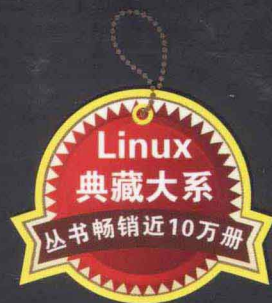
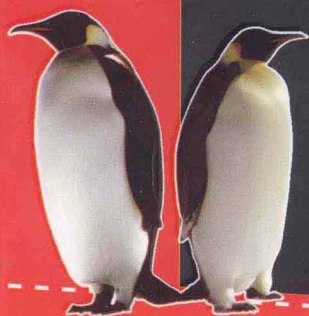


Linux驱动开发精品图书全面升级，ChinaUnix社区鼎力推荐
由浅入深，全面、系统地介绍Linux驱动开发的基础与核心技术
注重实战，重点阐述8种典型的Linux设备驱动程序的开发方法



Linux

郑强 等编著

驱动开发入门与实战 (第2版)

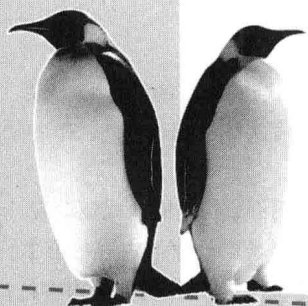
本书源代码及PPT下载网址：www.tup.com.cn或www.wanjuanchina.net

- ◎ 内容新颖：基于当前最流行的Linux 2.6.34内核，讲解最新的驱动开发
- ◎ 内容全面：涵盖Linux驱动开发的基础知识、核心技术与典型应用
- ◎ 讲解详细：结合典型开发实例与案例讲解各知识点，并力求详尽和透彻
- ◎ 注重实战：介绍了看门狗、LCD、触摸屏等8种典型设备的驱动开发方法
- ◎ 技巧性强：讲解时融入了作者多年的项目开发经验和实战技巧



清华大学出版社

典藏大家



Linux

郑强 等编著

驱动开发入门与实战

(第2版)

清华大学出版社

北 京

内 容 简 介

本书是获得了大量读者好评的“Linux 典藏大系”中的《Linux 驱动开发入门与实战》的第2版。本书由浅入深,全面、系统地介绍了Linux驱动开发技术,并提供了大量实例供读者实战演练。另外,作者在实例讲解中详细分析了各种重要的理论知识,让读者能够举一反三。

本书共分3篇。第1篇介绍了Linux驱动开发概述、嵌入式处理器和开发板简介、构建嵌入式驱动程序开发环境、构建嵌入式Linux操作系统、构建第一个驱动程序、简单的字符设备驱动程序等内容;第2篇介绍了设备驱动中的并发控制、设备驱动中的阻塞和同步机制、中断与时钟机制、内外存访问等内容;第3篇介绍了设备驱动模型、RTC实时时钟驱动程序、看门狗驱动程序、IIC设备驱动程序、LCD设备驱动程序、触摸屏设备驱动程序、输入子系统设计、块设备驱动程序、USB设备驱动程序等内容。

本书重点突出,涉及面广,实用性强,从基本知识到核心原理,再到实例开发,几乎涉及Linux驱动开发的所有重要知识。本书适合所有想学习Linux驱动开发的入门人员阅读,也适合作为驱动工程师的参考书,对于Linux驱动开发的专业开发人员也有很高的参考价值。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Linux 驱动开发入门与实战(第2版)/郑强等编著. —北京:清华大学出版社,2014

(Linux 典藏大系)

ISBN 978-7-302-33776-8

I. ①L… II. ①郑… III. ①Linux 操作系统 IV. ①TP316.89

中国版本图书馆CIP数据核字(2013)第211329号

责任编辑:夏兆彦

封面设计:欧振旭

责任校对:徐俊伟

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京鑫丰华彩印有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:28.5

