

linux input输入子系统分析《一》：初识input输入子系统

原创2012年07月29日 14:33:24 标签：input / linux / button / report / linux内核 / struct 11691

6

主要讲述本人在学习Linux内核input子系统的全部过程，如有分析不当，多谢指正。以下交流方式，文章欢迎转载，保留联系信息，以便交流。

邮箱：eabi010@gmail.com

主页：www.ielife.cn（爱嵌论坛——嵌入式技术学习交流）

博客：blog.csdn.net/ielife

1 开发环境

主 机：ubuntu10.04

开发板：mini2440

内 核：linux-2.6.22.6

编译器：arm-linux-gcc（3.4.5）

2 linux输入子系统

本节从整体上讲解了输入子系统的框架结构。有助于读者从整体上认识linux的输入子系统。在陷入代码分析的过程中，通过本节的知识能够找准方向，明白原理。

本节重点：

- 输入子系统的框架结构
- 各层对应内核中的文件位置
- 输入子系统的事件处理机制
- 输入子系统的驱动层基本操作流程
- 输入子系统的驱动层常用函数

本节难点：

- 输入子系统的事件处理机制
- 输入子系统的驱动工作流程

2.1 初识linux输入子系统

linux输入子系统（linux input subsystem）从上到下由三层实现，分别为：输入子系统事件处理层（EventHandler）、输入子系统核心层（InputCore）和输入子系统设备驱动层。

对于输入子系统设备驱动层而言，主要实现对硬件设备的读写访问，中断设置，并把硬件产生的事件转换为核心层定义的规范提交给事件处理层。

对于核心层而言，为设备驱动层提供了规范和接口。设备驱动层只要关心如何驱动硬件并获得硬件数据（例如按下的按键数据），然后调用核心层提供的接口，核心层会自动把数据提交给事件处理层。

对于事件处理层而言，则是用户编程的接口（设备节点），并处理驱动层提交的数据处理。

对于linux输入子系统的框架结构如下图1所示：

ielife

关注

原创	粉丝	喜欢
15	71	3

等级： 博客 3	访问量： 5
积分： 650	排名： 7万

高端网站设计

他的最新文章

音频处理与压缩的一些技术点和

CSR8670通过BLE拨打指定电话

CSR8670获取来电号码及开发流

苹果MFI bluetooth认证开发过

ce和accessory

bluetooth sniffer by wireshark

tu(linux)

文章分类

内核驱动

ARM裸机

Linux应用

音视频

C/C++

Object-C

展开

文章存档

2017年8月

2017年5月

2017年4月

2016年2月

2016年1月

2012年11月

展开

他的热门文章

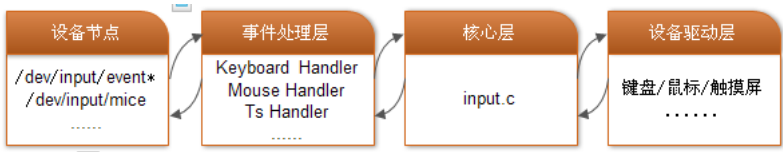


图1 linux输入子系统框架结构

由上图所展现的内容就是linux输入子系统的分层结构。

`/dev/input`目录下显示的是已经注册在内核中的设备编程接口，用户通过open这些设备文件来打开不同的输入设备进行硬件操作。

事件处理层为不同硬件类型提供了用户访问及处理接口。例如当我们打开设备`/dev/input/mice`时，会调用到事件处理层的Mouse Handler来处理输入事件，这也使得设备驱动层无需关心设备文件的操作，因为Mouse Handler已经有了对应事件处理的方法。

输入子系统由内核代码`drivers/input/input.c`构成，它的存在屏蔽了用户到设备驱动的交互细节，为设备驱动层和事件处理层提供了相互通信的统一界面。

