

存储技术原理分析

基于Linux 2.6内核源代码

敖青云 著

敖青云



存储技术原理分析

——基于Linux 2.6内核源代码

敖青云 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书通过对 Linux 2.6 内核源代码的分析, 详细讨论存储技术的内在实现原理。分为三条主线: 解释 PCI 设备、SCSI 设备及块设备的发现过程; 跟踪存储 I/O 路径, 即用户对文件的读/写请求怎么通过中间各个层次, 最终到达磁盘介质; 此外, 还简要介绍主机适配器、块设备驱动及文件系统等编程框架。

书中将设计一些主要的场景, 跟踪实现的各个层次, 对其中的主要函数进行代码级的讲解。在分析每个模块时, 会给出整体框架与主要数据结构之间的关系, 并列出了各个域的详细含义。

采用这种方式, 希望读者能对存储相关概念(如 RAID、快照等)的内在实现有具体的了解, 也试图帮助读者理解 Linux 内核设计和开发的一些思想, 为进一步分析其他模块(如进程管理、内存管理等)起借鉴作用。

本书适合作为高校计算机相关专业本科生和研究生学习操作系统的辅助和实践教材, 也适合作为 Linux 爱好者学习内核的参考书籍。同时, 它也是存储从业工程师深入理解存储架构, 以及软件开发工程师掌握软件架构的有效工具。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目(CIP)数据

存储技术原理分析: 基于 Linux 2.6 内核源代码 / 敖青云著. —北京: 电子工业出版社, 2011.9
ISBN 978-7-121-14432-5

I. ①存… II. ①敖… III. ①数据存贮 ②Linux 操作系统 IV. ①TP333 ②TP316.89

中国版本图书馆 CIP 数据核字(2011)第 171290 号

责任编辑: 许 艳

特约编辑: 顾慧芳

印 刷: 北京天宇星印刷厂

装 订: 三河市皇庄路通装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 860×1092 1/16 印张: 49.75 字数: 1450 千字

印 次: 2011 年 9 月第 1 次印刷

印 数: 3000 册 定价: 118.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

笔者寄语

我终于长长地嘘了一口气！诺大的空间显得格外安静，稀疏坐落着的人们还在埋头忙碌着。此刻的我，压抑住内心喜悦，轻轻地踱到窗前，透过厚厚的玻璃，看着雪花乱舞，或覆盖车顶，或湮没河面，或消失在那一片新搭起的葡萄架下的泥土里，自己的思绪竟随着跳动起来。

无从知晓，写这本书的念头一直充斥着我的脑海，多年来，片刻都没有改变。为了你，我不惜被放逐天际，只要求拥有支配自己时间的自由；为了你，虽不能够遁入山林，我也坚持做着这大都会的隐者——浮华非我所求，喧嚣与我无缘；为了你，是的，我什么都愿意。

人生有如登山。源于上帝赐予的机遇，以及个人的努力，我有幸能登上这么一个小小的山峰，途中的辛酸只能自己去体味，或者最多和身边的朋友倾诉，但一路过来的心得，以及登临绝顶后所看到的风景却实在不敢独享。既然喜欢文字作画，何不将它们记录下来，权当作同样有志于攀登的后来者的攻略，或者是无意于此的旁观者在为他事辛劳的间隙，也能够了解个中的美妙。

如果读者认为此书稍有所裨益，那么就烦请您到书店，或通过网络去购买此书。Linux 是一个有待去探索的深奥领域，我自信本书能为您提供一点线索，甚至作为随身指导，常常温故以知新。您的付出，无论相对于您的收获，还是相对于本书创作过程中的付出，都是值得的。当然，笔者也不避言，因为自己还是一个有志于其他山峰的攀登者，希望为下一次的旅程积累一些资本。

本书目标

大多数爱好者在阅读 Linux 内核源代码时会产生这样的困惑，我们很少能找到针对 Linux 操作系统，甚至某个单独的内核模块，在设计和开发方面的文档。仅有 Linux 社区的一些高手们对一些关键算法或者一些设计考虑的讨论。此外，当前大多数的源码分析书籍，都只是就函数或代码进行解释，没有给出整体和全面的视角。对于处于新手上路，或者小技初成级别的读者，只能获得局部和片面的认识，在理解这些讨论和阅读内核源码时会非常困难，常常产生挫败感。