字 数:707千字

版 次:2010年1月第1版 2014年2月第2版

印 次:2014年2月第1次印刷

印 数:5001~9000

定 价:69.00元

产品编号:052844-01

前 言

Linux 驱动程序开发是当前一个非常热门的领域，大多数基于 Linux 操作系统的嵌入式系统都需要编写驱动程序。随着嵌入式系统的广泛应用，出现了越来越多的硬件产品，必须有人不断地编写驱动使设备在 Linux 操作系统上工作。但是，Linux 驱动程序开发相对较难，高水平的开发人员也比较少，所以导致驱动程序跟不上硬件发展的问题。基于这个原因，笔者编写了这本书，希望借助本书能使驱动程序的开发更容易被开发人员所理解，从而迅速、高效地开发出相关的驱动程序来。

本书是获得了大量读者好评的“Linux 典藏大系”中的《Linux 驱动开发入门与实战》的第 2 版。在第 1 版的基础上，本书进行了全新改版，升级了 Linux 系统的编程环境，也升级了 Linux 的内核版本，并对书中的一些疏漏进行了修订，也对书中的一些实例和代码进行了重新表述，使得更加易读。相信读者可以在本书的引领下跨入 Linux 驱动开发大门，并成为一名驱动程序开发高手。

关于“Linux 典藏大系”

“Linux 典藏大系”是清华大学出版社自 2010 年 1 月以来陆续推出的一个图书系列，截止 2012 年，已经出版了 10 余个品种。该系列图书涵盖了 Linux 技术的方方面面，可以满足各个层次和各个领域的读者学习 Linux 技术的需求。该系列图书自出版以来获得了广大读者的好评，已经成为了 Linux 图书市场上最耀眼的明星品牌之一。其销量在同类图书中也名列前茅，其中一些图书还获得了“51CTO 读书频道”颁发的“最受读者喜爱的原创 IT 技术图书奖”。该系列图书出版过程中也得到了国内 Linux 领域最知名的技术社区 ChinaUnix（简称 CU）的大力支持和帮助，读者在 CU 社区中就图书的内容与活跃在 CU 社区中的 Linux 技术爱好者进行广泛交流，取得了良好的学习效果。

关于本书第 2 版

本书第 1 版出版后深受读者好评，并被 ChinaUNIX 技术社区所推荐。但是随着 Linux 技术的发展，本书第 1 版的内容与 Linux 各个新版本有一定出入，这给读者的学习造成了一些不便。应广大读者的要求，我们结合 Linux 技术的最新发展推出第 2 版图书。相比第 1 版，第 2 版图书在内容上的变化主要体现在以下几个方面：

- (1) Linux 系统由 Fedora 9 升级为 Fedora 18。
- (2) Linux 内核版本由 Linux 2.6.29 升级为 Linux 2.6.34。
- (3) 更新 Linux 驱动最新的开发接口。
- (4) 对最新的嵌入式处理器接口进行了介绍，更新了驱动程序的写法。
- (5) 优化了驱动程序的代码，让代码更易懂。

- (6) 对一些难懂的概念列举了例子，使读者更容易理解。
- (7) 对第1版中没讲到的复杂算法进行了讲解。
- (8) 修订了第1版中的一些疏漏，并将一些表达不准确的地方表述得更加准确。

本书特色

1. 最新内核，了解最新开发技术

本书基于 Linux 2.6.34 内核，这是目前较新的一个内核。该内核包含了大多数常用的驱动程序，便于学习和移植。

2. 内容全面、系统、深入

本书介绍了 Linux 驱动开发的基础知识、核心技术和一些驱动程序开发实例。内容的安排上力求全面、系统。在实例的选择上力求深入。

3. 讲解由浅入深、循序渐进，适合各个层次的读者阅读

本书从 Linux 驱动程序开发的基础开始讲解，逐步深入到 Linux 驱动的高级开发技术及应用，内容安排从易到难，讲解由浅入深、循序渐进，适合各个层次的读者阅读。

