

HZ BOOKS
华章科技



FreeBSD Device Drivers: A Guide for the Intrepid

深入理解 FreeBSD设备驱动程序开发

(加) Joseph Kong 著

陈毅东 译



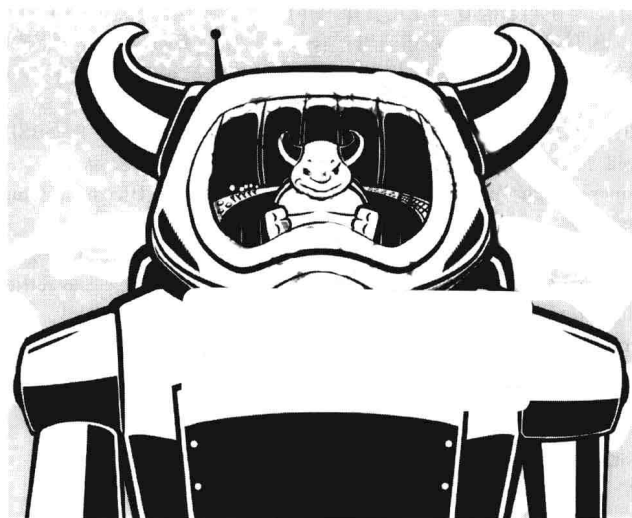
机械工业出版社
China Machine Press

FreeBSD Device Drivers: A Guide for the Intrepid

深入理解 FreeBSD设备驱动 程序开发

(加) Joseph Kong 著

陈毅东 译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

深入理解 FreeBSD 设备驱动程序开发 / (加) 约瑟夫 (Joseph, K.) 著; 陈毅东译. —北京: 机械工业出版社, 2013.2

(华章程序员书库)

书名原文: FreeBSD Device Drivers: A Guide for the Intrepid

ISBN 978-7-111-41157-4

I. 深… II. ① 约… ② 陈… III. 设备驱动程序—程序设计 IV. TP31

中国版本图书馆 CIP 数据核字 (2013) 第 008997 号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2012-5214

本书是 FreeBSD 设备驱动开发领域的经典著作, 由具有 20 余年从业经验的资深 FreeBSD 设备驱动编程专家撰写, FreeBSD 内核开发专家鼎力推荐! 本书从宏观上讲解了 FreeBSD 设备驱动程序的框架结构, 从微观上讲解了 FreeBSD 与驱动程序相关的各个接口和模块的作用及工作原理, 而且还从源码实现的角度分析了设备驱动程序的内核机制。本书面向的读者是驱动开发程序员, 更注重实践而非理论, 通过真实的驱动开发案例为编写健壮高效的驱动程序提供优秀指导。

本书共 17 章: 第 1 章介绍了设备驱动程序的类型、可加载内核模块的编译与加载; 第 2 章介绍了内核的分配与管理; 第 3 章讲解了在用户空间下如何控制设备驱动程序并与之通信; 第 4 章讨论了与多线程编程和并发执行有关的问题和解决方案; 第 5 章解析了延迟代码执行和异步代码执行等机制并解释了这些机制的深层原因; 第 6 章介绍了虚拟空调制解调器的实际应用; 第 7 章讲解了 FreeBSD 用作管理系统中硬件设备的底层基本结构; 第 8 章讨论 FreeBSD 的中断处理; 第 9 章全面讲解了并口打印机的驱动程序; 第 10 章讲解了端口映射和内存映射的输入输出; 第 11 章分析了智能平台管理接口驱动程序; 第 12 章讲解了如何在 FreeBSD 中使用直接内存访问机制; 第 13 章介绍了如何管理存储设备, 其中包括磁盘驱动器、闪存等; 第 14 章概述通用存取方法; 第 15 章讲解了如何管理 USB 设备; 第 16 章描述了网络驱动程序所使用的数据结构, 以及对消息驱动中断机制的介绍; 第 17 章详细介绍了 em(4) 的包接收与传输组件。

Copyright©2012 by Joseph Kóng. Title of English-language original: FreeBSD Device Drivers: A Guide for the Intrepid, ISBN 978-1-59327-204-3, published by No Starch Press.

