

图书基本信息

书名：<<深入理解FreeBSD设备驱动程序开发>>

13位ISBN编号：9787111411574

10位ISBN编号：7111411579

出版时间：2013-2-28

出版时间：机械工业出版社华章公司

作者：Joseph Kong

译者：陈毅东

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：<http://www.tushu007.com>

前言

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的优秀入门指南。

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。

【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

内容概要

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。

作者简介

作者：（加拿大）约瑟夫（Joseph, K.）译者：陈毅东Joseph Kong，资深FreeBSD设备驱动编程专家，具有20多年的计算机硬件、软件和驱动程序的设计与开发经验。对信息安全、操作系统理论、逆向工程和安全漏洞评估等技术也有非常深入的研究。此外，他还是一位知名的技术作家和技术顾问，著有《Designing BSD Rootkits》一书，担任过多本与设备驱动程序开发相关图书的技术审稿人，在权威计算机杂志上发表了多篇与设备驱动程序开发相关的文章，深受读者喜爱。

书籍目录

译者序前言前言致谢第1章构建并运行模块11.1 设备驱动程序的类型11.2 可加载内核模块21.2.1 模块事件处理程序21.2.2 DECLARE_MODULE宏31.3 Hello, world!51.4 编译与加载61.5 字符设备驱动程序71.5.1 d_foo函数81.5.2 字符设备开关表81.5.3 make_dev函数和des-troy_dev函数91.6 实例：内存字符设备驱动程序101.6.1 echo_write函数131.6.2 echo_read函数141.6.3 echo_modevent函数141.6.4 DEV_MODULE宏151.6.5 试运行161.7 块设备驱动程序消失了161.8 本章小结17第2章分配内存182.1 内存管理例程182.2 malloc_type结构192.2.1 MALLOC_DE-FINE宏202.2.2 MALLOC_DE-CLARE宏202.3 让一切配合起来212.4 连续物理内存管理例程232.5 一个简单的例子242.6 本章小结26第3章设备通信与控制273.1 ioctl接口273.2 定义ioctl命令293.3 实现ioctl303.3.1 echo_write函数343.3.2 echo_set_buffer_size函数353.3.3 echo_ioctl函数363.3.4 echo_modevent函数373.3.5 试运行383.4 调用ioctl383.5 sysctl413.6 实现sysctl(第1部分)413.7 sysctl上下文管理例程453.8 创建动态的sysctl453.8.1 SYSCTL_STATIC_CHILDREN宏483.8.2 SYSCTL_CHILDREN宏483.9 实现sysctl(第2部分)483.9.1 sysctl_set_buffer_size函数523.9.2 试运行533.10 本章小结53第4章线程同步544.1 一个简单的同步问题544.2 一个更复杂的同步问题554.2.1 race_new函数594.2.2 race_find函数594.2.3 race_destroy函数604.2.4 race_ioctl函数604.2.5 race_modevent函数614.2.6 问题的根源624.3 防止竞态条件664.4 互斥锁674.4.1 自旋互斥锁674.4.2 休眠互斥锁674.5 互斥锁管理例程684.6 实现互斥锁704.6.1 race_modevent函数734.6.2 试运行744.7 共享/独占锁754.8 共享/独占锁管理例程754.9 实现共享/独占锁774.10 读者/写者锁814.11 读者/写者锁管理例程814.12 条件变量834.13 条件变量管理例程834.14 一般性原则854.14.1 避免在独占锁上递归854.14.2 避免长时间占用独占锁854.15 本章小结85第5章延迟执行865.1 