4. 贯穿大量的开发实例和技巧，迅速提升开发水平

本书在讲解知识点时穿插了大量驱动程序的典型实例，并给出了大量的开发技巧，以便让读者更好地理解各种概念和开发技术，体验实际编程，迅速提高开发水平。

5. 从工程应用出发，具有很强的实用性

本书详细介绍了多个驱动开发实例。通过这些应用实例，可以提高读者的驱动开发水平，从而具备独立进行驱动程序开发的能力。

本书内容及知识体系

第1篇 Linux 驱动开发基础（第1~6章）

本篇主要包括：Linux 驱动开发概述、嵌入式处理器和开发板简介、构建嵌入式驱动程序开发环境、构建嵌入式 Linux 操作系统、构建第一个驱动程序、简单的字符设备驱动程序。通过对本篇内容的学习，读者可以掌握 Linux 驱动开发的基本概念和基本环境。

第2篇 Linux 驱动开发核心技术（第7~10章）

本篇主要包括：设备驱动中的并发控制、设备驱动中的阻塞和同步机制、中断与时钟机制、内外存访问等内容。通过本篇的学习，读者可以掌握 Linux 驱动开发的基础知识和核心技术。

第3篇 Linux 驱动开发应用实战（第11~19章）

本篇主要包括：设备驱动模型、RTC 实时时钟驱动程序、看门狗驱动程序、IIC

设备驱动程序、LCD 设备驱动程序、触摸屏设备驱动程序、输入子系统设计、块设备驱动程序、USB 设备驱动程序等。通过对本篇内容的学习，读者可以掌握编写各种设备驱动程序的方法。

本书读者对象

- ☐ Linux 内核爱好者；
- ☐ 想学习 Linux 驱动开发的入门人员；
- ☐ Linux 驱动程序专业开发人员；
- ☐ 嵌入式工程师；
- ☐ 大中专院校的学生；
- ☐ 社会培训班的学员；
- ☐ 需要了解驱动程序开发的技术人员。

本书作者

本书由郑强主笔编写。其他参与编写的人员有陈杰、陈贞、樊俊、高彩丽、高莹婷、管磊、郭丽、韩亚、李红、李龙海、梁伟、刘忆智、曲宝军、孙忠贤、唐正兵、王全政、王勇浩、武文琛、徐学英、闫伍平、于轶、占海明、张帆。

您在阅读本书的过程中若碰到什么问题，请通过以下方式联系我们，我们会及时地答复您。

E-mail: book@wanjuanchina.net 或 bookservice2008@163.com

论坛网址: <http://www.wanjuanchina.net>

编者

目 录

第 1 篇 Linux 驱动开发基础

第 1 章 Linux 驱动开发概述	2
1.1 Linux 设备驱动的基本概念	2
1.1.1 设备驱动程序概述	2
1.1.2 设备驱动程序的作用	2
1.1.3 设备驱动的分类	3
1.2 Linux 操作系统与驱动的关系	4
1.3 Linux 驱动程序开发	4
1.3.1 用户态和内核态	4
1.3.2 模块机制	5
1.3.3 编写设备驱动程序需要了解的知识	6
1.4 编写设备驱动程序的注意事项	6
1.4.1 应用程序开发与驱动程序开发的差异	6
1.4.2 GUN C 开发驱动程序	7
1.4.3 不能使用 C 库开发驱动程序	7
1.4.4 没有内存保护机制	8
1.4.5 小内核栈	8
1.4.6 重视可移植性	8
1.5 Linux 驱动的发展趋势	9
1.5.1 Linux 驱动的发展	9
1.5.2 驱动的应用	9
1.5.3 相关学习资源	9
1.6 小结	10
第 2 章 嵌入式处理器和开发板简介	11
2.1 处理器的选择	11
2.1.1 处理器简述	11
2.1.2 处理器的种类	11
2.2 ARM 处理器	13
2.2.1 ARM 处理器简介	14
2.2.2 ARM 处理器系列	14
2.2.3 ARM 处理器的应用	16