Simplified Chinese-language edition copyright©2013 by Beijing Huazhang Graphics & Information Co., China Machine Press.

No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or any information storage and retrieval system, without permission, in writing, from the publisher.

All rights reserved.

本书中文简体字版由 No Starch Press 授权机械工业出版社在全球独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

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: 秦 健

北京市荣盛彩色印刷有限公司印刷

2013 年 2 月第 1 版第 1 次印刷

186mm×240mm·20.75 印张

标准书号: ISBN 978-7-111-41157-4

定 价: 69.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzsj@hzbook.com

译者序

驱动程序是各操作系统内核相关编程中最复杂的部分之一，牵涉面甚广，因而也是较难掌握的内容之一。与此同时，作为控制外部设备并与外部设备通信的关键模块，驱动程序又是系统中的重要组成部分。尽管重要，但相关方面的专业书籍却不多，本书则是其中的佼佼者。本书的作者 Joseph Kong 在 FreeBSD 系统底层方面有着极深厚的功底和丰富的经验，他通过对大量实例的剖析，深入浅出地向读者介绍了 FreeBSD 驱动程序开发的方方面面，为读者提供了一本不可多得优秀入门指南。本书内容丰富、涵盖面宽，涉及了 FreeBSD 驱动程序编程从基础知识到高级概念的几乎所有内容并配有若干个完整实例的讲解。而且，本书对代码的讲解非常细致，书中每个实例的关键点都很仔细地在代码中标记并详细讲解，这种方式使读者更容易理解代码，进而也能更好地理解书籍的内容。

译者本人主要从事自然语言处理与机器翻译的相关研究，同时也对操作系统的底层十分感兴趣。承接本书的翻译有两个目的：一方面是希望把这本优秀的书籍推荐给大家；另一方面是希望通过本书的翻译过程切身体验人工译者的需求和汉英语言的差异。钱锺书先生在《林纾的翻译》一文中通过对“译”字本义的探究，总结出翻译所向往的最高境界是“化”，难以避免的毛病是“讹”，能起的作用是“诱”[⊖]。对照本书的翻译，尽管译者在翻译过程中一直在为保持原书的原汁原味而努力，但距离达到“化”的境界却还相距很远，反而是受译者本人的水平所限，译书中不可避免地还存在“讹”的毛病，也希望读者能海涵一二并不吝赐教（译者声明：译书中的疏漏都是翻译所造成的，与原著无关）。同时，译者也希望本书能起到“诱”的作用，让读者在看了译书后能有兴趣看原著，特别希望读者在入门后拿原著作为今后驱动程序开发过程中的参考手册。

非常感谢机械工业出版社给我翻译本书的机会；感谢吴怡编辑全程的悉心指导；感谢我的家人在我整个翻译过程中所给予的大力支持。没有你们的支持，我将很难完成此书的翻译。

⊖ 参见钱锺书所著《七缀集》。

序 言

操作系统的多数模块都是由专攻相应系统的人开发和维护的，设备驱动程序却很特别，其开发人员的情况要复杂得多：一些设备驱动程序的作者对某个特定的操作系统很有经验，而另一些则熟知特殊的硬件组件并需跨平台为该组件开发设备驱动程序。而且，设备驱动程序通常是相对独立的，开发人员可以在维护一个设备驱动程序时将系统的其他部分看成黑箱子。

当然，所谓黑箱子还是有接口的，每个操作系统都为设备驱动程序提供了各自的接口集。任何系统上的设备驱动程序都要执行很多通用的操作，例如：发现设备、为连接设备分配资源、管理异步事件等。但每个操作系统在处理这些任务时都有各自的方式，并且提供的高层接口也有所不同。因此，要编写一个健壮且高效的设备驱动程序，关键是理解好特定操作系统所提供接口的特殊细节。

本书是 FreeBSD 上最通用的设备驱动程序接口的优秀指南。本书的内容同时涵盖低层接口和高层接口：低层接口包括如何连接可选设备以及如何管理设备资源等；高层接口则包括连接网络以及存储栈等。此外，本书还介绍了一些内核环境下可用的 API，例如：内存分配、定时器以及同步原语等。这些内容对从事 FreeBSD 内核相关工作的人员也是很有用的。本书对于 FreeBSD 设备驱动程序开发人员而言将是一份可喜的参考资料。

