Linux输入子系统学习笔记

——何良斌

1. **linux输入子系统的整体框架**

Linux输入子系统将输入驱动抽象为三层：事件处理层、核心层、设备驱动层。应用程序只需要跟事件处理层打交道，核心层是负责管理输入设备，并将消息在事件处理层和设备驱动层之间传递。Linux输入子系统整体框架如图所示。



1. **涉及的重要数据结构**
2. **驱动程的数据结构struct input\_dev**

用途：

驱动程input设备的基本数据结构，驱动的主体，需要完成的大部分工作围绕它进行，每个struct input\_dev代表一个输入设备。

位置：

Linux/input.h

数据结构的分配和初始化：

在具体的设备驱动中分配和填充具体的设备结构。

1. **事件处理层的数据结构struct input\_handler**

用途：

每个handler代表一种处理事件的方式，允许多个handler共存。

位置：

Linux/input.h

数据结构的分配和初始化：

在event handler层定义具体的struct input\_handler。

1. **数据结构stuct input\_handle**

用途：

用来创建驱动层struct input\_dev和struct input\_handler链表的链表项结构，使事件处理层和驱动层产生联系。

位置：

Linux/input.h

数据结构的分配和初始化：

Event Handler层中分配，包含在evdev.c/tsdev.c等中。

1. **内核统一使用的数据结构struct input\_event**

用途：

它是所有输入事件传送的载体，输入系统的事件都包装成该结构体传送给用户空间。

位置：

Linux/input.h

具体定义如下：

struct input\_event {

         struct timeval time;    /\* 时间戳   \*/

         \_\_u16 type;             /\* 事件类型\*/

         \_\_u16 code;             /\* 事件代码 \*/

         \_\_s32 value;            /\* 事件值，如坐标的偏移值 \*/

};

type：如EV\_KEY,键盘、EV\_REL,相对坐标、EV\_ABS,绝对坐标等

code：如果事件的类型代码是EV\_KEY,该代码code为设备键盘代码.代码值0~127为键盘上的按键代码,0x110~0x116 为鼠标上按键代码（在input.h中有详细定义）。

value：如果事件的类型代码是EV\_KEY,当按键按下时值为1,松开时值为0。

1. **输入设备节点操作示意图**

以下是用户空间操作输入设备的一个示意图，其中蓝色部分为input设备的公共处理，紫色部分evdev中的相关操作。