写作本书的初衷，就是希望从软件设计和开发者的角度对与 Linux 存储相关模块作一个梳理。我们将设计一些主要的场景，跟踪实现的各个层次，对其中的主要函数进行代码级的讲解。在分析每个模块时，会给出整体框架与主要数据结构之间的关系，并列出来各个域的详细含义，比起单调的代码阅读理解，相信会达到更好的效果。

本书在构思之初，所计划的目标远非上面列举的几项。笔者曾经试图加入设备仿真原理部分，分析虚拟机是如何仿真 CPU、PCI 子系统、SCSI 子系统、存储适配器、网络适配器以及磁盘设备的。笔者还曾设想从 Linux 内核开发者的角度，尽量贴近软件设计和开发文档，讲解各个子系统和模块的构思过程、设计思想、技术难点及解决方案等。不过由于时间和精力所限，最后有所取舍。

本书最终确定命名为《存储技术原理分析——基于 Linux2.6 内核源代码》。坦率地说，这只是最初设定的写作目标中基础的部分，此外，还准备了网络子系统、iSCSI、NFS、Heartbeat 虚拟机等诸多素材，分别被规划为进阶篇、高级篇的内容。至于最终会不会有机会呈现给读者，那就要看本书的市场反应和笔者的精力了，本书旨在通过分析 Linux 内核源代码讲解存储、网络和虚拟机的相关技术。如果读者有足够的耐心和毅力跟随我们完成这一次旅程，您将理解以下几点：

- 设备发现过程：了解操作系统如何发现 PCI 设备、SCSI 设备、块设备，并和驱动绑定起来；
- 存储 I/O 路径：了解用户对文件的读/写请求怎么通过 I/O 路径，最终到达磁盘介质上；
- 内核编程模式：理解 PCI-SCSI HBA 驱动、块设备驱动，以及文件系统等编程框架。

本书读者

本书所针对的读者应该具有一定基础。应该对存储技术有些了解，应该对操作系统有些领悟，应该对软件开发有些心得，但是这种了解、领悟和心得又似乎浮于表面，经不住考验。最为重要的是，应该有对细节的渴望和对未知的好奇。本书通过分析 Linux 源代码系统阐述存储技术的内在原理，它要告诉的，不是 Linux 的历史有多长、功能有多强，或是那些古老的话题，重复在我们的身旁；它要讲述的，是一些执着的人们思考着什么，又最终决定了什么，在并不为我们所知的地方。

本书导读

本书针对的 Linux 内核版本号为 2.6.34，并且以 x86 为硬件平台为目标。为了突出重点，将在各个部分讨论最具普遍性、而不是最新的技术。例如，会探讨 PCI 和 SCSI，但不会涉及 PCIe 或者 SAS。至于文件系统部分，还是以 Minix 作为基础，尽管当前有很多非常强大的文件系统存在，并且还会不断被推出。在行文过程中，可能会谈到某些厂商的产品或驱动，这绝非出于市场考虑，也并不表明该产品是行业内功能最强或性能最优的，或者这些驱动是“没有错误 (bug free)”的。

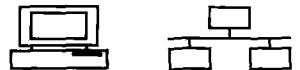
本书涉及的内容较多，分为三部分共 8 章。为了使读者在阅读过程中不至于迷失，请参考如下导读。

1. 第一部分为第 1 章，介绍存储技术的概况。



这一章讨论存储系统的组成元素：磁盘驱动器、存储设备、服务器部件和存储软件，阐述和存储相关的各种技术，如备份、快照、连续数据保护、重复数据删除、虚拟化及虚拟机技术等。

2. 第二部分包括第 2 章到第 4 章，介绍设备发现的过程。



第 2 章讲述 Linux 驱动模型。它为 Linux 内核构建了一个综合的、统一的机制，以内核对象为基础，将总线、设备和驱动关联起来，组织成一个层次结构的系统，方便了各种类型设备的热插拔和电源管理，同时借助 sysfs 提供了一个完全层次结构的用户视图。