John Baldwin

FreeBSD 内核资深开发专家

前言

欢迎阅读本书！本书旨在对你理解 FreeBSD 设备驱动程序提供帮助。当你看完本书，你将能建立、配置并管理自己的 FreeBSD 设备驱动程序。

本书内容虽然主要覆盖 FreeBSD 第 8 版，这也是本书成书时的主流版本，但是，其中多数内容都能适用于早期的版本，也能适用于未来的版本。

本书的读者对象

我是以程序员的身份写的这本书，因而这本书也主要是写给程序员看的。因此，本书将重点关注程序开发而不是理论，并且本书将介绍真实的设备驱动程序（即那些能控制硬件的程序）。如果说有一个人在写成一本书前从来没有看过任何一本书，这是难以想象的！对于编写设备驱动程序而言也同样如此。

所需的前期基础

为了更好地理解本书内容，读者应该熟悉 C 程序设计语言，同时还应该了解操作系统设计的基本知识，如进程和线程之间的区别等。

如果你缺乏上述必要的背景知识，我建议先阅读以下几本书，或者至少把它们放在手边以备参考：

- 《The C Programming Language》，Brian W. Kernighan 和 Dennis M. Ritchie 著，Prentice Hall PTR, 1988。[⊖]
- 《Expert C Programming》，Peter van der Linden 著，Prentice Hall PTR, 1994。
- 《The Design and Implementation of the FreeBSD Operating System》，Marshall Kirk

⊖ 本书中文版《C 程序设计语言》（978-7-111-12806-0）及影印版（978-7-111-19626-0）由机械工业出版社引进出版。

McKusick 和 George V. Neville-Neil 著, Addison-Wesley Professional, 2005。

全书内容概览

本书共包含 17 章。

第 1 章 构建并运行模块

该章综述设备驱动程序编程的基本概念和术语等。

第 2 章 分配内存

该章描述 FreeBSD 的内核内存管理例程。

第 3 章 设备通信与控制

该章介绍在用户空间下如何控制设备驱动程序并与之通信。

第 4 章 线程同步

该章讨论与多线程编程和并发执行有关的问题和解决方案。

第 5 章 延迟执行

该章描述延迟代码执行和异步代码执行等机制并解释为什么需要这些机制。

第 6 章 案例研究：虚拟空调制解调器

该章介绍本书第一个实际可用的设备驱动程序。

第 7 章 Newbus 和资源分配

该章介绍 FreeBSD 用于管理系统中硬件设备的底层基本结构。该章以后的内容中将完全以真实硬件对象进行讲解。

第 8 章 中断处理

该章讨论 FreeBSD 中的中断处理。

第 9 章 案例研究：并口打印机驱动程序

该章全面介绍 lpt(4)，即并口打印机驱动程序。

第 10 章 管理与使用资源

该章内容涉及端口映射输入输出（port-mapped I/O）和内存映射输入输出（memory-mapped I/O）。

第 11 章 案例研究：智能平台管理接口驱动

该章分析 ipmi(4)，即智能平台管理接口（intelligent platform management interface）驱动，该驱动使用第 10 章的端口映射输入输出和内存映射输入输出。

第 12 章 直接内存访问

该章讲解如何在 FreeBSD 中使用直接内存访问（Direct Memory Access, DMA）机制。

第 13 章 存储驱动

该章介绍如何管理存储设备，其中包括磁盘驱动器、闪存等。

第 14 章 通用存取方法

该章概要介绍通用存取方法（Common Access Method, CAM），借助 CAM 你将可以管理主机总线适配器（HBA）。

第 15 章 USB 驱动

该章介绍如何管理 USB 设备。其中也包括对 `ulpt(4)` 即 USB 打印驱动的全面介绍。

第 16 章 网络驱动程序 I：数据结构

该章描述网络驱动程序所使用的数据结构，其中也包括对消息驱动中断（Message Signaled Interrupts, MSI）机制的介绍。

第 17 章 网络驱动程序 II：包的接收与传输

该章详细介绍 `em(4)` 的包接收与传输组件，`em(4)` 是 Intel PCI 千兆以太网卡（`intel PCI gigabit ethernet adapter`）的驱动程序。