主动上下文切换/休眠865.2 实现休眠与条件变量885.2.1 sleep_modevent函数925.2.2 load函数925.2.3 sleep_thread函数935.2.4 sysctl_debug_sleep_test函数945.2.5 unload函数955.2.6 试运行965.3 内核事件处理器965.4 callout985.5 callout和竞态条件1005.6 任务队列1015.6.1 全局任务队列1015.6.2 任务队列管理例程1025.7 本章小结103第6章案例研究：虚拟空调制解调器1046.1 前期基础1046.2 代码分析1056.2.1 nmdm_modevent函数1076.2.2 nmdm_clone函数1086.2.3 nmdm_alloc函数1106.2.4 nmdm_outwakeup函数1116.2.5 nmdm_task_tty函数1126.2.6 nmdm_inwakeup函数1136.2.7 nmdm_modem函数1146.2.8 nmdm_param函数1156.2.9 nmdm_timeout函数1166.2.10 bits_per_char函数1176.2.11 试运行1186.3 本章小结118第7章Newbus和资源分配1197.1 自动配置与Newbus驱动程序1197.1.1 device_foo函数1197.1.2 设备方法表1217.1.3 DRIVER_MODULE宏1217.2 驱动程序示例1227.2.1 foo_pci_probe函数1257.2.2 foo_pci_attach函数1267.2.3 d_foo函数1277.2.4 foo_pci_detach函数1277.2.5 试运行1287.3 硬件资源管理1287.4 本章小结130第8章中断处理1318.1 注册中断处理程序1318.2 FreeBSD中的中断处理程序1328.3 实现中断处理程序1338.3.1 pint_identify函数1388.3.2 pint_probe函数1388.3.3 pint_attach函数1398.3.4 pint_detach函数1408.3.5 pint_open函数1408.3.6 pint_close函数1428.3.7 pint_write函数1428.3.8 pint_read函数1438.3.9 pint_intr函数1448.3.10 试运行1448.4 生成并口中断1458.5 本章小结146第9章案例研究：并口打印机驱动程序1479.1 代码分析1479.1.1 lpt_identify函数1529.1.2 lpt_probe函数1529.1.3 lpt_detect函数1529.1.4 lpt_port_test函数1539.1.5 lpt_attach函数1549.1.6 lpt_detach函数1569.1.7 lpt_open函数1579.1.8 lpt_read函数1599.1.9 lpt_write函数1619.1.10 lpt_intr函数1639.1.11 lpt_timeout函数1649.1.12 lpt_push_bytes函数1659.1.13 lpt_close函数1669.1.14 lpt_ioctl函数1679.1.15 lpt_request_ppbus函数1699.1.16 lpt_release_ppbus函数1699.2 本章小结170第10章管理与使用资源17110.1 输入输出端口与输入输出内存17110.1.1 从输入输出端口和输入输出内存读取数据17110.1.2 向输入输出端口和输入输出内存写入数据17310.1.3 流操作17510.2 内存栅栏17810.3 让一切配合起来17910.3.1 led_identify函数18310.3.2 led_probe函数18410.3.3 led_attach函数18410.3.4 led_detach函数18510.3.5 led_open函数18610.3.6 led_close函数18610.3.7 led_read函数18710.3.8 led_write函数18810.4 本章小结188第11章案例研究：智能平台管理接口驱动18911.1 代码分析18911.1.1 ipmi_pci_probe函数19111.1.2 ipmi_pci_match函数19211.1.3 ipmi_pci_attach函数19211.1.4 ipmi2_pci_probe函数19511.1.5 ipmi2_pci_attach函数19511.2 本章小

