**字符从键盘输入到在显示器显示出来的详细过程**

https://blog.csdn.net/sunwhut333/article/details/48557291

2015-09-18 23:18:18 [sunwhut333](https://me.csdn.net/sunwhut333) 阅读数 2535 文章标签： [os](https://so.csdn.net/so/search/s.do?q=os&t=blog) 更多

分类专栏： [java开发](https://blog.csdn.net/sunwhut333/article/category/3093149)

曾经有一段时间，一直想搞清楚，当我敲下键盘时，os到底是做了些什么，让我能看到一个字符被显示出来。

现把过程总结一下，消除一下和我有同样感觉的孩子们。

1）首先知道一下键盘的构成；6位的计数器经过两个八选一译码器对键盘进行扫描 在键盘没有任何按键时，扫描随着计数器的循环计数反复进行。

可以这么想，扫描就是一束光线，当没有键被按下时，光线可以横穿过键盘的任何位置。

2）若有键按下，扫描会发现该信号（键被按下以后挡住了光线的横穿），该信号使计数器停止计数，将扫描码发送给 8042。

每一个键的位置都用一组数字表示，这组数字就是所谓的扫描码。

你可能好奇，扫描码怎么就传给8042了？ 其实特简单，就是直接用电路联通的。

3）8042将该扫描码放在缓冲区中，并告诉8259A产生了中断。

缓冲区大小为1byte，如果此时键盘又有新的键按下，8042将不再接收，一直等到缓冲区被清理。

4）8259A识别出键盘中断，将cpu的中断寄存器的特定位置为1。

注意到目前为止cpu一直再执行其它程序或者空闲，根本还对键盘按下的事件没做任何反应。虽然描述的比较复杂，但这个过程其实特快。

5）接下来就是典型的中断处理过程了。 cpu执行完一条指令后，发现有中断存在，进入中断周期，通过IDT的映射将其处理交给键盘处理程序。

6）键盘处理程序第一步就是从8042的缓冲区(仅有一个字节)将该扫描码移动到键盘缓冲区（链表实现的 大小可调整，太大了也没什么用），此时可以接受下一个字符按键了。

7）若键盘缓冲区非空，则开始处理这些按键码，首先将扫描码转换为ASCII码

若是可打印字符如a 则将其显示

其他功能键做相应处理。

最后就是显示的过程了：

可通过bios调用或者写显存实现。

1 将该字符的ASCII码加上一些属性（如颜色控制，最简单的rgb，此时只有最纯的rgb三种颜色及其叠加的组合共8种），这些属性也是由01表示的，0代表无，1代表有 放入显存中（集显就是内存中的特定位置 ）。

2 字符发生器，通过读这个特定位置的数据，将ASCII码及其属性信息转换为一个光点矩阵，1表示有光。

字符发生器实际是一个rom，用映射的功能实现这种转换。

如ASCII = 65， 字符发生器会把其转换为a的光点矩阵，颜色由后面的3把枪控制。

3 该光点矩阵可以作为显示器的输入信号。每个点都有3把电子枪发射（分别发rgb）的光叠加成的。

**键盘的输入如何被系统识别**

　　键盘的按键信号被8048键盘编码器处理后发送给8042键盘控制器，8042接收和解码8048传递过来的数据。然后8048给8259A一个中断信号，表明用户有键盘上的输入动作了。8259A再给cpu一个信号，cpu响应后会调用键盘中断处理程序。至于为什么cpu能找到键盘中断而不会错误地调用硬盘中断。

1. IDT表中有各序号对应的处理函数，只要序号正确cpu就能找到正确的处理函数
2. 序号是8259A给的，8259A通过初始化时候的设置，确定好每个中断引脚对应的序号，发生中断的时候就可以得到正确的序号了。

**键盘中断**

　　　　键盘中断用来做什么事呢？很简单，就是从0x60端口读数据就可以了。另外如果不把0x60端口的数据读出来，那么下一次敲击键盘的信号就不能被8042接收了。