欢迎上路

希望本书能让你感觉到有用而且有趣。对于本书的意见或勘误修复随时欢迎发送到 joe@thestackframe.org。

好了，介绍得够多了，让我们开始吧。

致 谢

没有一本书是孤立无援的，如果没有下列人员的帮助和支持，你无法看到这本书，因此我要在此隆重致谢。

首先，感谢 Bill Pollock 和 No Starch 出版社的诸位编辑给我写这本书的机会以及一路以来的帮助。特别感谢 Alison Law、Riley Hoffman 和 Tyler Ortman 将所有条件综合起来。Alison，我要感谢你至少两次：感谢你多次找到我的输入错误；感谢你不断鞭策我跟上进度。还要感谢 Damon Larson 和 Megan Dunchak 两位技术编辑以及 Jessica Miller。

没有 John Baldwin 出色的技术评阅，我也无法完成本书。他耐心地回答了我所有的（愚蠢的）问题并帮我改善了代码。感谢我的哥哥 Justin Kong 多次评阅此书，你肯定当得起“铁人”奖。感谢 Aharon Robbins 的评阅，也感谢我的朋友 Elizabeth C. Mitchell 为我画图表（以及为我烤核桃仁巧克力饼）。同时也感谢 George Neville-Neil 和 Michael W. Lucas 所提的建议。

感谢我的父亲提供给我硬件方面的专业知识以及让我可用来写作本书的设备。感谢我的母亲给我的爱与支持。我知道您每天都在为我祈祷。感谢所有给过我帮助的朋友。

最后也是最重要的，感谢开源软件和 FreeBSD 社区慷慨共享。没有他们，我只能是一个差劲的程序员，没有什么东西可写。

目 录

译者序 序言 前言 致谢

第 1 章 构建并运行模块	1
1.1 设备驱动程序的类型	1
1.2 可加载内核模块	2
1.2.1 模块事件处理程序	2
1.2.2 DECLARE_MODULE 宏	3
1.3 Hello, world!	5
1.4 编译与加载	6
1.5 字符设备驱动程序	7
1.5.1 d_foo 函数	8
1.5.2 字符设备开关表	8
1.5.3 make_dev 函数和 destroy_dev 函数	9
1.6 实例：内存字符设备驱动程序	10
1.6.1 echo_write 函数	13
1.6.2 echo_read 函数	14
1.6.3 echo_modevent 函数	14
1.6.4 DEV_MODULE 宏	15

1.6.5 试运行	16
1.7 块设备驱动程序消失了	16
1.8 本章小结	17
第 2 章 分配内存	18
2.1 内存管理例程	18
2.2 malloc_type 结构	19
2.2.1 MALLOC_DEFINE 宏	20
2.2.2 MALLOC_DECLARE 宏	20
2.3 让一切配合起来	21
2.4 连续物理内存管理例程	23
2.5 一个简单的例子	24
2.6 本章小结	26
第 3 章 设备通信与控制	27
3.1 ioctl 接口	27
3.2 定义 ioctl 命令	29
3.3 实现 ioctl	30
3.3.1 echo_write 函数	34
3.3.2 echo_set_buffer_size 函数	35
3.3.3 echo_ioctl 函数	36
3.3.4 echo_modevent 函数	37

