HZ BOOKS

经 典 原 版 书 库

PEARSON Prentice Hall

Solaris内核结构

（英文版 · 第2版）

SECOND EDITION

Solaris Internals

SOLARIS 10 AND OPENSOLARIS KERNEL ARCHITECTURE

Richard McDougall and Jim Mauro
Foreword by Bryan Cantrill

（美） Richard McDougall
Jim Mauro 著

机械工业出版社
China Machine Press

经 典 原 版 书 库

Solaris内核结构

（英文版 · 第2版）

Solaris Internals

Solaris 10 and OpenSolaris Kernel Architecture

（美） Richard McDougall
Jim Mauro 著

机械工业出版社
China Machine Press

本书版权登记号：图字：01-2006-6893

图书在版编目（CIP）数据

Solaris内核结构（英文版·第2版）/（美）麦克杜格尔（McDougall, R.）等著. －北京：机械工业出版社，2007.1

（经典原版书库）

书名原文：Solaris Internals: Solaris 10 and OpenSolaris Kernel Architecture, Second Edition

ISBN 7-111-20418-2

Ⅰ.S… Ⅱ.麦… Ⅲ.操作系统（软件），Solaris－英文 Ⅳ.TP316.89

中国版本图书馆CIP数据核字（2006）第139956号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）

责任编辑：迟振春

北京京北制版印刷厂印刷·新华书店北京发行所发行

2007年1月第1版第1次印刷

170mm×242mm · 66.5印张

定价：99.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及庋藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权

威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T.，Stanford，U.C. Berkeley，C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzjsj@hzbook.com
联系电话：(010) 68995264
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

专家指导委员会

（按姓氏笔画顺序）

尤晋元	王　珊	冯博琴	史忠植	史美林
石教英	吕　建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范　明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅　宏	程　旭	程时端	谢希仁
裘宗燕	戴　葵			

For Traci, Madi, and Boston—
for your love, encouragement, and support . . .
—Richard

Once again . . .
For Donna, Frank, and Dominick.
All my love, always . . .
—Jim

Foreword

Over the past decade, a regrettable idea took hold: Operating systems, while interesting, were a finished, solved problem. The genesis of this idea is manifold, but the greatest contributing factor may simply be that operating systems were not understood; they were largely delivered not as transparent systems, but rather as proprietary black boxes, welded shut to even the merely curious. This is anathema to understanding; if something can't be taken apart—if its inner workings remain hidden—its intricacies can never be understood nor its engineering nuances appreciated. This is especially true of software systems, which can't even be taken apart in the traditional sense. Software is, despite the metaphors, information, not machine, and a closed software system is just about as resistant to understanding as an engineered system can be.

This was the state of Solaris circa 2000, and it was indeed not well understood. Its internals were publicly described only in arcane block comments or old USENIX papers, its behavior was opaque to existing tools, and its source code was cloistered in chambers unknown. Starting in 2000, this began to change (if slowly) —heralded in part by the first edition of the volume that you now hold in your hands: Jim Mauro and Richard McDougall's *Solaris™ Internals*. Jim and Richard had taken on an extraordinary challenge—to describe the inner workings of a system so complicated that no one person actually understands all of it. Over the course of working on their book, Jim and Richard presumably realized that no one book could contain it either. Despite scaling back their ambition to (for example) not include networking, the first edition of *Solaris™ Internals* still weighed in at over six hundred pages.

The publishing of *Solaris™ Internals* marked the beginning of change that accelerated through the first half of the decade, as the barriers to using and understanding Solaris were broken down. Solaris became free, its engineers began to talk about its implementation extensively through new media like blogs, and, most importantly, Solaris itself became open source in June 2005, becoming the first operating system to leap the chasm from proprietary to open. At the same time, the mechanics of Solaris became much more interesting as several revolutionary new technologies made their debut in Solaris 10. These technologies have swayed many a naysayer, and have proved that operating systems are alive after all. Furthermore, there are still hard, important problems to be solved.

If 2000 is viewed as the beginning of the changes in Solaris, 2005 may well be viewed as the end of the beginning. By the end of 2005, what was a seemingly finished, proprietary product had been transformed into an exciting, open source system, alive with potential and possibility. It is especially fitting that these changes are welcomed with this second edition of *Solaris™ Internals*. Faced with the impossible task of reflecting a half-decade of massive engineering change, Jim and Richard made an important decision—they enlisted the explicit help of the engineers that designed the subsystems and wrote the code. In several cases these engineers have wholly authored the chapter on their "baby." The result is a second edition that is both dramatically expanded and highly authoritative—and very much in keeping with the new Solaris zeitgeist of community development and authorship.

On a personal note, it has been rewarding to see Jim and Richard use DTrace, the technology that Mike Shapiro, Adam Leventhal, and I developed in Solaris 10. Mike, Adam, and I were all teaching assistants for our university operating systems course, and an unspoken goal of ours was to develop a pedagogical tool that would revolutionize the way that operating systems are taught. I therefore encourage you not just to read *Solaris™ Internals*, but to *download* Solaris, *run* it on your desktop or laptop or under a virtual machine, and *use* DTrace yourself to see the concepts that Jim and Richard describe—live, and on your own machine!