2.2.4	ARM 处理器的选型	16
2.2.5	ARM 处理器选型举例	19
2.3	S3C2440 开发板	20
2.3.1	S3C2440 开发板简介	20
2.3.2	S3C2440 开发板的特性	20
2.3.3	其他开发板	22
2.4	小结	22
第 3 章	构建嵌入式驱动程序开发环境	23
3.1	虚拟机和 Linux 安装	23
3.1.1	在 Windows 上安装虚拟机	23
3.1.2	在虚拟机上安装 Linux	27
3.1.3	设置共享目录	29
3.2	代码阅读工具 Source Insight	29
3.2.1	Source Insight 简介	30
3.2.2	阅读源代码	30
3.3	小结	33
第 4 章	构建嵌入式 Linux 操作系统	34
4.1	Linux 操作系统的介绍	34
4.1.1	Linux 操作系统	34
4.1.2	Linux 操作系统的优点	35
4.2	Linux 内核子系统	36
4.2.1	进程管理	36
4.2.2	内存管理	37
4.2.3	文件系统	37
4.2.4	设备管理	38
4.2.5	网络功能	38
4.3	Linux 源代码结构分析	38
4.3.1	arch 目录	38
4.3.2	drivers 目录	39
4.3.3	fs 目录	39
4.3.4	其他目录	40
4.4	内核配置选项	41
4.4.1	配置编译过程	41
4.4.2	常规配置	42
4.4.3	模块配置	44
4.4.4	块设备层配置	44
4.4.5	CPU 类型和特性配置	45
4.4.6	电源管理配置	47
4.4.7	总线配置	49
4.4.8	网络配置	50

4.4.9 设备驱动配置	53
4.4.10 文件系统配置	60
4.5 嵌入式文件系统基础知识	62
4.5.1 嵌入式文件系统	62
4.5.2 嵌入式系统的存储介质	63
4.5.3 JFFS 文件系统	64
4.5.4 YAFFS 文件系统	64
4.6 构建根文件系统	64
4.6.1 根文件系统概述	65
4.6.2 Linux 根文件系统目录结构	65
4.6.3 BusyBox 构建根文件系统	66
4.7 小结	71
第 5 章 构建第一个驱动程序	72
5.1 开发环境配置之内核升级	72
5.1.1 为什么升级内核	72
5.1.2 内核升级	73
5.1.3 make menconfig 的注意事项	75
5.2 Hello World 驱动程序	77
5.2.1 驱动模块的组成	77
5.2.2 Hello World 模块	78
5.2.3 编译 Hello World 模块	79
5.2.4 模块的操作	81
5.2.5 Hello World 模块加载后文件系统的变化	82
5.3 模块参数和模块之间通信	82
5.3.1 模块参数	83
5.3.2 模块的文件格式 ELF	83
5.3.3 模块之间的通信	84
5.3.4 模块之间的通信实例	84
5.4 将模块加入内核	88
5.4.1 向内核添加模块	88
5.4.2 Kconfig	88
5.4.3 Kconfig 的语法	89
5.4.4 应用实例：在内核中新增加 add_sub 模块	92
5.4.5 对 add_sub 模块进行配置	93
5.5 小结	95
第 6 章 简单的字符设备驱动程序	96
6.1 字符设备驱动程序框架	96
6.1.1 字符设备和块设备	96
6.1.2 主设备号和次设备号	97
6.1.3 申请和释放设备号	98