3.3.5 试运行	38	4.6.1 race_modevent 函数	73
3.4 调用 ioctl	38	4.6.2 试运行	74
3.5 sysctl	41	4.7 共享 / 独占锁	75
3.6 实现 sysctl (第 1 部分)	41	4.8 共享 / 独占锁管理例程	75
3.7 sysctl 上下文管理例程	45	4.9 实现共享 / 独占锁	77
3.8 创建动态的 sysctl	45	4.10 读者 / 写者锁	81
3.8.1 SYSCTL_STATIC_		4.11 读者 / 写者锁管理例程	81
CHILDREN 宏	48	4.12 条件变量	83
3.8.2 SYSCTL_CHILD-		4.13 条件变量管理例程	83
REN 宏	48	4.14 一般性原则	85
3.9 实现 sysctl (第 2 部分)	48	4.14.1 避免在独占锁上	
3.9.1 sysctl_set_buffer_		递归	85
size 函数	52	4.14.2 避免长时间占用独	
3.9.2 试运行	53	占锁	85
3.10 本章小结	53	4.15 本章小结	85
第 4 章 线程同步	54	第 5 章 延迟执行	86
4.1 一个简单的同步问题	54	5.1 主动上下文切换 / 休眠	86
4.2 一个更复杂的同步问题	55	5.2 实现休眠与条件变量	88
4.2.1 race_new 函数	59	5.2.1 sleep_modevent 函数	92
4.2.2 race_find 函数	59	5.2.2 load 函数	92
4.2.3 race_destroy 函数	60	5.2.3 sleep_thread 函数	93
4.2.4 race_ioctl 函数	60	5.2.4 sysctl_debug_sleep_	
4.2.5 race_modevent 函数	61	test 函数	94
4.2.6 问题的根源	62	5.2.5 unload 函数	95
4.3 防止竞态条件	66	5.2.6 试运行	96
4.4 互斥锁	67	5.3 内核事件处理器	96
4.4.1 自旋互斥锁	67	5.4 callout	98
4.4.2 休眠互斥锁	67	5.5 callout 和竞态条件	100
4.5 互斥锁管理例程	68	5.6 任务队列	101
4.6 实现互斥锁	70	5.6.1 全局任务队列	101

5.6.2 任务队列管理例程	102	7.2.1 foo_pci_probe 函数	125
5.7 本章小结	103	7.2.2 foo_pci_attach 函数	126
第 6 章 案例研究：虚拟空调制解		7.2.3 d_foo 函数	127
调器	104	7.2.4 foo_pci_detach 函数	127
6.1 前期基础	104	7.2.5 试运行	128
6.2 代码分析	105	7.3 硬件资源管理	128
6.2.1 nmdm_modevent 函数	107	7.4 本章小结	130
6.2.2 nmdm_clone 函数	108	第 8 章 中断处理	131
6.2.3 nmdm_alloc 函数	110	8.1 注册中断处理程序	131
6.2.4 nmdm_outwakeup 函数	111	8.2 FreeBSD 中的中断处理 程序	132
6.2.5 nmdm_task_tty 函数	112	8.3 实现中断处理程序	133
6.2.6 nmdm_inwakeup 函数	113	8.3.1 pint_identify 函数	138
6.2.7 nmdm_modem 函数	114	8.3.2 pint_probe 函数	138
6.2.8 nmdm_param 函数	115	8.3.3 pint_attach 函数	139
6.2.9 nmdm_timeout 函数	116	8.3.4 pint_detach 函数	140
6.2.10 bits_per_char 函数	117	8.3.5 pint_open 函数	140
6.2.11 试运行	118	8.3.6 pint_close 函数	142
6.3 本章小结	118	8.3.7 pint_write 函数	142
第 7 章 Newbus 和资源分配	119	8.3.8 pint_read 函数	143
7.1 自动配置与 Newbus 驱动 程序	119	8.3.9 pint_intr 函数	144
7.1.1 device_foo 函数	119	8.3.10 试运行	144
7.1.2 设备方法表	121	8.4 生成并口中断	145
7.1.3 DRIVER_MO- DULE 宏	121	8.5 本章小结	146
7.2 驱动程序示例	122	第 9 章 案例研究：并口打印机驱	
		动程序	147
		9.1 代码分析	147
		9.1.1 lpt_identify 函数	152
		9.1.2 lpt_probe 函数	152

