1.1. 详细描述 NEMU 和 Kernel 响应时钟中断的过程和先前的系统调用过程不同之处在哪里？相同的地方又在哪里？可以通过文字或画图的方式来完成。

答：之前的系统调用是使用自陷指令（int），而响应时钟的系统调用则是利用外部中断来完成的。事实上，kernel对这两种中断的处理没有本质区别，都是进入中断处理函数然后根据中断号进行相应的跳转，但在nemu的触发方式上有区别，一个是自陷触发，另一个则是硬件（IRQ引脚触发）（考虑到nemu是一个模拟器，严谨的说法应是软件模拟的硬件触发）

1. 注册监听键盘事件是怎么完成的？

答：echo调用本文件的add\_irq\_handler函数，事实上是执行了一条中断号为0的int指令，在kernel中，int 0将会使处理函数进入kernel的add\_irq\_handler函数，在这里，将会有一个中断服务函数池，以链表形式存在。该函数在链表中增加一个节点，插入传入的处理函数。在遇到中断号对应的中断时，便会进入这个函数。  
2. 从键盘按下一个键到控制台输出对应的字符，系统的执行过程是什么？如果涉及与之前报告重复的内容， 简单引用之前的内容即可。

答：键盘按下一个键后键盘驱动程序将会作用，给物理的cpu一个中断并处理，此时nemu有一个进程在持续监听键盘，收到键盘的按键后将会把nemu的irq置1并传入相应的中断号，nemu根据这个中断号找到中断服务函数池（中断处理函数的注册见1）中的对应处理函数并执行.此时进入echo中的键盘处理函数执行，从键盘端口读取字符并将其转换为大写。然后又利用中断将字符送入串口并输出到屏幕。