第 3 章讲述 PCI 子系统。它将 SCSI 适配器、网络适配器等设备连接到主机 I/O 总线。

第 4 章讲述 SCSI 子系统。它将磁盘或者外部存储设备连接到主机 I/O 总线。

3. 第三部分包括第 5 章到第 8 章，介绍存储 I/O 的路径。



第 5 章讲述块 I/O 子系统。它向上层提供 I/O 请求 API，并实现 I/O 调度，将请求派发到具体块设备的请求

队列执行，以及提供请求完成的下半部处理 API，直至最终调用上层的请求完成回调函数结束 I/O。

第 6 章讲述 Linux RAID。MD 模块是一个虚拟块设备层。这一章将以 MD 模块源码为基础，阐述 Linux 内核中如何支持各种不同的 RAID 级别。

第 7 章讲述 Device Mapper。这一章将以 Device Mapper 模块源码为基础，阐述设备映射原理及各种映射规则的实现。

第 8 章讲述文件系统。文件系统是存储和组织文件（即一系列相关的数据），以便可以方便地进行查找和访问的一种机制。这一章将介绍 Linux 内核如何通过虚拟文件系统层，实现对各种具体文件系统的支持，以及从装载文件系统到访问文件等系统调用的流程。

本书图例

本书不会仅限于从概念上来讲解存储技术，而是以 Linux 内核源码为基础进行详细的阐述。理解 Linux 内核的关键在于把握各个数据结构之间的关系，用图示方式描画出各个数据结构之间的关系无疑是最为直观的。在描画各个关系图时，本书遵循如图所示的图例注解：

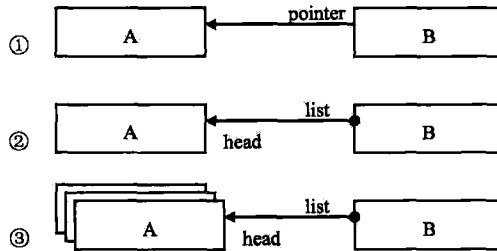


图 本书图例注解


图中①表示数据结构 B 有一个 pointer 域指向数据结构 A。它的示例代码如下：

```
struct B {  
    ...  
    struct A *pointer;  
    ...  
};
```

②表示数据结构 B 通过 list 域链入数据结构 A 的以 head 域为表头的链表，它的示例代码如下：

```
struct A {  
    ...  
    list_head head;  
    ...  
};  
struct B {  
    ...  
    list_head list;  
    ...  
};  
...  
void link_B_to_A(struct A *a, struct B *b) {  
    list_add(&b->list, &a->head);  
}
```

我们看到，在数据结构 A 中定义了 head 域，数据结构 B 中定义了 list 域。两个域都是 list_head 类型，这是 Linux 内核代码中表示链表的数据结构，在分析代码中遇到它再进行分析。上面的例子中还有一个函数 link_B_to_A，用于将结构 A 的对象通过 list 域，链接到结构 B 的对象以 head 域为首的链表中，其中 list 域被称为连接件，head 域被称为表头。

③表示数据结构 B 通过 list 域链入哈希表的某个哈希桶中。哈希桶的数据结构为 A，其 head 域为链表表头。表示结构 A 的数组。

感谢与联系方式

本书是一本技术性组织的书籍，能通过对 Linux 源码的分析来讲解存储、网络和虚拟机相关技术的内涵，外延及这些技术的原理。在写作过程中笔者查阅了很多资料，在此一并向有关资料的提供者表示感谢。

感谢我的导师上海交大计算机系的白英彩教授！白老师是我一生的导师，自毕业以来，始终给予我职业上的指导，并提供了许多宝贵的建议和机会。

感谢博文视点公司的几位编辑。如果不是有幸遇到他们，本书可能还只是孕育着的一个想法。在写作过程中，他们提供了很多的专业指导，尤为感动的是，在因延期交稿而忐忑时，还给予了从作者角度设身处地的劝慰，使得我安心、愉快地继续创作。

真诚地感谢我的家人，让我在紧张写作的间隙，总能够享受家的温馨。尤其是我的爱人邓玉洁，对于我的每一个想法，尽管可能不见得是主流，她总是给予极大的支持。由于她的努力，我可以无需过多顾虑生活上的压力，感谢她，还有我们生命中最重要的小末末！