9.1.3	lpt_detect 函数	152	10.3	让一切配合起来	179
9.1.4	lpt_port_test 函数	153	10.3.1	led_identify 函数	183
9.1.5	lpt_attach 函数	154	10.3.2	led_probe 函数	184
9.1.6	lpt_detach 函数	156	10.3.3	led_attach 函数	184
9.1.7	lpt_open 函数	157	10.3.4	led_detach 函数	185
9.1.8	lpt_read 函数	159	10.3.5	led_open 函数	186
9.1.9	lpt_write 函数	161	10.3.6	led_close 函数	186
9.1.10	lpt_intr 函数	163	10.3.7	led_read 函数	187
9.1.11	lpt_timeout 函数	164	10.3.8	led_write 函数	188
9.1.12	lpt_push_bytes 函数	165	10.4	本章小结	188
9.1.13	lpt_close 函数	166	第 11 章 案例研究：智能平台管理		
9.1.14	lpt_ioctl 函数	167	接口驱动	189	
9.1.15	lpt_request_ppbus 函数	169	11.1 代码分析	189	
9.1.16	lpt_release_ppbus 函数	169	11.1.1 ipmi_pci_probe 函数	191	
9.2	本章小结	170	11.1.2 ipmi_pci_match 函数	192	
第 10 章 管理与使用资源	171		11.1.3 ipmi_pci_attach 函数	192	
10.1 输入输出端口与输入 输出内存	171		11.1.4 ipmi2_pci_probe 函数	195	
10.1.1 从输入输出端口和 输入输出内存读取 数据	171		11.1.5 ipmi2_pci_attach 函数	195	
10.1.2 向输入输出端口和 输入输出内存写入 数据	173		11.2 本章小结	197	
10.1.3 流操作	175		第 12 章 直接内存访问	198	
10.2 内存栅栏	178		12.1 实现 DMA	198	
			12.1.1 初始化 DMA 数据 传输	200	
			12.1.2 取消 DMA	201	
			12.2 创建 DMA 标签	201	

12.3	销毁 DMA 标签	203	13.1.5	驱动程序私有数据 字段	213
12.4	DMA 映射管理例程 (第 1 部分)	204	13.2	disk 结构管理例程	214
12.5	将 (DMA) 缓冲区载入 DMA 映射	204	13.3	块输入输出结构	214
12.5.1	bus_dma_segment 结构	205	13.4	块输入输出队列	216
12.5.2	bus_dmamap_load 函数	205	13.5	让一切配合起来	217
12.5.3	bus_dmamap_load_ mbuf 函数	206	13.5.1	at45d_attach 函数	222
12.5.4	bus_dmamap_load_ mbuf_sg 函数	207	13.5.2	at45d_delayed_ attach 函数	223
12.5.5	bus_dmamap_load_ uio 函数	207	13.5.3	at45d_get_info 函数	224
12.5.6	bus_dmamap_unload 函数	207	13.5.4	at45d_wait_for_ device_ready 函数	224
12.6	DMA 映射管理例程 (第 2 部分)	207	13.5.5	at45d_get_status 函数	225
12.7	一个简单的例子	208	13.5.6	at45d_strategy 函数	225
12.8	同步 DMA 缓冲区	210	13.5.7	at45d_task 函数	226
12.9	本章小结	210	13.6	块输入输出完成例程	228
第 13 章	存储驱动	211	13.7	本章小结	228
13.1	disk 结构	211	第 14 章	通用存取方法	229
13.1.1	描述性字段	212	14.1	CAM 是如何工作的	230
13.1.2	存储设备方法字段	212	14.2	一个 (略为) 简单的例子	231
13.1.3	必需的媒体特性 字段	213	14.2.1	mfip_attach 函数	239
13.1.4	可选的媒体特性 字段	213	14.2.2	mfip_detach 函数	240
			14.2.3	mfip_action 函数	240
			14.2.4	mfip_poll 函数	243
			14.2.5	mfip_start 函数	243
			14.2.6	mfip_done 函数	245
			14.3	SIM 注册例程	246
			14.3.1	cam_simq_alloc 函数	247

