()	245
readi()	246
writei()	248
iomove()	250
getf()	252
10.5 指定文件的读写位置	252
系统调用seek	252
10.6 关闭文件	254
系统调用close	254
closef()	254



closei()	255
10.7 目录的生成	255
系统调用mknod	255
10.8 文件的链接	256
系统调用link	257
suser()	258
10.9 删除文件	258
系统调用unlink	258
10.10 小结	260
第11章 管道	261
11.1 什么是管道	261
使用管道的优点	262
11.2 开始管道通信	262
系统调用pipe	262
11.3 收发数据	264
writep()	264
readp()	266
plock()	267
prele()	267
11.4 结束管道通信	268
closef()	268
11.5 建立管道通信的流程	268
建立父子进程间的通信	268
系统调用dup	272
11.6 小结	273
第VI 部分	
字符I/O 系统	275
第12章 字符设备	276
12.1 字符设备驱动	276
字符设备缓冲区	277
对缓冲区的操作	278
初始化缓冲区池	280
12.2 LP11设备驱动	281
什么是LP11	281
LP11设备驱动的功能	282
lopen()	284
lpwrite()	285
lpcanon()	285
lpoutput()	287
lpstart()	288
lpint()	288
lpclose()	289
12.3 小结	289
第13章 电传终端	290
13.1 什么是电传终端	290
电传终端的接口	290
特殊文件	291
tty 结构体	292
mactab[]	295

partab[]	295
KL11/DL11	296
KL11/DL11设备驱动的规格	297
KL11/DL11设备驱动函数	298
13.2 终端的开启和关闭	298
klopen()	298
klclose()	300
wflushtty()	300
flushtty()	301
13.3 终端的设定	302
gtty()	302
stty()	302
sgtty()	303
klsgtty()	304
ttystty()	304
13.4 从终端输入文字	305
klrint()	306
ttyinput()	306
13.5 读取输入的数据	308
klread()	309
ttread()	309
canon()	310
13.6 向终端输出数据	313
klwrite()	314
ttwrite()	315
ttyoutput()	316
ttstart()	318
ttrstrt()	319
klxint()	320
13.7 小结	320
第VII 部分	
启动系统	321
第14章 启动系统	322
14.1 启动的流程	322
start	323
main()	326
/etc/init	329
14.2 小结	330
附录 参考资料等	331
A.1 参考文献、网站	331
A.2 pre K&R C	335
后记	337
索引	338

# 《Unix内核源码剖析》

## 精彩短评

- 1、前面写得还可以，后面感觉有点慌慌张张尽快结束。
- 2、内核源码的入门书，V6的1975年的架构，1万行，平民os吧
- 3、匆匆翻了一遍，还是FreeBSD内核博大深奥啊
- 4、还是不错的，页数很少，简短，和那本《深入Linux内核架构》相比真是博客般的存在，在地铁上就可以轻松读完的读报类型的书，值得小读一下。
- 5、讲的太简单了，没用。还是老版的pre-c语言，没救了。建议分析MIT xv6源码。不看了，没意思。
- 6、现在的水平读，只能粗略的翻过。还不能细读。
- 7、不太能看懂 但零零散散的算是看得差不多了
- 8、看不懂 所以要差评 哈哈
- 9、基本每一段代码都有详细的注释,有些讲得不是特别清楚的地方可以结合《返璞归真:UNIX技术内幕》以及《unix操作系统教程》看
- 10、因为好奇，走上不归路；读完释怀，回头有岸

# 《Unix内核源码剖析》

## 精彩书评

- 1、虽然翻译审稿君很用心，看了前4章还是发现了些漏错。如果能针对大比例的不写内核程序的程序猿写些对应代码注意就更好了。抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了
- 2、在了解了操作系统大概功能之后，想看具体实现，推荐这本书。相比《understanding the Linux kernel》，这本用了更多的图表，更方便读者理解（和30天实现操作系统那本书一样，感觉日本作者确实用心写书）。前几章仔细阅读源码，看懂进程切换的精髓，后几章重点看数据结构，代码只看了看注释，不必注意细节了。几点心得：1.内核对外可以看做一个黑盒子，只提供系统调用；而内部却是相互配合的齿轮，大量运用全局变量，以及goto语句这些普通编程中本应尽量避免的做法。或许这就是水平的差别，天才摒弃偏见，将goto语句当瑞士军刀使用；而普通程序员用它写出一堆bug，然后还总结出尽量不要用goto的经验。2.内核中好多数据结构都是数组和链表的结合：数组用来分配空间，但操作起来却是用链表来实现。

# 《Unix内核源码剖析》

## 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)