下图2简单描述了linux输入子系统的事件处理机制：

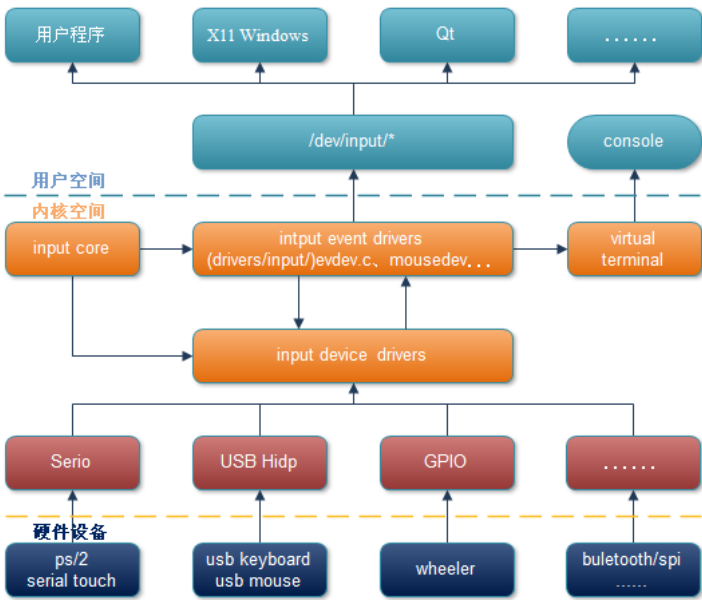


图2 linux输入子系统事件处理机制

由上图可知输入子系统核心层提供的支持以及如何上报事件到input event drivers。

作为输入设备的驱动开发者，需要做以下几步：

- 在驱动加载模块中，设置你的input设备支持的事件类型，类型参见表1设置
- 注册中断处理函数，例如键盘设备需要编写按键的抬起、放下，触摸屏设备需要编写按下、抬起、绝对移动，鼠标设备需要编写单击、抬起、相对移动，并且需要在必要的时候提交硬件数据（键值/坐标/状态等等）
- 将输入设备注册到输入子系统中

表1 Linux输入子系统支持的数据类型

	EV_SYN	0x00	同步事件
	EV_KEY	0x01	按键事件
	EV_REL	0x02	相对坐标(如：鼠标移动，报告相对最后一次位置的偏移)

- linux input输入子系统分析 《一》：初识input输入子系统 11681
- linux input输入子系统分析 《四》：t子系统整体流程全面分析 9396
- linux input输入子系统分析 《二》：440的ADC简单驱动实例分析 8362
- linux input输入子系统分析 《三》：440的触摸屏驱动实例 5390
- Linux外部模块.ko建立流程 2751
- 苹果MFI bluetooth认证开发过程及accessory 1812
- 推荐一本学习Linux高级编程入门 1666
- rtsp客户端做的事情(testRTSPClient) 1621
- live555ProxyServer.cpp分析：代码实现 1531
- 推荐一本ArmLinux系统底层的书籍 1096



十佳笔记本电脑



联系我们



- 请扫描二维码联系
- webmaster@csdn.net
- 400-660-0111
- QQ客服

关于 招聘 广告服务
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

EV_ABS	0x03	绝对坐标(如：触摸屏或操作杆，报告绝对的坐标位置)
EV_MSC	0x04	其它
EV_SW	0x05	开关
EV_LED	0x11	按键/设备灯
EV_SND	0x12	声音/警报
EV_REP	0x14	重复
EV_FF	0x15	力反馈
EV_PWR	0x16	电源
EV_FF_STATUS	0x17	力反馈状态

EV_MAX 0x1f 事件类型最大个数和提供位掩码支持

由表1可知，设备所能表示的事件种类，一个设备可以选择一个或多个事件类型上报给输入子系统。

Linux输入子系统提供了设备驱动层上报输入事件的函数，在include/linux/input.h中：

```
void input_report_key(struct input_dev *dev, unsigned int code, int value); //上报按键事件
void input_report_rel(struct input_dev *dev, unsigned int code, int value); //上报相对坐标事件
void input_report_abs(struct input_dev *dev, unsigned int code, int value); //上报绝对坐标事件
.....
```

当提交输入设备产生的输入事件之后，需要调用下面的函数来通知输入子系统，以处理设备产生的完整事件：

```
[cpp]
1. void input_sync(struct input_dev *dev);
```