14.3.2	cam_sim_alloc 函数	247	15.7	让一切配合起来	271
14.3.3	xpt_bus_register 函数	248	15.7.1	ulpt_probe 函数	276
14.4	动作例程	248	15.7.2	ulpt_attach 函数	276
14.4.1	XPT_PATH_INQ 常量	248	15.7.3	ulpt_detach 函数	278
14.4.2	XPT_RESET_BUS 常量	250	15.7.4	ulpt_open 函数	279
14.4.3	XPT_GET_TRAN_ SETTINGS 常量	251	15.7.5	ulpt_reset 函数	279
14.4.4	XPT_SET_TRAN_ SETTINGS 常量	254	15.7.6	unlpt_open 函数	280
14.4.5	XPT_SCSI_IO 常量	255	15.7.7	ulpt_close 函数	281
14.4.6	XPT_RESET_DEV 常量	260	15.7.8	ulpt_ioctl 函数	282
14.5	本章小结	261	15.7.9	ulpt_watchdog 函数	282
第 15 章	USB 驱动	262	15.7.10	ulpt_start_read 函数	282
15.1	关于 USB 设备	262	15.7.11	ulpt_stop_read 函数	283
15.2	关于 USB 设备的更多 信息	263	15.7.12	ulpt_start_write 函数	283
15.3	USB 配置结构	264	15.7.13	ulpt_stop_write 函数	284
15.3.1	必需的字段	265	15.7.14	ulpt_write_callback 函数	284
15.3.2	可选的字段	265	15.7.15	ulpt_read_callback 函数	285
15.3.3	USB 传送标志	266	15.7.16	ulpt_status_callback 函数	287
15.4	USB 传输 (针对 FreeBSD)	267	15.8	本章小结	288
15.5	USB 配置结构管理例程	269	第 16 章	网络驱动程序 I: 数据 结构	289
15.6	USB 方法结构	270	16.1	网络接口结构	289
			16.2	网络接口结构管理例程	292

16.2.1 ether_ifattach 函数	293	第 17 章 网络驱动程序 II: 包的接收与传输	305
16.2.2 ether_ifdetach 函数	294	17.1 数据包接收	305
16.3 网络接口媒体结构	295	17.1.1 em_rxeof 函数	305
16.4 网络接口媒体结构管理例程	295	17.1.2 em_handle_rx 函数	309
16.5 Hello, world	297	17.2 数据包发送	310
16.6 mbuf 结构	300	17.2.1 em_start_locked 函数	310
16.7 消息驱动中断	300	17.2.2 em_txeof 函数	311
16.8 实现 MSI	301	17.3 后数据包发送	314
16.9 MSI 管理例程	303	17.4 本章小结	315
16.10 本章小结	304	参考文献	316

第 ① 章

构建并运行模块

本章概要介绍 FreeBSD 设备驱动程序。我们将先描述 4 种不同类型的 UNIX 设备驱动程序以及它们在 FreeBSD 中是如何对应的，然后描述建立和运行可加载内核模块的基础知识。本章结束前我们还将介绍字符设备驱动。

注意 如果你对上面用到的术语暂时还不理解，请不用着急，本章将会给出它们的具体定义。

1.1 设备驱动程序的类型

在 FreeBSD 中，设备（device）是指系统中所有与硬件相关的对象，包括磁盘驱动器、打印机、显卡等，而设备驱动程序（device driver）是指控制或驱动一个或多个设备的计算机程序。在传统的 UNIX 以及 FreeBSD 的 4.0 版本以前，共有 4 种不同类型的设备驱动程序，它们分别是：

- 字符设备驱动程序（character driver）：控制字符设备。
- 块设备驱动程序（block driver）：控制块设备。
- 网络驱动程序（network driver）：控制网络设备。
- 伪设备驱动程序（pseudo-device driver）：控制伪设备。

字符设备（character device）提供的是面向字符流的输入输出接口，这类接口也称为无结构（unstructured）接口或原生（raw）接口（McKusick and Neville-Neil, 2005）。

块设备（block device）以固定大小的块传送可随机存取的数据（Corbet et al., 2005）。FreeBSD 自发布 4.0 版以后取消了块设备驱动程序（关于这一点的更多内容请参见 1.7 节）。

网络设备（network device）发送并接收由网络子系统所驱动的数据包（Corbet et al., 2005）。