

计算机系统基础

实验报告

PA 4

计算机科学与技术系
191220008 陈南曠

4-2:

1、注册监听键盘事件是怎么完成的？

首先查看 echo.c 的 main 函数：

```
add_irq_handler(1, keyboard_event_handler);
```

echo.c 的 main 函数调用了 add_irq_handler 函数，该函数通过 int0x80 指令调用系统函数 add_irq_handle，将对应的处理程序（函数指针）添加到 kernel 的异常处理程序中。

对键盘展开模拟时，键盘事件首先在 nemu/src/device/sdl.c 中由 NEMU_SDL_Thread() 线程捕获。NEMU 捕获两类事件：键盘按下和抬起。当检测到相应事件后，将对应键的扫描码作为参数传送给 keyboard.c 中的模拟键盘函数。模拟键盘缓存扫描码，并通过中断请求的方式通知 CPU 有按键或抬起的事件，键盘的中断请求号为 1。

```
void add_irq_handler(int irq, void *handler)
{
    // refer to kernel/src/syscall/do_syscall.c to understand what has happened
    asm volatile("int $0x80"
                  :
                  : "a"(0), "b"(irq), "c"(handler));
}
```

CPU 收到中断请求后调用 Kernel 的中断响应程序。在响应程序中，Kernel 会查找是否有应用程序注册了对键盘事件的响应，若有，则通过调用注册的响应函数的方式来通知应用程序。此时在应用程序的键盘响应函数中，可以通过 in 指令从键盘的数据端口读取扫描码完成数据交换。

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。

当我们真正按下一个物理按键的时候，我们会调用一个 `keyboard_down` 函数：

```
void keyboard_down(uint32_t sym)
{
    // put the scan code into the buffer
    scan_code_buf = sym2scancode[sym >> 8][sym & 0xff];
    // issue an interrupt
    i8259_raise_intr(KEYBOARD_IRQ);
    // maybe the kernel will be interested and come to read on the data port
}
```

将传进来的编码变成标准的扫描码 缓存到 `keyboard.c` 的 `scan_code_buf`，然后通过 `i8259` 产生 1 号中断。

之后会跳转到 `echo` 注册的处理程序执行。中间的过程见 `pa4-1` 报告。处理程序通过 `in` 指令从键盘 `io` 读入数据，并将其转换为 `ascii` 码，再通过系统调用将 `ascii` 码输出。