2.2 输入设备驱动的简单案例

在Linux内核文档的documentation/input下，有一个input-programming.txt文件，讲解了编写输入设备驱动程序的核心步骤。

提供的案例代码描述了一个button设备，产生的事件通过BUTTON_PORT引脚获取，当有按下/释放发生时，BUTTON_IRQ被触发，以下是驱动的源代码：

```
[cpp]
1. #include <linux/input.h>
2. #include <linux/module.h>
3. #include <linux/init.h>
4.
5. #include <asm/irq.h>
6. #include <asm/io.h>
7.
8. static struct input_dev *button_dev;
9.
10. static void button_interrupt(int irq, void*dummy, struct pt_regs *fp)
11. {
12.     input_report_key(button_dev, BTN_1, inb(BUTTON_PORT) & 1);
13.     input_sync(button_dev);
14. }
15.
16. static int __init button_init(void)
17. {
18.     int error;
19.
20.     if (request_irq(BUTTON_IRQ, button_interrupt, 0, "button", NULL)) {
21.         printk(KERN_ERR "button.c: Can't allocate irq %d\n", button_irq);
22.         return -EBUSY;
23.     }
24.
25.     button_dev = input_allocate_device();
26.     if (!button_dev) {
27.         printk(KERN_ERR "button.c: Not enough memory\n");
28.         error = -ENOMEM;
29.         goto err_free_irq;
30.     }
```

```

31.
32.     button_dev->evbit[0] = BIT(EV_KEY);
33.     button_dev->keybit[LONG(BTN_0)] = BIT(BTN_0);
34.
35.     error = input_register_device(button_dev);
36.     if (error) {
37.         printk(KERN_ERR"button.c: Failed to register device\n");
38.         goto err_free_dev;
39.     }
40.
41.     return 0;
42.
43. err_free_dev:
44.     input_free_device(button_dev);
45. err_free_irq:
46.     free_irq(BUTTON_IRQ, button_interrupt);
47.     return error;
48. }
49.
50. static void __exit button_exit(void)
51. {
52.     input_unregister_device(button_dev);
53.     free_irq(BUTTON_IRQ, button_interrupt);
54. }
55.
56. module_init(button_init);
57. module_exit(button_exit);

```

编写基于输入子系统的设备驱动程序需要包含<linux/input.h>，因为它包含了输入子系统的接口和所有的宏定义，这些内容在编写输入设备驱动程序时需要用到。

button_init函数说明：

当模块加载（insmod）或内核引导过程中，button_init函数会被调用。首先做的工作是获取能够正确控制硬件设备的硬件资源（例如内存、IO内存、中断和DMA），在代码中BUTTON_IRQ作为BUTTON设备的中断资源，通过request_irq（）函数被申请注册。当有按键按下/释放时，调用button_interrupt（）中断处理函数获取按键值BUTTON_PORT（BUTTON设备的I/O资源）。

那么输入子系统怎么能够知道这个设备为输入设备呢？通过第8行为设备定义一个用于描述一个输入设备对象。

[cpp]

```
1. static struct input_dev *button_dev;
```

定义了button_dev之后，如何通知输入子系统有新的输入设备了呢？或者说如何把一个新的输入设备加入到输入子系统中呢？可以通过输入子系统核心层input.c中提供的函数分配一个输入设备，在代码的第25行。

[cpp]

```
1. button_dev= input_allocate_device();
```

有了输入设备的描述，当事件产生时，输入子系统怎么能够知道设备产生的事件类型呢？通过32和33行的代码。

[cpp]

```
1. button_dev->evbit[0]= BIT(EV_KEY);
2. button_dev->keybit[LONG(BTN_0)]= BIT(BTN_0);
```

其中evbit和keybit成员分别代表设备产生的事件类型和上报的按键值。其中输入子系统的一些位操作NBITS、BIT、LONG经常被用到：

[cpp]

```
1. #defineNBITS(x) (((x)/BITS_PER_LONG)+1)           //通过位x获取数组的长度
2. #defineBIT(x)      (1UL<<((x)%BITS_PER_LONG))      //返回位x在数组中的位域
3. #defineLONG(x) ((x)/BITS_PER_LONG)                 //返回位x的索引
```

