# 第一章 键盘初始化

1 drv\_usb\_kbd\_init（）搜索键盘并在找到时注册，用for循环扫描所有USB设备调用usb\_get\_dev\_index根据该设备在设备列表的位置找到该设备，并调用函数usb\_kbd\_probe\_dev（）探测该设备是不是键盘。

2 usb\_kbd\_probe\_dev（USB设备，接口号）根据接口号获取该接口，根据接口类型、接口子类型、接口协议以及端点来判断该设备是不是键盘，如果不是的话不进行后续步骤。否则开辟键盘数据缓存区并清除此区域残留数据，进行输入缓冲区对齐并调整大小以进行 USB\_DMA对齐，将这个缓存区分配给当前设备。

设置当前设备的中断服务程序usb\_kbd\_irq，设置接收数据的中断管道、中断传输数据包大小、中断间隔以及上一次的报告。调用usb\_set\_protocol（★）函数设置协议安装USB键盘，调用usb\_set\_idle（★）函数设置空闲，最后调用函数usb\_int\_msg（★）测试该键盘是否可用。

# 第二章 键盘中断服务程序

本章为顺序结构依次向下看即可。

1 usb\_kbd\_irq（USB设备）先判断传输状态和数据长度是否符合要求，是的话调用usb\_kbd\_irq\_worker（）函数。

2 usb\_kbd\_irq\_worker（USB设备）在该函数中先判断有没有组合键按下，再判断是否按下两侧的“Ctrl”按键，for循环调用usb\_kbd\_service\_key（）函数判断按键按下与弹起。若按键仍被按下则调用usb\_kbd\_translate（已解释）函数，最后调用完成函数usb\_kbd\_setled（），并将data->new\_数据拷贝到data->old之中。

3 usb\_kbd\_service\_key（USB设备，键盘启动报告的第i个字节位置，弹起），如果up=1就反向new\_和old。用memscan（地址，值，大小）函数扫描从new\_ + 2地址开始长度为（USB\_KBD\_BOOT\_REPORT\_SIZE - 2）的范围内字符old[i]第一次出现的位置，如果该位置等于new\_ + USB\_KBD\_BOOT\_REPORT\_SIZE，则调用函数usb\_kbd\_translate（）函数。

4 usb\_kbd\_translate（数据，扫描码，修改—SHIFT标志，按下）如果按下标志是0表示按键释放直接返回，如果按下标志为2也直接返回。

判断扫描码在3--0x1d之间表示按下字母，先将按键字符设置为scancode - 4 + 'a'即小写字母，再根据数据标志判断是否按下了大写锁定就将按键字符设置为大写字母，再判断其是否按下了“Shift”以及同时按下“大写锁定”+“Shift”以改变大小写。

判断扫描码在0x1d--0x39之间表示按下键盘上方的数字键盘，其又根据有无“Shift”按下分为两种情况：如果有按下“Shift”就寻址特殊按键数组usb\_kbd\_numkey\_shifted[ ]得出按键字符，如果没有就寻址数字数组usb\_kbd\_numkey[ ]得出按键字符。

判断扫描码在0x54—0x67之间表示按下数字先键盘，直接寻址数字数组usb\_kbd\_num\_keypad[ ]得出案件字符。

如果是按下按键并且扫描码等于数字锁、大写锁定或者滚动锁定，就改变相应标志位后退出。

如果keycode不为0即有按下按键就调用函数usb\_kbd\_put\_queue（）报告按键字符。

如果扫描码在0x4e--0x53之间表示按下了“上、下、左、右”方向键，也调用usb\_kbd\_put\_queue（）报告按键字符。

5 usb\_kbd\_put\_queue（USB键盘数据指针，按键字符）此函数实现将字符放入队列并设置输入和输出指针。

①如果输入指针指向缓存区末尾并且输出指针指向缓存区开头表示缓存区数据已满，直接返回；②如果输入指针指向缓存区末尾并且输出指针不再指向开头，表示还有缓存区还有空位但是输入指针要指向缓存区开头；③如果输入指针没指向缓存区末尾，但是指向输出指针的前一个表示缓存区满，直接返回；④如果输入指针没指向缓存区末尾也没指向输出指针的前一个则指针加一指向缓存区。

上述过程执行完毕表示输入与输出指针设置完毕，向该缓存区写入按键字符data->usb\_kbd\_buffer[data->usb\_in\_pointer] = c。

**总结**：触发中断时old与new\_交替检测来判断键盘按键的变化，防止长按输出一大串相同字符。发现有变化的扫描码后，将此扫描码转换为相应的字符并存放在数据缓存区data->usb\_kbd\_buffer[ ]中。

# 第三章 检测是否有按键输入

1 usb\_kbd\_testc（）函数可实现检测是否有键盘输入，调用函数usb\_kbd\_poll\_for\_event（）实现中断轮询，如果数据缓存区输入指针等于输出指针表示数据缓存区为空返回0，否则返回1。

2 usb\_kbd\_poll\_for\_event（USB设备）函数实现中断轮询，调用usb\_int\_msg（★）提交中断传输请求，进而调用函数usb\_kbd\_irq\_worker（☆）执行中断服务程序。

# 第四章 获取输入字符

usb\_kbd\_getc（void）如果不存在USB键盘设备返回空，如果输入指针等于输出指针即数据缓存区为空则不断执行usb\_kbd\_poll\_for\_event（☆）实现中断轮询以积极获取数据。输出指针循环后移：如果输出指针指向数据缓存区最后一个存储空间，则将该输出指针置为0，否则让输出指针+1。返回数据缓存区中输出指针指向的字符data->usb\_kbd\_buffer[data->usb\_out\_pointer]。