最后，感谢我所有的朋友，以前的和现在的，有联系的和没有联系的。曾经为远离从前的朋友而遗憾，怕从此友情不再。现在终于释怀，人生就好像一个旅程，朋友就是我一路的风景，不同的阶段风景会不同，感谢他们，见证了我生命的历程。特别的感激送给：柴剑锋、陈颖、戴广成、丁旭华、董歆奕、方敏、谷大武、黄昉、蒋川群、蒋文蓉、金崇英、李虎、李静、李小勇、梁坚、刘俊、刘伟、任松林、田忠、王安保、王金浦、杨向群、张冠群。同样，向很多没有列出名字的朋友说声抱歉——你们在我心中。

当然，由于资料难觅，加之笔者水平有限，本书难免存在不少的瑕疵和错误的理解，敬请读者批评指正，可以发送邮件至 ao_qy@hotmail.com 进行联系。

最后附上小诗一首，结束全文。

【登南岳赋】

二零一零年，清明，霁，游寿岳衡山，经半山亭，南天门等，至祝融峰晋香，感自然浩瀚，叹世俗虚华，以此诗记之：

半山烟雨半山空，一路流云一路松，
携风漫道南门外，相邀长住寿天宫。

敖青云

2010年7月~2011年2月
写作于上海浦东新区图书馆
完稿于上海第二工业大学

目 录

第一部分 存储技术

第 1 章 存储技术概论	2
1.1 存储系统元素	2
1.1.1 磁盘驱动器	2
1.1.2 存储设备	4
1.1.3 服务器部件	7
1.1.4 存储软件	9
1.2 存储相关技术	10
1.2.1 备份技术	10
1.2.2 快照技术	13
1.2.3 连续数据保护技术	21
1.2.4 RAID 技术	22
1.2.5 “多路径”技术	36
1.2.6 虚拟化技术	39
1.3 网络存储结构	40
1.3.1 直接连接存储	40
1.3.2 网络连接存储	40
1.3.3 存储区域网络	41
1.4 存储 I/O 通道	41
1.4.1 存储 I/O 物理通道	42
1.4.2 存储 I/O 逻辑通道	43
1.4.3 虚拟机 I/O 逻辑通道	44
1.5 存储应用举例	45
1.5.1 同时提供文件服务和块服务	45
1.5.2 按需扩容、按需取用延缓企业投资	45
1.5.3 计算与存储分离便于故障恢复和系统升级	45
1.5.4 为高可用性集群提供共享存储	46
1.5.5 利用快照技术恢复被病毒破坏的数据	47
1.5.6 基于文件的数据备份和远程镜像方案	47
1.5.7 利用 PXE 和 iSCSI 实现远程引导和映像恢复	48

1.5.8 虚拟机故障的检测及迁移	49
-------------------------	----

第二部分 设 备

第 2 章 Linux 驱动模型	52
2.1 概述	52
2.2 引用计数	53
2.3 内核对象及集合	55
2.3.1 创建或初始化内核对象	58
2.3.2 将内核对象添加到 sysfs 文件系统	59
2.3.3 创建、初始化、添加内核对象集	63
2.3.4 发送内核对象变化事件到用户空间	63
2.4 sysfs 文件系统	69
2.4.1 构建内核对象、对象属性和对象关系的内部树	70
2.4.2 对 sysfs 文件的读/写转换为对属性的 show 和 store 操作	73
2.4.3 为具体内核对象定义属性的规范流程	77
2.5 kobject 编程模式	80
2.6 驱动模型对象	81
2.6.1 总线类型	82
2.6.2 设备	86
2.6.3 驱动	100
2.6.4 类	105
2.6.5 接口	107
2.7 驱动模型编程模式	108
第 3 章 PCI 子系统	110
3.1 概述	110
3.2 PCI 子系统对象	115
3.2.1 pci_bus: PCI 总线	116
3.2.2 pci_dev: PCI 设备	117
3.3 PCI 核心初始化	121
3.4 配置访问方法	124
3.4.1 机制#1 方式	126
3.4.2 PCIBIOS 方式	128
3.4.3 配置访问接口	133
3.5 PCI 总线扫描	133
3.5.1 PCI 总线编号范例	133
3.5.2 PCI 总线扫描流程	137
3.6 PCI 中断路由	160

