

嵌入式Linux驱动开发

版本：**v0.6**

Crifan Li

摘要

此文主要介绍的嵌入式Linux驱动开发的概念，逻辑，具体开发步骤，所要涉及哪些内容。



本文提供多种格式供：

在线阅读	HTML ¹	HTMLs ²	PDF ³	CHM ⁴	TXT ⁵	RTF ⁶	WEBHELP ⁷
下载（7zip压缩包）	HTML ⁸	HTMLs ⁹	PDF ¹⁰	CHM ¹¹	TXT ¹²	RTF ¹³	WEBHELP ¹⁴

HTML版本的在线地址为：

http://www.crifan.com/files/doc/docbook/embedded_linux_drv_dev/release/html/embedded_linux_drv_dev.html

有任何意见，建议，提交bug等，都欢迎去讨论组发帖讨论：

http://www.crifan.com/bbs/categories/embedded_linux_drv_dev/

修订历史

修订 0.6

2015-03-15

crl

1. 解释嵌入式Linux驱动开发的概念

¹ http://www.crifan.com/files/doc/docbook/embedded_linux_drv_dev/release/html/embedded_linux_drv_dev.html

² http://www.crifan.com/files/doc/docbook/embedded_linux_drv_dev/release/htmls/index.html

³ http://www.crifan.com/files/doc/docbook/embedded_linux_drv_dev/release/pdf/embedded_linux_drv_dev.pdf

⁴ http://www.crifan.com/files/doc/docbook/embedded_linux_drv_dev/release/chm/embedded_linux_drv_dev.chm

⁵ http://www.crifan.com/files/doc/docbook/embedded_linux_drv_dev/release/txt/embedded_linux_drv_dev.txt

⁶ http://www.crifan.com/files/doc/docbook/embedded_linux_drv_dev/release/rtf/embedded_linux_drv_dev.rtf

⁷ http://www.crifan.com/files/doc/docbook/embedded_linux_drv_dev/release/webhelp/index.html

⁸ http://www.crifan.com/files/doc/docbook/embedded_linux_drv_dev/release/html/embedded_linux_drv_dev.html.7z

⁹ http://www.crifan.com/files/doc/docbook/embedded_linux_drv_dev/release/htmls/index.html.7z

¹⁰ http://www.crifan.com/files/doc/docbook/embedded_linux_drv_dev/release/pdf/embedded_linux_drv_dev.pdf.7z

¹¹ http://www.crifan.com/files/doc/docbook/embedded_linux_drv_dev/release/chm/embedded_linux_drv_dev.chm.7z

¹² http://www.crifan.com/files/doc/docbook/embedded_linux_drv_dev/release/txt/embedded_linux_drv_dev.txt.7z

¹³ http://www.crifan.com/files/doc/docbook/embedded_linux_drv_dev/release/rtf/embedded_linux_drv_dev.rtf.7z

¹⁴ http://www.crifan.com/files/doc/docbook/embedded_linux_drv_dev/release/webhelp/embedded_linux_drv_dev.webhelp.7z

嵌入式Linux驱动开发:

Crifan Li

版本 : v0.6

出版日期 2015-03-15

版权 © 2015 Crifan, <http://crifan.com>

本文章遵从 : [署名-非商业性使用 2.5 中国大陆\(CC BY-NC 2.5\)](#)¹⁵

¹⁵ http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#cc_by_nc

目录

正文之前	iv
1. 此文目的	iv
1. Linux驱动模型	1
1.1. 为什么要设计一个Linux驱动模型	1
2. Linux中某模块所属的子驱动框架	2
3. 实现子驱动框架下面的设备相关部分的功能	4
4. 嵌入式Linux驱动开发相关资源	5
4.1. 嵌入式Linux驱动开发书籍	5
参考书目	6

正文之前

1. 此文目的

此文主要介绍的嵌入式Linux驱动开发的基本逻辑和如何开发。目的是搞懂如何进行嵌入式Linux的驱动开发。

第 1 章 Linux驱动模型

1.1. 为什么要设计一个Linux驱动模型

为了方便统一管理Linux下面的，如此众多的，成千上万的，设备的驱动，

所以要设计一个，统一的框架，即Linux驱动模型

此模型，就相当于一个模子，其他驱动，都要合乎这个模子，长得看起来都很类似，

如此才能注册到Linux驱动框架中，让驱动接受Linux系统的统一的管理。

第 2 章 Linux中某模块所属的子驱动框架

接下来，才是某个特定的设备的驱动的框架

想要实现某个特定的设备的驱动之前，就先要对该设备，在Linux的整体的驱动体系中，处于什么部分，属于什么驱动方面的子框架，要有所了解。

而这些子驱动框架，则是针对一类设备，统一管理，实现了该类设备的通用的功能和逻辑

这样作为驱动开发人员，可以免去这部分的开发的工作，省了很多心

比如：

- MTD
就是针对，内存类存储设备，而设计的一套框架，而属于内存类设备的，比如Nand Flash，Nor Flash等等，在编写驱动时，就要知道自己所要写的设备的驱动，是从属与MTD框架的