Be you student or professional, reading for a course, for work, or for curiosity, it is my pleasure to welcome you to your guides through the internals of Solaris. Enjoy your tour, and remember that Solaris is not a finished work, but rather a living, evolving technology. If you're interested in accelerating that evolution—or even if you just have questions on using or understanding Solaris—please join us in the many communities at `http://www.opensolaris.org`. Welcome!

Bryan Cantrill
San Francisco, California
June 2006

Preface

Welcome to the second edition of *Solaris™ Internals* and its companion volume, *Solaris™ Performance and Tools*. It has been almost five years since the release of the first edition, during which time we have had the opportunity to communicate with a great many Solaris users, software developers, system administrators, database administrators, performance analysts, and even the occasional kernel hacker. We are grateful for all the feedback, and we have made specific changes to the format and content of this edition based on reader input. Read on to learn what is different. We look forward to continued communication with the Solaris community.

About These Books

These books are about the internals of Sun's Solaris Operating System—specifically, the SunOS kernel. Other components of Solaris, such as windowing systems for desktops, are not covered. The first edition of *Solaris™ Internals* covered Solaris releases 2.5.1, 2.6, and Solaris 7. These volumes focus on Solaris 10, with updated information for Solaris 8 and 9.

In the first edition, we wanted not only to describe the internal components that make the Solaris kernel tick, but also to provide guidance on putting the information to practical use. These same goals apply to this work, with further emphasis on the use of bundled (and in some cases unbundled) tools and utilities that can be used to examine and probe a running system. Our ability to illustrate more of the

kernel's inner workings with observability tools is facilitated in no small part by the inclusion of some revolutionary and innovative technology in Solaris 10—DTrace, a dynamic kernel tracing framework. DTrace is one of many new technologies in Solaris 10, and is used extensively throughout this text.

In working on the second edition, we enlisted the help of several friends and colleagues, many of whom are part of Solaris kernel engineering. Their expertise and guidance contributed significantly to the quality and content of these books. We also found ourselves expanding topics along the way, demonstrating the use of `dtrace(1)`, `mdb(1)`, `kstat(1)`, and other bundled tools. So much so that we decided early on that some specific coverage of these tools was necessary, and chapters were written to provide readers with the required background information on the tools and utilities. From this, an entire chapter on using the tools for performance and behavior analysis evolved.

As we neared completion of the work, and began building the entire manuscript, we ran into a bit of a problem—the size. The book had grown to over 1,500 pages. This, we discovered, presented some problems in the publishing and production of the book. After some discussion with the publisher, it was decided we should break the work up into two volumes.

Solaris™ Internals. This represents an update to the first edition, including a significant amount of new material. All major kernel subsystems are included: the virtual memory (VM) system, processes and threads, the kernel dispatcher and scheduling classes, file systems and the virtual file system (VFS) framework, and core kernel facilities. New Solaris facilities for resource management are covered as well, along with a new chapter on networking. New features in Solaris 8 and Solaris 9 are called out as appropriate throughout the text. Examples of Solaris utilities and tools for performance and analysis work, described in the companion volume, are used throughout the text.

Solaris™ Performance and Tools. This book contains chapters on the tools and utilities bundled with Solaris 10: `dtrace(1)`, `mdb(1)`, `kstat(1)`, etc. There are also extensive chapters on using the tools to analyze the performance and behavior of a Solaris system.

The two texts are designed as companion volumes, and can be used in conjunction with access to the Solaris source code on

`http://www.opensolaris.org`

Readers interested in specific releases before Solaris 8 should continue to use the first edition as a reference.

Intended Audience

We believe that these books will serve as a useful reference for a variety of technical staff members working with the Solaris Operating System.

Application developers can find information in these books about how Solaris OS implements functions behind the application programming interfaces. This information helps developers understand performance, scalability, and implementation specifics of each interface when they develop Solaris applications. The system overview section and sections on scheduling, interprocess communication, and file system behavior should be the most useful sections.

Device driver and kernel module developers of drivers, STREAMS modules, loadable system calls, etc., can find herein the general architecture and implementation theory of the Solaris OS. The Solaris kernel framework and facilities portions of the books (especially the locking and synchronization primitives chapters) are particularly relevant.

Systems administrators, systems analysts, database administrators, and Enterprise Resource Planning (ERP) managers responsible for performance tuning and capacity planning can learn about the behavioral characteristics of the major Solaris subsystems. The file system caching and memory management chapters provide a great deal of information about how Solaris behaves in real-world environments. The algorithms behind Solaris tunable parameters are covered in depth throughout the books.

Technical support staff responsible for the diagnosis, debugging, and support of Solaris will find a wealth of information about implementation details of Solaris. Major data structures and data flow diagrams are provided in each chapter to aid debugging and navigation of Solaris systems.

System users who just want to know more about how the Solaris kernel works will find high-level overviews at the start of each chapter.