3.6.1	中断路由初始化	165
3.6.2	查找中断路由表	166
3.6.3	查找中断路由驱动	167
3.6.4	分配 ISA IRQ 号	171
3.7	PCI 资源分配	177
3.7.1	PCI 资源分配范例	178
3.7.2	PCI 资源分配流程	181
3.8	PCI 设备驱动编程模式	193
3.8.1	定义 PCI 驱动结构	194
3.8.2	定义支持设备 ID 列表	194
3.8.3	实现 probe 回调方法	196
3.8.4	实现 remove 回调方法	198
3.8.5	实现其他回调方法	199
3.8.6	注册与注销 PCI 驱动	199
第 4 章	SCSI 子系统	201
4.1	概述	201
4.2	SCSI 子系统对象	202
4.2.1	scsi_host_template: SCSI 主机适配器模板	203
4.2.2	Scsi_Host: SCSI 主机适配器	207
4.2.3	scsi_target: SCSI 目标节点	210
4.2.4	scsi_device: SCSI 逻辑设备	211
4.2.5	scsi_cmnd: SCSI 命令	215
4.3	SCSI 子系统初始化	216
4.4	添加适配器到系统	216
4.5	SCSI 设备探测	222
4.5.1	探测流程入口	224
4.5.2	探测逻辑单元	232
4.5.3	添加 SCSI 设备	237
4.6	SCSI 磁盘驱动	241
4.6.1	同步执行部分	244
4.6.2	异步执行部分	247
4.6.3	重新校验磁盘	249
4.6.4	让磁盘转起来	251
4.7	SCSI 命令执行	254
4.8	SCSI 错误恢复	259
4.8.1	命令进入错误恢复	261
4.8.2	错误恢复线程执行	262
4.8.3	发送错误恢复命令	275

4.9	SCSI 低层驱动编程模式	279
4.9.1	定义主机适配器模板	279
4.9.2	完善探测回调处理逻辑	279
4.9.3	实现 <code>queucommand</code> 回调函数	279
4.9.4	实现中断处理函数	283
4.9.5	实现其他回调函数	283
4.9.6	模块加载和卸载	283

第三部分 存储 I/O

第 5 章	块 I/O 子系统	286
5.1	概述	286
5.2	块 I/O 子系统对象	287
5.2.1	<code>gendisk</code> : 通用磁盘	289
5.2.2	<code>hd_struct</code> : 分区	291
5.2.3	<code>block_device</code> : 块设备	292
5.2.4	<code>request_queue</code> : 请求队列	293
5.2.5	<code>request</code> : 块设备驱动层请求	296
5.2.6	<code>bio</code> : 通用块层请求	298
5.3	添加磁盘到系统	300
5.3.1	分配通用磁盘描述符	300
5.3.2	添加到 <code>sysfs</code> 文件系统	302
5.3.3	获取磁盘块设备描述符	305
5.3.4	打开磁盘块设备描述符	306
5.3.5	重新扫描磁盘分区	310
5.3.6	设备号映射机制	314
5.4	请求处理过程	315
5.4.1	上层向块 I/O 子系统提交请求	315
5.4.2	构造、排序或合并请求	320
5.4.3	SCSI 策略例程逐个处理请求	327
5.4.4	为请求构造 SCSI 命令	334
5.4.5	为 SCSI 命令准备聚散列表	343
5.4.6	派发 SCSI 命令到低层驱动	349
5.5	I/O 调度算法	352
5.5.1	为请求队列建立关联的 I/O 调度队列	356
5.5.2	判断 <code>bio</code> 是否可以被合并到 <code>request</code>	356
5.5.3	将请求添加到 I/O 调度队列或请求队列	359
5.5.4	从 I/O 调度队列派发请求到请求队列	362
5.6	请求处理完成	366