要知道，MTD框架，已经为我们所实现了哪些功能

比如MTD中的drivers/mtd/nand/nand_base.c中已经帮我们实现了，作为通用的MTD设备中的Nand Flash都会用到的：

- 通用的命令：nand_command，nand_command_lp
- 通用的数据读写：nand_read_buf，nand_write_buf
- 基于page的各种操作：nand_read_page_swecc，nand_read_page_hwecc，nand_write_page_raw，nand_write_page_hwecc
- 针对oob的各种操作：nand_read_oob_std，nand_write_oob_std
- 针对block的各种操作：nand_block_isbad，nand_block_markbad，nand_block_isreserved

如此众多的功能和函数，都是Linux的MTD框架都是帮你写好了的。

换句话说，如果没有Linux的MTD框架，让你从头到尾，一点点写出一个完整的Linux的Nand Flash的驱动的话，对于上述提到的各种针对oob，page，block，command等等的操作，都是需要你自己从无到有去实现出来的。

- USB
Linux中的USB子驱动框架，也是帮我们实现了太多的基础的通用的功能。包括class，interface，endpoint，等等方面的基本功能，比如：

- 所有USB类型都要用到的：drivers/usb/common/common.c
 - USB的速度：speed_names
 - USB的各种状态：usb_device_state
 - USB的模式：usb_dr_modes
- 以及各种和class类型相关的：
 - drivers/usb/core/devices.c
 - class类型定义：clas_info
 - USB设备的读取：usb_device_read

- 各种描述符的获取：usb_dump_device_descriptor , usb_dump_hub_descriptor , usb_dump_config_descriptor , usb_dump_interface_descriptor
- 以及某种class类型的内部所通用的功能
比如，最常见的USB大容量存储设备，即常说的U盘：
drivers/usb/storage/protocol.c
- UFI的命令：usb_stor_ufi_command
- SCSI的命令：usb_stor_transparent_scsi_command

上述列出了很多不同模块和领域的驱动相关的名称，概念，乍一看会显得难以理解，但是实际上不用太关心此处的细节，而只需要记住一点即可：

Linux的驱动，对于在某个子领域内，已经帮我们设计好了，适用于该领域的通用的框架，对应着某个驱动的子框架，且已经帮我们实现好了，大量的，常见和通用的功能

这样我们在实现对应设备驱动的时候，对于该设备所属的领域内的通用的功能，就可以不用自己再重新写了，就避免了重复造轮子的无用功的浪费了。

由此可见：在Linux下面写某设备的驱动，虽然需要增加额外的精力去学习该设备所属的驱动的子框架，但是这点额外的学习成本，和该框架帮你省下的精力相对，要划算的多。

即：Linux子驱动框架已经帮我们实现了非常多通用的功能部分，可以帮我们写驱动时，省掉很大一部分精力，而我们只需要搞懂子驱动框架后，去实现余下的和设备相关的部分即可。

第 3 章 实现子驱动框架下面的设备相关部分的功能

接下来，就是要去实现，该设备的所在的子驱动框架下，自己的设备相关的部分了。

往往也都是些：

- 设备初始化
- 数据的读写
 - 数据的读（取）read操作==数据发送send操作
 - 数据的写（入）write操作==数据接收receive操作

下面举一些例子来说明

- 实现Nand Flash驱动时要实现哪些部分
- 实现USB驱动时要实现哪些部分

第 4 章 嵌入式Linux驱动开发相关资源

4.1. 嵌入式Linux驱动开发书籍

关于嵌入式Linux驱动开发，有一些还不错的值得推荐的书：

- [Linux设备驱动开发详解 \(豆瓣\)](http://book.douban.com/subject/2984156/)¹

相对来说，这本书是我看到的，国内的在介绍嵌入式Linux驱动方面解释的算是逻辑性还不错的。

虽然也像之前某人评价的，当你对于嵌入式Linux驱动已经入门之后，了解了驱动框架，进入实际的驱动开发时，

你又会发现，实际上该书对你的帮助就不是那么大了，会觉得该书有点“虚”，解释的不是那么透彻和扎实了。

但是总体来说，还是值得一看了。

¹ <http://book.douban.com/subject/2984156/>

参考书目

- [1] [关于嵌入式驱动的学习？ - 知乎](#)¹
- [2] [硬件行业知识体系概要-CSDN论坛-CSDN.NET](#)²
- [3] [嵌入式开发之嵌入式背景知识](#)³
- [4] [嵌入式开发之基本概念和开发逻辑](#)⁴
- [5] [【问题解答】用一个嵌入式开发板去实现数据采集的基本思路和逻辑](#)⁵
- [6] [【整理】嵌入式领域的名词概念解释](#)⁶
- [7] [【整理】嵌入式开发经验和心得总结](#)⁷
- [8] [【整理】类比解释：嵌入式开发需要掌握哪些知识](#)⁸
- [9] [【整理】嵌入式软件知识体系概述](#)⁹

¹ <http://www.zhihu.com/question/21457018/answer/19517807>

² <http://bbs.csdn.net/topics/390487802>

³ http://www.crifan.com/embedded_development_background/

⁴ http://www.crifan.com/embedded_development_basic_concept_develop_logic/

⁵ http://www.crifan.com/qanda_embedded_data_acquisition_system_implement_process/

⁶ http://www.crifan.com/summary_embedded_concept_name_explanation/

⁷ http://www.crifan.com/summary_embedded_development_experience_and_note/

⁸ http://www.crifan.com/summary_use_similar_building_field_explain_embedded_system_need_learn_which_kind_of_knowledge/

⁹ http://www.crifan.com/embedded_system_software_field_knowledge_overview/