以上的工作做完之后，即可注册为输入设备了，代码的35行。

[cpp]

```
1. input_register_device(button_dev);
```

这个函数把button_dev输入设备挂入输入设备链表中，并且通知事件处理层调用connect函数完成设备和事件处理的绑定，当用户打开设备时，便能够调用到相应的事件处理接口获得硬件上报的数据了。input_register_device（）函数是会睡眠的函数，因此不能够在中断上下文和持有自旋锁的代码中调用。

当我们把上面的工作做完之后，设备驱动中唯一值得关注的就是button_interrupt（）中断处理函数了。当按键动作发生，button_interrupt（）函数被调用，完成事件的上报由其中的两条语句完成。

[cpp]

1. input_report_key(button_dev, BTN_1, inb(BUTTON_PORT) & 1);
2. input_sync(button_dev);

其中input_report_key上报了这是一个按键事件，且它的值为inb(BUTTON_PORT) & 1，由于案例代码只产生一个按键的值，因此input_sync（）在这里不起关键作用。但如果是一个触摸屏，即有x坐标和y坐标，则需要通过input_sync（）函数把x和y坐标完整地传递给输入子系统。

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/ielife/article/details/7798952>

写下你的评论...

- qq_34888036

2017-12-22 19:43

#5楼

博主写得很好，一下子理清了头绪，谢谢

回复
- u011467781

2015-03-11 11:09

#4楼

学习了。。谢谢

回复
- chwenj

2013-09-22 09:16

#3楼

学习了

回复
- 查看 5 条热评

Linux驱动之输入子系统框架

lizuobin2

2016-05-26 17:06:04

6549

好记性不如烂笔头，整理一下笔记~ Linux驱动之输入子系统框架 输入子系统将该类驱动划分为3部分 1、核心层 input.c 2、设备层 Gpio_keys.c ... 3、事件处理层 Evd...

Input输入子系统个人总结

anlory

2016-06-03 15:17:55

1374

Input输入子系统个人总结Input输入子系统在驱动开发中的重要性，大家都知道，就不多说了。个人高驱动开发也有了一年了，突发奇想想写博客，来写写工作中常遇到的知识点，总结一下，为了更加清晰的理解。我...

你知道AI人工智能工程师有多缺乏吗？

21世纪什么最贵？人才啊.....



Linux输入子系统分析（一）

Leichelle

2012-05-14 23:27:09

12049

首先说明一下，本文是基于Linux-2.6.38版本内核来分析Linux输入子系统架构和原理的。这阵子本来没有打算花时间来分析Linux input system的，然而当在研究S3C6410触摸屏驱...

Linux输入子系统

qq_21792169

2015-09-22 21:34:01


10485

Linux输入子系统(Input Subsystem) Linux 的输入子系统不仅支持鼠标、键盘等常规输入设备,而且还支持蜂鸣器、触摸屏等设备。本章将对 Linux 输...

企业宣传片制作

广告画册设计

百度广告



input输入子系统整体流程



mike8825

2016-03-02 19:37:25

🔒 1645

input输入子系统整体流程 本节分析input子系统在内核中的实现，包括输入子系统（Input Core），事件处理层（Event Handler）和设备驱动层。由于上节代码讲解了设备驱动层的写法...

linux输入子系统概念介绍



lwj103862095

2014-01-01 15:08:47

🔒 6511

👍 6 此文章之前，我们讲解的都是简单的字符驱动，涉及的内容有字符驱动的框架、自动创建设备节点、linux中断、poll机制、异步通知、同步互斥、非阻塞、定时器去抖动。上一节文章链接：<http://blo...>

linux输入子系统框架分析及输入设备驱动编程




were0415

2016-11-29 16:22:23

🔒 805

linux输入子系统框架linux输入子系统（linux input subsystem）从上到下由三层实现，分别为：输入子系统事件驱动程序层（EventHandler）、输入子系统核心层（Input...