6.2	初识 cdev 结构	99
6.2.1	cdev 结构体	99
6.2.2	file_operations 结构体	101
6.2.3	cdev 和 file_operations 结构体的关系	102
6.2.4	inode 结构体	102
6.3	字符设备驱动的组成	103
6.3.1	字符设备加载和卸载函数	103
6.3.2	file_operations 结构体和其成员函数	104
6.3.3	驱动程序与应用程序的数据交换	105
6.3.4	字符设备驱动程序组成小结	106
6.4	VirtualDisk 字符设备驱动	106
6.4.1	VirtualDisk 的头文件、宏和设备结构体	106
6.4.2	加载和卸载驱动程序	107
6.4.3	cdev 的初始化和注册	108
6.4.4	打开和释放函数	109
6.4.5	读写函数	110
6.4.6	seek()函数	111
6.4.7	ioctl()函数	112
6.5	小结	113

第 2 篇 Linux 驱动开发核心技术

第 7 章	设备驱动中的并发控制	116
7.1	并发与竞争	116
7.2	原子变量操作	116
7.2.1	原子变量操作	116
7.2.2	原子整型操作	117
7.2.3	原子位操作	119
7.3	自旋锁	120
7.3.1	自旋锁概述	120
7.3.2	自旋锁的使用	120
7.3.3	自旋锁的使用注意事项	122
7.4	信号量	122
7.4.1	信号量概述	122
7.4.2	信号量的实现	123
7.4.3	信号量的使用	123
7.4.4	自旋锁与信号量的对比	125
7.5	完成量	126
7.5.1	完成量概述	126
7.5.2	完成量的实现	126

7.5.3 完成量的使用	127
7.6 小结	128
第 8 章 设备驱动中的阻塞和同步机制	129
8.1 阻塞和非阻塞	129
8.2 等待队列	130
8.2.1 等待队列概述	130
8.2.2 等待队列的实现	130
8.2.3 等待队列的使用	130
8.3 同步机制实验	132
8.3.1 同步机制设计	132
8.3.2 实验验证	136
8.4 小结	137
第 9 章 中断与时钟机制	138
9.1 中断简述	138
9.1.1 中断的概念	138
9.1.2 中断的宏观分类	139
9.1.3 中断产生的位置分类	139
9.1.4 同步和异步中断	140
9.1.5 中断小结	140
9.2 中断的实现过程	140
9.2.1 中断信号线 (IRQ)	140
9.2.2 中断控制器	141
9.2.3 中断处理过程	141
9.2.4 中断的安装与释放	142
9.3 按键中断实例	143
9.3.1 按键设备原理图	143
9.3.2 有寄存器设备和无寄存器设备	144
9.3.3 按键设备相关端口寄存器	144
9.4 按键中断实例程序分析	146
9.4.1 按键驱动程序组成	147
9.4.2 初始化函数 s3c2440_buttons_init()	147
9.4.3 中断处理函数 isr_button()	148
9.4.4 退出函数 s3c2440_buttons_exit()	149
9.5 时钟机制	149
9.5.1 时间度量	149
9.5.2 时间延时	150
9.6 小结	151
第 10 章 内外存访问	152
10.1 内存分配	152
10.1.1 kmalloc()函数	152

10.1.2	vmalloc()函数	153
10.1.3	后备高速缓存	155
10.2	页面分配	156
10.2.1	内存分配	156
10.2.2	物理地址和虚拟地址之间的转换	159
10.3	设备 I/O 端口的访问	160
10.3.1	Linux I/O 端口读写函数	160
10.3.2	I/O 内存读写	160
10.3.3	使用 I/O 端口	164
10.4	小结	166

第3篇 Linux 驱动开发实用实战

第11章	设备驱动模型	168
11.1	设备驱动模型概述	168
11.1.1	设备驱动模型的功能	168
11.1.2	sysfs 文件系统	169
11.1.3	sysfs 文件系统的目录结构	170
11.2	设备驱动模型的核心数据结构	171
11.2.1	kobject 结构体	171
11.2.2	设备属性 kobj_type	175
11.3	注册 kobject 到 sysfs 中的实例	179
11.3.1	设备驱动模型结构	179
11.3.2	kset 集合	179
11.3.3	kset 与 kobject 的关系	181
11.3.4	kset 相关的操作函数	182
11.3.5	注册 kobject 到 sysfs 中的实例	183
11.3.6	实例测试	187
11.4	设备驱动模型的三大组件	188
11.4.1	总线	188
11.4.2	总线属性和总线方法	191
11.4.3	设备	193
11.4.4	驱动	195
11.5	小结	198
第12章	RTC 实时时钟驱动	199
12.1	RTC 实时时钟硬件原理	199
12.1.1	RTC 实时时钟	199
12.1.2	RTC 实时时钟的功能	199
12.1.2	RTC 实时时钟的工作原理	201