结197第12章直接内存访问19812.1 实现DMA19812.1.1 初始化DMA数据传输20012.1.2 取消DMA20112.2 创建DMA标签20112.3 销毁DMA标签20312.4 DMA映射管理例程(第1部分) 20412.5 将(DMA)缓冲区载入DMA映射20412.5.1 bus_dma_segment结构20512.5.2 bus_dmamap_load函数20512.5.3 bus_dmamap_load_mbuf函数20612.5.4 bus_dmamap_load_mbuf_sg函数20712.5.5 bus_dmamap_load_uio函数20712.5.6 bus_dmamap_unload函数20712.6 DMA映射管理例程(第2部分) 20712.7 一个简单的例子20812.8 同步DMA缓冲区21012.9 本章小结210第13章存储驱动21113.1 disk结构21113.1.1 描述性字段21213.1.2 存储设备方法字段21213.1.3 必需的媒体特性字段21313.1.4 可选的媒体特性字段21313.1.5 驱动程序私有数据字段21313.2 disk结构管理例程21413.3 块输入输出结构21413.4 块输入输出队列21613.5 让一切配合起来21713.5.1 at45d_attach函数22213.5.2 at45d_delayed_attach函数22313.5.3 at45d_get_info函数22413.5.4 at45d_wait_for_device_ready函数22413.5.5 at45d_get_status函数22513.5.6 at45d_strategy函数22513.5.7 at45d_task函数22613.6 块输入输出完成例程22813.7 本章小结228第14章通用存取方法22914.1 CAM是如何工作的23014.2 一个(略为)简单的例子23114.2.1 mfip_attach函数23914.2.2 mfip_detach函数24014.2.3 mfip_action函数24014.2.4 mfip_poll函数24314.2.5 mfip_start函数24314.2.6 mfip_done函数24514.3 SIM注册例程24614.3.1 cam_simq_alloc函数24714.3.2 cam_sim_alloc函数24714.3.3 xpt_bus_register函数24814.4 动作例程24814.4.1 XPT_PATH_INQ常量24814.4.2 XPT_RESET_BUS常量25014.4.3 XPT_GET_TRAN_SETTINGS常量25114.4.4 XPT_SET_TRAN_SETTINGS常量25414.4.5 XPT SCSI_IO常量25514.4.6 XPT_RESET_DEV常量26014.5 本章小结261第15章USB驱动26215.1 关于USB设备26215.2 关于USB设备的更多信息26315.3 USB配置结构26415.3.1 必需的字段26515.3.2 可选的字段26515.3.3 USB传送标志26615.4 USB传输(针对FreeBSD) 26715.5 USB配置结构管理例程26915.6 USB方法结构27015.7 让一切配合起来27115.7.1 ulpt_probe函数27615.7.2 ulpt_attach函数27615.7.3 ulpt_detach函数27815.7.4 ulpt_open函数27915.7.5 ulpt_reset函数27915.7.6 unlpt_open函数28015.7.7 ulpt_close函数28115.7.8 ulpt_ioctl函数28215.7.9 ulpt_watchdog函数28215.7.10 ulpt_start_read函数28215.7.11 ulpt_stop_read函数28315.7.12 ulpt_start_write函数28315.7.13 ulpt_stop_write函数28415.7.14 ulpt_write_callback函数28415.7.15 ulpt_read_callback函数28515.7.16 ulpt_status_callback函数28715.8 本章小结288第16章网络驱动程序I: 数据结构28916.1 网络接口结构28916.2 网络接口结构管理例程29216.2.1 ether_ifattach函数29316.2.2 ether_ifdetach函数29416.3 网络接口媒体结构29516.4 网络接口媒体结构管理例程29516.5 Hello, world29716.6 mbuf结构30016.7 消息驱动中断30016.8 实现MSI30116.9 MSI管理例程30316.10 本章小结304第17章网络驱动程序II: 包的接收与传输30517.1 数据包接收30517.1.1 em_rxeof函数30517.1.2 em_handle_rx函数30917.2 数据包发送31017.2.1 em_start_locked函数31017.2.2 em_txeof函数31117.3 后数据包发送31417.4 本章小结315参考文献316

章节摘录

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）。

伪设备（pseudo-device）是一个使用纯软件来模拟设备行为的计算机程序。

所谓纯软件意味着它不对应任何下层硬件设备。

编辑推荐

《深入理解FreeBSD设备驱动程序开发》编辑推荐：资深FreeBSD设备驱动编程专家20余年工作经验结晶，FreeBSD内核开发专家鼎力推荐。

从程序员的视角出发，以实际的驱动开发为案例，系统全面地讲解了FreeBSD设备驱动开发的各个方面，为编写健壮高效的驱动程序提供优秀指导。

名人推荐

本书详尽地讲解了FreeBSD上最通用的设备驱动程序的接口，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；高层接口包括连接网络以及存储栈等。

此外，本书内容还包括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。

对于FreeBSD设备驱动程序开发人员而言，本书是一份难得的参考资料。

——John Baldwin 资深FreeBSD内核开发专家

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:<http://www.tushu007.com>