5.6.1	低层驱动调用完成回调函数	366
5.6.2	引发块 I/O 子系统的软中断	368
5.6.3	调用请求队列的软中断回调	369
5.6.4	调用上层的完成回调函数	383
5.7	屏障 I/O 处理	386
5.7.1	屏障 I/O 接口	386
5.7.2	添加屏障请求	388
5.7.3	处理屏障请求	389
5.7.4	完成屏障请求	393
5.8	完整性保护	396
5.8.1	数据完整性对象	397
5.8.2	为块设备注册完整性能力	400
5.8.3	为 bio 准备完整性元数据	402
5.8.4	校验完整性元数据	406
5.8.5	修正 bio 基准标签	408
5.9	磁盘类设备驱动编程模式	411
5.9.1	定义磁盘类设备私有数据结构	411
5.9.2	定义和实现块设备操作表	411
5.9.3	分配和初始化磁盘类设备相关结构	411
5.9.4	为磁盘类设备准备请求队列并添加通用磁盘到系统	412
第 6 章	Multi-Disk (MD) 模块	413
6.1	概述	413
6.2	RAID 模块对象	414
6.2.1	mddev_t: RAID 设备	414
6.2.2	mdk_rdev_t: 成员磁盘	418
6.2.3	mdk_personality: MD 个性	419
6.3	MD 模块初始化	420
6.4	MD 设备创建	423
6.4.1	从用户空间打开 MD 设备	424
6.4.2	用户空间发送 ioctl 创建 MD	428
6.4.3	自动检测和运行 RAID	439
6.5	MD 设备请求执行	439
6.6	MD 个性化编程模式	440
6.6.1	定义私有数据结构	441
6.6.2	声明个性化结构	442
6.6.3	实现个性化方法	442
6.6.4	实现模块加载和卸载方法	445
6.7	RAID0 模块	445

6.7.1	为 RAID0 设备构造条带区域.....	446
6.7.2	查找包含给定偏移的条带区域.....	451
6.7.3	映射到成员设备及其扇区偏移.....	451
6.8	RAID5 模块.....	452
6.8.1	RAID5 模块对象.....	452
6.8.2	请求执行过程.....	459
6.8.3	同步和恢复过程.....	507
第 7 章	Device Mapper 模块.....	509
7.1	概述.....	509
7.2	Device Mapper 对象.....	510
7.2.1	dm_table: 映射表结构.....	512
7.2.2	dm_target: 映射目标结构.....	513
7.2.3	mapped_device: 映射设备结构.....	514
7.2.4	dm_dev: 低层设备结构.....	515
7.2.5	target_type: 映射目标类型.....	516
7.3	Device Mapper 模块初始化.....	518
7.4	映射设备的创建.....	519
7.4.1	分配映射设备描述符.....	521
7.4.2	加载映射表.....	526
7.4.3	恢复映射设备.....	532
7.5	映射设备的请求执行.....	536
7.5.1	添加到延迟链表.....	537
7.5.2	分割与处理 bio.....	539
7.6	内核复制线程.....	549
7.6.1	准备复制任务.....	551
7.6.2	任务处理流程.....	553
7.7	Device Mapper 目标类型编程模式.....	556
7.7.1	定义私有数据结构.....	557
7.7.2	声明目标类型结构.....	557
7.7.3	实现目标类型方法.....	557
7.7.4	实现模块加载和卸载方法.....	558
7.8	条带映射模块.....	558
7.8.1	构造函数.....	559
7.8.2	析构函数.....	562
7.8.3	映射函数.....	562
7.8.4	end_io 函数.....	563
7.9	快照映射模块.....	563
7.9.1	快照映射对象.....	564

7.9.2	快照源构造	569
7.9.3	快照构造	570
7.9.4	快照源读/写	577
7.9.5	快照读/写	585
7.9.6	例外仓库	588
第 8 章	文件系统	593
8.1	概述	593
8.2	文件系统对象	595
8.2.1	file_system_type: 文件系统类型	596
8.2.2	super_block: VFS 超级块	597
8.2.3	inode: VFS 索引节点	602
8.2.4	dentry: VFS 目录项	610
8.2.5	vfsmount: 文件系统装载	612
8.3	装载文件系统	614
8.3.1	mount 系统调用的处理流程	618
8.3.2	构建子文件系统装载实例	621
8.3.3	关联文件系统的超级块实例	623
8.3.4	调用回调函数填充超级块	626
8.3.5	装载到全局文件系统树	630
8.4	路径查找	632
8.4.1	路径查找入口	635
8.4.2	逐个分量解析	637
8.4.3	解析单个分量	642
8.4.4	上溯通过装载点	645
8.4.5	下溯通过装载点	646
8.4.6	处理符号链接	646
8.5	打开文件	651
8.5.1	open 系统调用的处理流程	653
8.5.2	解析路径最后一个分量	658
8.5.3	填充文件描述符的内容	662
8.6	读文件	665
8.6.1	read 系统调用的处理流程	670
8.6.2	基于缓冲页面构造 I/O 请求	683
8.6.3	直接针对页面构造 I/O 请求	690
8.6.4	从文件块编号推导磁盘块编号	696
8.7	写文件	700
8.7.1	write 系统调用的处理流程	700
8.7.2	通知为缓冲写请求作准备	707