12.2	RTC 实时时钟架构	205
12.2.1	加载卸载函数	205
12.2.2	RTC 实时时钟的平台驱动	206
12.2.3	RTC 驱动探测函数	207
12.2.4	RTC 实时时钟的使能函数 <code>s3c_rtc_enable()</code>	210
12.2.5	RTC 实时时钟设置频率函数 <code>s3c_rtc_setfreq()</code>	211
12.2.6	RTC 设备注册函数 <code>rtc_device_register()</code>	212
12.3	RTC 文件系统接口	214
12.3.1	文件系统接口 <code>rtc_class_ops</code>	214
12.3.2	RTC 实时时钟打开函数 <code>s3c_rtc_open()</code>	215
12.3.3	RTC 实时时钟关闭函数 <code>s3c_rtc_release()</code>	216
12.3.4	RTC 实时时钟获得时间函数 <code>s3c_rtc_gettime()</code>	216
12.3.5	RTC 实时时钟设置时间函数 <code>s3c_rtc_settime()</code>	218
12.3.6	RTC 驱动探测函数 <code>s3c_rtc_getalarm()</code>	219
12.3.7	RTC 实时时钟设置报警时间函数 <code>s3c_rtc_setalarm()</code>	220
12.3.8	RTC 设置脉冲中断使能函数 <code>s3c_rtc_setpie()</code>	221
12.3.9	RTC 时钟脉冲中断判断函数 <code>s3c_rtc_proc()</code>	222
12.4	小结	222
第 13 章	看门狗驱动程序	223
13.1	看门狗硬件原理	223
13.1.1	看门狗	223
13.1.2	看门狗工作原理	223
13.2	平台设备模型	225
13.2.1	平台设备模型	226
13.2.2	平台设备	226
13.2.3	平台设备驱动	228
13.2.4	平台设备驱动的注册和注销	229
13.2.5	混杂设备	230
13.2.6	混杂设备的注册和注销	231
13.3	看门狗设备驱动程序分析	231
13.3.1	看门狗驱动程序的一些变量定义	231
13.3.2	看门狗模块的加载和卸载函数	232
13.3.3	看门狗驱动程序探测函数	233
13.3.4	设置看门狗复位时间函数 <code>s3c2410wdt_set_heartbeat()</code>	234
13.3.5	看门狗的开始函数 <code>s3c2410wdt_start()</code> 和停止函数 <code>s3c2410wdt_stop()</code>	236
13.3.6	看门狗驱动程序移除函数 <code>s3c2410wdt_remove()</code>	237
13.3.7	平台设备驱动 <code>s3c2410wdt_driver</code> 中的其他重要函数	237
13.3.8	混杂设备的 <code>file_operations</code> 中的函数	238
13.3.9	看门狗中断处理函数 <code>s3c2410wdt_irq()</code>	241