Beyond the technical user community, those in academia studying operating systems will find that this text will work well as a reference. Solaris OS is a robust, feature-rich, volume production operating system, well suited to a variety of workloads, ranging from uniprocessor desktops to very large multiprocessor systems with large memory and input/output (I/O) configurations. The robustness and scalability of Solaris OS for commercial data processing, Web services, network applications, and scientific workloads is without peer in the industry. Much can be learned from studying such an operating system.

OpenSolaris

In June 2005, Sun Microsystems introduced OpenSolaris, a fully functional Solaris operating system release built from open source. As part of the OpenSolaris initiative, the Solaris kernel source was made generally available through an open license offering. This has some obvious benefits to this text. We can now include Solaris source directly in the text where appropriate, as well as refer to full source listings made available through the OpenSolaris initiative.

With OpenSolaris, a worldwide community of developers now has access to Solaris source code, and developers can contribute to whatever component of the operating system they find interesting. Source code accessibility allows us to structure the books such that we can cross-reference specific source files, right down to line numbers in the source tree.

OpenSolaris represents a significant milestone for technologists worldwide; a world-class, mature, robust, and feature-rich operating system is now easily accessible to anyone wishing to use Solaris, explore it, and contribute to its development.

Visit the Open Solaris Website to learn more about OpenSolaris:

```
http://www.opensolaris.org
```

The OpenSolaris source code is available at:

```
http://cvs.opensolaris.org/source
```

Source code references used throughout this text are relative to that starting location.

How the Books Are Organized

We organized the *Solaris™ Internals* volumes into several logical parts, each part grouping several chapters containing related information. Our goal was to provide a building block approach to the material by which later sections could build on information provided in earlier chapters. However, for readers familiar with particular aspects of operating systems design and implementation, the individual parts and chapters can stand on their own in terms of the subject matter they cover.

Volume 1: *Solaris™ Internals*

Part One: Introduction to Solaris Internals

Chapter 1 — Introduction

Part Two: The Process Model

Chapter 2 — The Solaris Process Model

Chapter 3 — Scheduling Classes and the Dispatcher

Chapter 4 — Interprocess Communication

Chapter 5 — Process Rights Management

Part Three: Resource Management

Chapter 6 — Zones

Chapter 7 — Projects, Tasks, and Resource Controls

Part Four: Memory

Chapter 8 — Introduction to Solaris Memory

Chapter 9 — Virtual Memory

Chapter 10 — Physical Memory

Chapter 11 — Kernel Memory

Chapter 12 — Hardware Address Translation

Chapter 13 — Working with Multiple Page Sizes in Solaris

Part Five: File Systems

Chapter 14 — File System Framework

Chapter 15 — The UFS File System

Part Six: Platform Specifics

Chapter 16 — Support for NUMA and CMT Hardware

Chapter 17 — Locking and Synchronization

Part Seven: Networking

Chapter 18 — The Solaris Network Stack

Part Eight: Kernel Services

Chapter 19 — Clocks and Timers

Volume 2: *Solaris™ Performance and Tools*

Updates and Related Material

To complement these books, we created a Web site at which we will place updated material, tools we refer to, and links to related material on the topics covered. We will regularly update the Web site (`http://www.solarisinternals.com`) with information about this text and future work on *Solaris™ Internals*. The Web site will be enhanced to provide a forum for Frequently Asked Questions (FAQs) related to the text, as well as general questions about Solaris internals, performance, and behavior. If bugs are discovered in the text, we will post errata on the Web site as well.

Notational Conventions

Table P.1 describes the typographic conventions used throughout these books, and Table P.2 shows the default system prompt for the utilities we describe.

Table P.1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	Command names, file names, and data structures.	The `vmstat` command. The `<sys/proc.h>` header file. The `proc` structure.
`AaBbCc123()`	Function names.	`page_create_va()`
`AaBbCc123(2)`	Manual pages.	Please see `vmstat(1M)`.

`AaBbCc123`	Commands you type within an example.	`$ vmstat`
		`r b w swap free re mf 0 0 0`
		`464440 18920 1 13`

AaBbCc123	New terms as they are introduced.	A *major page fault* occurs when...
MDB	The modular debuggers, including the user-mode debugger (mdb) and the kernel in-situ debugger (kmdb).	Examples that are applicable to both the user-mode and the in-situ kernel debugger.
mdb	The user-mode modular debugger.	Examples that are applicable the user-mode debugger.
kmdb	The in-situ debugger	Examples that are applicable to the in-situ kernel debugger.

Table P.2 Command Prompts

Shell	Prompt
Shell prompt	`minimum-osversion$`
Shell superuser prompt	`minimum-osversion#`
The mdb debugger prompt	>
The kmdb debugger prompt	[cpu]>

A Note from the Authors

Once again, a large investment in time and energy proved enormously rewarding for the authors. The support from Sun's Solaris kernel development group, the Solaris user community, and readers of the first edition has been extremely gratifying. We believe we have been able to achieve more with the second edition in terms of providing Solaris users with a valuable reference text. We certainly extended our knowledge in writing it, and we look forward to hearing from readers.