

寄存器的保护与恢复 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/460/2021\\_2022\\_\\_E5\\_AF\\_84\\_E5\\_AD\\_98\\_E5\\_99\\_A8\\_E7\\_c98\\_460690.htm](https://www.100test.com/kao_ti2020/460/2021_2022__E5_AF_84_E5_AD_98_E5_99_A8_E7_c98_460690.htm)

由于计算机的硬件资源只有一套，当子程序修改了寄存器的内容后，返回到调用它的程序时，这些寄存器的内容也就不会是调用子程序前的内容。这样，子程序修改寄存器内容就可能变成了调用它的副作用，这种副作用常常会导致调用程序的出错。为此，在编写子程序时，除了能对作为入口和出口参数的寄存器进行修改外，对其它寄存器的修改对调用程序来说都要是透明的，也就是说，在调用子程序指令的前后，除了作为入口和出口参数的寄存器内容可以不同外，其它寄存器的内容要保持不变。有时，也要求作为入口参数的寄存器内容保持不变。在子程序中，保存和恢复寄存器内容的主要方法是：在子程序的开始把它所用到的寄存器压进栈，在返回前，再把它们弹出栈。这样编写的好处是该子程序可以被任何其它程序来调用。在调用指令前，不需要保存寄存器，在调用指令后，也无需恢复寄存器。利用堆栈来保存和恢复寄存器内容方法的一般形式如下：XXXXXPROC PUSHREG1 ...

PUSHREGn.把子程序要使用的寄存器压栈，REGi代表某个寄存器 ... ..子程序的处理功能语句 POPREGn.把前面压栈的寄存器弹出，注意它们的次序 ... POPREG1 RET

XXXXXENDP 例7.2就是一个在子程序中利用堆栈来保存和恢复寄存器内容的例子。利用堆栈来实现此项功能时，应注意以下几点：、用堆栈保存和恢复寄存器的内容，要注意堆栈“先进后出”的操作特点；、通常情况下不保护入口参数寄

寄存器的内容，当然，也可以根据事先的约定而对它们加以保护；、如果用寄存器带回子程序的处理结果，那么，这些寄存器就一定不能加以保护；、整个子程序的执行几乎肯定要改变标志位，可用PUSHF和POPF来保护和恢复标志位，但一般在子程序中不保护标志位，除非有此特殊需要； 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)