13.4 小结	242
第 14 章 IIC 设备驱动程序	243
14.1 IIC 设备的总线及其协议	243
14.1.1 IIC 总线的特点	243
14.1.2 IIC 总线的信号类型	244
14.1.3 IIC 总线的数据传输	244
14.2 IIC 设备的硬件原理	244
14.3 IIC 设备驱动程序的层次结构	246
14.3.1 IIC 设备驱动的概述	246
14.3.2 IIC 设备层	247
14.3.3 i2c_driver 和 i2c_client 的关系	249
14.3.4 IIC 总线层	249
14.3.5 IIC 设备层和总线层的关系	251
14.3.6 写 IIC 设备驱动的步骤	251
14.4 IIC 子系统的初始化	252
14.4.1 IIC 子系统初始化函数 i2c_init()	252
14.4.2 IIC 子系统退出函数 i2c_exit()	253
14.5 适配器驱动程序	253
14.5.1 s3c2440 对应的适配器结构体	253
14.5.2 IIC 适配器加载函数 i2c_add_adapter()	255
14.5.3 IDR 机制	256
14.5.4 适配器卸载函数 i2c_del_adapter()	258
14.5.5 IIC 总线通信方法 s3c24xx_i2c_algorithm 结构体	258
14.5.6 适配器的传输函数 s3c24xx_i2c_doxfer()	260
14.5.7 适配器的中断处理函数 s3c24xx_i2c_irq()	263
14.5.8 字节传输函数 i2s_s3c_irq_nextbyte()	265
14.5.9 适配器传输停止函数 s3c24xx_i2c_stop()	267
14.5.10 中断处理函数的一些辅助函数	268
14.6 IIC 设备层驱动程序	268
14.6.1 IIC 设备驱动模块加载和卸载	269
14.6.2 探测函数 s3c24xx_i2c_probe()	270
14.6.3 移除函数 s3c24xx_i2c_remove()	272
14.6.4 控制器初始化函数 s3c24xx_i2c_init()	273
14.6.5 设置控制器数据发送频率函数 s3c24xx_i2c_clockrate()	274
14.7 小结	276
第 15 章 LCD 设备驱动程序	277
15.1 FrameBuffer 概述	277
15.1.1 FrameBuffer 的概念	277
15.1.2 FrameBuffer 与应用程序的交互	278
15.1.3 FrameBuffer 显示原理	278

15.1.4	LCD 显示原理	278
15.2	Framebuffer 的结构分析	279
15.2.1	Framebuffer 架构和其关系	279
15.2.2	Framebuffer 驱动程序的实现	280
15.2.3	Framebuffer 架构及其关系	281
15.3	LCD 驱动程序分析	286
15.3.1	LCD 模块的加载和卸载函数	286
15.3.2	LCD 驱动程序的平台数据	287
15.3.3	LCD 模块的探测函数	289
15.3.4	移除函数	293
15.4	小结	294
第 16 章	触摸屏设备驱动程序	295
16.1	触摸屏设备工作原理	295
16.1.1	触摸屏设备概述	295
16.1.2	触摸屏设备的类型	295
16.1.3	电阻式触摸屏	296
16.2	触摸屏设备硬件结构	296
16.2.1	s3c2440 触摸屏接口概述	296
16.2.2	s3c2440 触摸屏接口的工作模式	297
16.2.3	s3c2440 触摸屏设备寄存器	297
16.3	触摸屏设备驱动程序分析	301
16.3.1	触摸屏设备驱动程序组成	301
16.3.2	s3c2440 触摸屏驱动模块的加载和卸载函数	302
16.3.3	s3c2440 触摸屏驱动模块的探测函数	303
16.3.4	触摸屏设备配置	305
16.3.5	触摸屏设备中断处理函数	307
16.3.6	s3c2440 触摸屏驱动模块的 remove()函数	311
16.4	测试触摸屏驱动程序	312
16.5	小结	313
第 17 章	输入子系统设计	314
17.1	input 子系统入门	314
17.1.1	简单的实例	314
17.1.2	注册函数 input_register_device()	316
17.1.3	向子系统报告事件	319
17.2	handler 注册分析	324
17.2.1	输入子系统的组成	325
17.2.2	input_handler 结构体	325
17.2.3	注册 input_handler	326
17.2.4	input_handle 结构体	327
17.2.5	注册 input_handle	328

