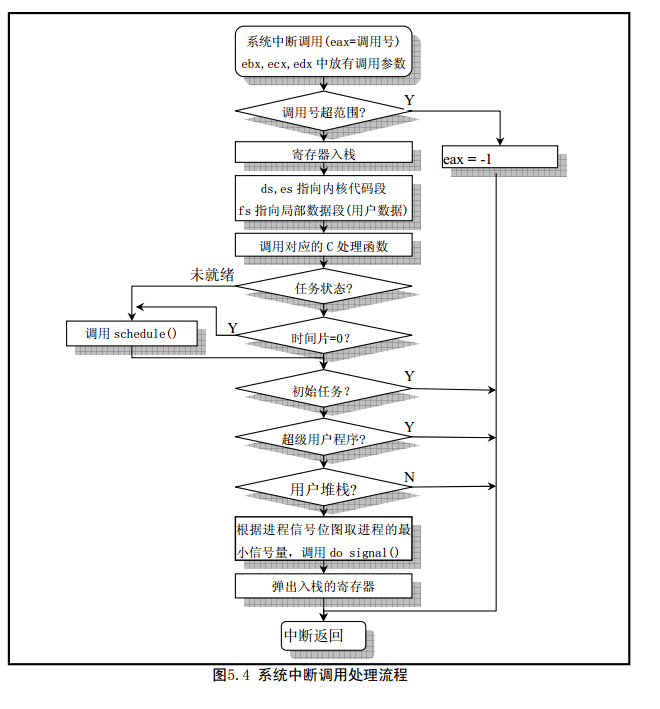


在开始执行程序之前，堆栈指针 esp 指在中断返回地址一栏(图中 esp0 处)。当把将要调用的 C 函数 do\_divide\_error()或其它 C 函数地址入栈后，指针位置是 esp1 处,此时通过交换指令，该函数的地址被放入 eax 寄存器中，而原来 eax 的值被保存到堆栈上。在把一些寄存器入栈后，堆栈指针位置在 esp2 处。当正式调用do\_divide\_error()之前，程序将开始执行时的 esp0 堆栈指针值压入堆栈，放到了 esp3 处，并在中断返回弹出入栈的寄存器之前指针通过加上 8 又回到 esp2 处。正 式调用 do\_divide\_error() 之 前把出错 代码以及 esp0 入栈 的原因是 为了作 为调用 C 函数do\_divide\_error()的参数。在 traps.c 中该函数的原形为：void do\_divide\_error(long esp, long error\_code)。因此在这个 C 函数中就可以打印出出错的位置和错误号。程序中其余异常出错的处理过程与这里描述的过程基本类似。



系统中断调用图

中断向量