8.7.3	通知数据已复制到缓冲区	712
8.8	冲刷文件	715
8.8.1	BDI 相关对象	715
8.8.2	注册后备设备信息	719
8.8.3	forker 线程执行流程	721
8.8.4	flusher 线程执行流程	723
8.8.5	同步相关系统调用	749
8.9	块设备文件	761
8.9.1	块设备的主 inode 和次 inode	762
8.9.2	对块设备文件的操作转换为对块设备的操作	764
8.9.3	对块设备文件的读/写作用于块设备之上	767
8.10	文件系统编程模式	767
主要参考文献		769

图 索 引

图 1-1	磁盘驱动器的物理组件	3
图 1-2	SSD 驱动器的框图	4
图 1-3	SAS 磁盘柜的前视图和后视图	5
图 1-4	双控制器 SAS 磁盘柜内部框图	5
图 1-5	NAS 网络结构	6
图 1-6	iSCSI 层次	6
图 1-7	NAS/iSCSI 集成存储系统模块示意图	7
图 1-8	SAS 主机适配器框图	8
图 1-9	适配器硬件、固件及驱动之间的关系	8
图 1-10	网络适配器、TOE 网络适配器和 iSCSI 适配器	9
图 1-11	直接连接备份模型	11
图 1-12	网络连接备份模型	11
图 1-13	脱局域网备份模型	12
图 1-14	脱服务器备份模型	12
图 1-15	备份过程中可能导致的数据不一致问题	13
图 1-16	创建快照	14
图 1-17	首次写源卷	15
图 1-18	后续写源卷	15
图 1-19	读源卷	15
图 1-20	读快照卷——未命中	16
图 1-21	读快照卷——命中	16
图 1-22	快照回滚	17
图 1-23	回滚和前滚	17
图 1-24	写时转向	18
图 1-25	拆分镜像备份过程	19
图 1-26	做快照时刷新应用程序缓存	20
图 1-27	连续数据保护范例	21
图 1-28	条带化模型	23
图 1-29	RAID0 布局	24

图 1-30	RAID1 布局	24
图 1-31	RAID1 在正常和降级状态下的读/写操作	25
图 1-32	RAID2 布局	25
图 1-33	RAID3 布局	26
图 1-34	RAID3 在正常和降级状态下的读/写操作	26
图 1-35	RAID4 布局	27
图 1-36	RAID5 布局	27
图 1-37	RAID5 校验分布算法	28
图 1-38	RAID5 在正常和降级状态下的读操作	28
图 1-39	RAID5 在正常状态下的写操作	29
图 1-40	RAID5 在降级状态下的写操作	29
图 1-41	RAID6 的 P+Q 方案	30
图 1-42	RAID6 的 EVEN-ODD 方案中水平校验计算	31
图 1-43	RAID6 的 EVEN-ODD 方案中对角校验计算	32
图 1-44	利用 RAID6 的 EVEN-ODD 算法恢复两个数据磁盘故障	33
图 1-45	RAID6 恢复实例——正常布局	34
图 1-46	RAID6 恢复实例——磁盘故障	34
图 1-47	RAID6 恢复实例——第一轮	34
图 1-48	RAID6 恢复实例——第二轮	34
图 1-49	RAID6 恢复实例——第三轮	35
图 1-50	RAID6 恢复实例——第四轮	35
图 1-51	RAID6 恢复实例——第五轮	35
图 1-52	RAID6 恢复实例——回到正常状态	35
图 1-53	RAID6 的 DH1 方案	36
图 1-54	RAID6 的 DH2 方案	36
图 1-55	“多路径”硬件配置（从左到右依次为①、②、③、④）	37
图 1-56	“多路径”软件的实现层次	38
图 1-57	存储虚拟化分类	39
图 1-58	存储技术分类	40
图 1-59	Intel 北桥/南桥架构	41
图 1-60	应用服务器的存储 I/O 物理通道	42
图 1-61	网络存储子系统的存储 I/O 物理通道	43
图 1-62	存储 I/O 逻辑通道	44
图 1-63	虚拟机存储栈	45
图 1-64	存储设备同时提供文件服务和块服务	45
图 1-65	即插即用的在线存储扩容	46