17.3	input 子系统.....	329
17.3.1	子系统初始化函数 <code>input_init()</code>	329
17.3.2	文件打开函数 <code>input_open_file()</code>	330
17.4	evdev 输入事件驱动分析	331
17.4.1	evdev 的初始化.....	331
17.4.2	evdev 设备的打开.....	334
17.5	小结.....	336
第 18 章	块设备驱动程序	337
18.1	块设备简介	337
18.1.1	块设备总体概述	337
18.1.2	块设备的结构	338
18.2	块设备驱动程序的架构	340
18.2.1	块设备加载过程	340
18.2.2	块设备卸载过程	341
18.3	通用块层.....	342
18.3.1	通用块层	342
18.3.2	<code>alloc_disk()</code> 函数对应的 <code>gendisk</code> 结构体.....	342
18.3.3	块设备的注册和注销	344
18.3.4	请求队列	345
18.3.5	设置 <code>gendisk</code> 属性中的 <code>block_device_operations</code> 结构体.....	346
18.4	不使用请求队列的块设备驱动	347
18.4.1	不使用请求队列的块设备驱动程序的组成	347
18.4.2	宏定义和全局变量	348
18.4.3	加载函数	349
18.4.4	卸载函数	350
18.4.5	自定义请求处理函数	351
18.4.6	驱动测试	352
18.5	I/O 调度器.....	355
18.5.1	数据从内存到磁盘的过程	355
18.5.2	块 I/O 请求 (<code>bio</code>)	355
18.5.3	请求结构 (<code>request</code>)	358
18.5.4	请求队列 (<code>request_queue</code>)	360
18.5.5	请求队列、请求结构、 <code>bio</code> 等之间的关系	361
18.5.6	四种调度算法	361
18.6	自定义 I/O 调度器.....	363
18.6.1	<code>Virtual_blkdev</code> 块设备的缺陷.....	363
18.6.2	指定 <code>noop</code> 调度器	363
18.6.3	<code>Virtual_blkdev</code> 的改进实例	364
18.6.4	编译和测试	365

18.7	脱离 I/O 调度器	365
18.7.1	请求队列中的 bio 处理函数	365
18.7.2	通用块层函数调用关系	367
18.7.3	对 Virtual_blkdev 块设备的改进	369
18.7.4	编译和测试	371
18.8	块设备的物理结构	372
18.8.1	为 Virtual_blkdev 块设备添加分区	372
18.8.2	对新的 Virtual_blkdev 代码的分析	373
18.8.3	编译和测试	374
18.8.4	分区数的计算	376
18.8.5	设置 Virtual_blkdev 的结构	377
18.8.6	编译和测试	379
18.9	小结	383
第 19 章	USB 设备驱动程序	384
19.1	USB 概述	384
19.1.1	USB 概念	384
19.1.2	USB 的特点	385
19.1.3	USB 总线拓扑结构	386
19.1.4	USB 驱动总体架构	386
19.2	USB 设备驱动模型	389
19.2.1	USB 驱动初探	389
19.2.2	USB 设备驱动模型	392
19.2.3	USB 驱动结构 usb_driver	394
19.3	USB 设备驱动程序	398
19.3.1	USB 设备驱动加载和卸载函数	398
19.3.2	探测函数 probe() 的参数 usb_interface	400
19.3.3	USB 协议中的设备	400
19.3.4	端点的传输方式	406
19.3.5	设置	408
19.3.6	探测函数 storage_probe()	410
19.4	获得 USB 设备信息	412
19.4.1	设备关联函数 associate_dev()	412
19.4.2	获得设备信息函数 get_device_info()	413
19.4.3	得到传输协议 get_transport() 函数	415
19.4.4	获得协议信息函数 get_protocol()	416
19.4.5	获得管道信息函数 get_pipes()	417
19.5	资源的初始化	420
19.5.1	storage_probe() 函数调用过程	420
19.5.2	资源获取函数 usb_stor_acquire_resources()	420
19.5.3	USB 请求块 (urb)	421