我对linux驱动 输入子系统的理解



zouleideboke

2017-04-24 22:40:40

🔒 834


前言：学习驱动有一段时间了，之前所学习的驱动都是驱动最原始的写法，也移植过很多驱动,然而并没有对内核自带的驱动进行深入的了解，于是出于好奇，就认真的学习了一下内核自带的驱动，我发现之前所学习的驱动都是...

Linux驱动之输入子系统

2017年01月12日 09:44

linux input输入子系统分析《四》：input子系统整体流程全面分析

主要讲述本人在学习Linux内核input子系统的全部过程，如有分析不当，多谢指正。以下方式均可联系，文章欢迎转载，保留联系信息，以便交流。 邮箱：eabi010@gmail.com 主页：www...




ielife

2012-07-31 14:29:38


🔒 9398

专门正对Exe及dll的加壳工具

堪比VMP安全强度的加壳工具，无需编程就能达到极高的保护强度



Linux内核Input输入子系统浅解



G_linuxer_


2016-07-01 16:03:37

🔒 1829

Linux输入设备总类繁杂，常见的包括有按键、键盘、触摸屏、鼠标、摇杆等等，他们本身就是字符设备，而linux内核将这些设备的共同性抽象出来，简化驱动开发建立了一个input子系统。子系统共分为三层， ...

[Linux]input 子系统学习笔记（简单范例和四个基本函数）

输入子系统是为了将输入设备的功能呈现给应用程序。它支持 鼠标、键盘、蜂鸣器、触摸屏、传感器等需要不断上报数据的设备。分析了四个函数：1. input_allocate_device 在内存中...



dearsq

2016-05-19 14:54:51

🔒 5985

Linux Input子系统浅析（二）-- 模拟tp上报键值

通过前一节的分析得到，linux Input子系统上传数据本质上是将input_dev的数据，上报给input_handler，当用户读入event时，驱动层只需要利用copy_to_user将数...




xiaopangzi313

2016-08-31 12:36:04

🔒 1372

Linux input子系统



bianyuke

2015-12-03 09:45:48

🔒 360

一、Input子系统分层思想 input子系统是典型的字符设备。首先分析输入子系统的工作机理。底层设备(按键、触摸等)发生动作时，产生一个事件(抽象)，CPU读取事件数据放入缓冲区，字符设备驱...

全网络对Linux input子系统最清晰、详尽的分析



yueqian_scut

2015-08-27 14:27:40

🔒 2597

本文应是全网对linux input子系统分析最有系统逻辑性和最清晰的分析文章了，主要结构input-core, input-handler和input-dev ice三者的关系以及应用open和rea...

一点点奶茶加盟费用 总成本预算是多少

加盟一点点



复制广告 6

Linux/Android——input子系统核心 (三) jscese 2014-12-26 15:10:07 5104

之前的博客有涉及到linux的input子系统，这里学习记录一下input模块. input子系统，作为管理输入设备与系统进行交互的中，任何的输入设备驱动都要通过input向内核注册其设备，常...

Linux Input子系统--概述 vanbreaker 2012-07-04 19:54:25 6064

水平有限，描述不当之处还请指出，转载请注明出处http://blog.csdn.net/vanbreaker/article/details/7714188 输入设备总类繁杂，包括按键...

Linux input子系统分析之一:软件层次 yueqian_scut 2015-08-23 10:49:35 3296

输入输出是用户和产品交互的手段，因此输入驱动开发在Linux驱动开发中很常见。同时，input子系统的分层架构思想在Linux驱动设计中极具代表性和先进性，因此对Linux input子系统。...

linux input 子系统分析 三 YAOZHENGUO2006 2011-09-14 19:48:59 7636

linux input子系统分析--子系统核心.事件处理层.事件传递过程 一. 输入子系统核心分析。 1.输入子系统核心对应与/drivers/input/input.c文件,这个也...

input子系统——kernel中input设备介绍 u013604527 2016-12-02 11:00:28 1189

一、输入子系统驱动层分析 在键盘驱动代码分析的笔记中，接触到了input子系统，键盘驱动，键盘驱动将检测到的所有按键都上报给了input子系统。Input子系统是所有I/O设备驱动的中...