

子程序的调用和返回指令 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/454/2021\\_2022\\_\\_E5\\_AD\\_90\\_E7\\_A8\\_8B\\_E5\\_BA\\_8F\\_E7\\_c98\\_454798.htm](https://www.100test.com/kao_ti2020/454/2021_2022__E5_AD_90_E7_A8_8B_E5_BA_8F_E7_c98_454798.htm) 子程序的调用和返回

是一对互逆操作，也是一种特殊的转移操作。一方面，之所以说是转移，是因为当调用一个子程序时，程序的执行顺序被改变，CPU将转而执行子程序中的指令序列，在这方面，调用子程序的操作含有转移指令的功能，子程序的返回指令的转移特性与此类似；另一方面，转移指令是一种“一去不复返”的操作，而当子程序完后，还要求CPU能转而执行调用指令之下的指令，它是一种“有去有回”的操作。为了满足子程序调用和返回操作的特殊性，在指令系统中设置了相应的特定指令。调用指令(CALL)调用子程序指令的格式如下：  
CALL 子程序名/Reg/Mem 子程序的调用指令分为近(near)调用和远(far)调用。如果被调用子程序的属性是近的，那么，CALL指令将产生一个近调用，它把该指令之后地址的偏移量(用一个字来表示的)压栈，把被调用子程序入口地址的偏移量送给指令指针寄存器IP即可实现执行程序的转移。近调用指令的堆栈操作。如果被调用子程序的属性是远的，那么，CALL指令将产生一个远调用。这时，调用指令不仅要把该指令之后地址的偏移量压进栈，而且也要把段寄存器CS的值压进栈。在此之后，再把被调用子程序入口地址的偏移量和段值分别送给IP和CS，这样完成了子程序的远调用操作。远调用指令的堆栈操。子程序调用指令本身的执行不影响任何标志位，但子程序体中指令的执行会改变标志位，所以，如果希望子程序的执行不能改变调用指令前后的标志位，那么

，就要在子程序的开始处保护标志位，在子程序的返回前恢复标志位。例如：CALL DISPLAY.DISPLAY是子程序名  
CALL BX.BX的内容是子程序的偏移量 CALL WORD1.WORD1  
是内存字变量，其值是子程序的偏移量 CALL  
DWORD1.DWORD1是双字变量，其值是子程序的偏移量和  
段值 CALL word ptr [BX].BX所指内存字单元的值是子程序的  
偏移量 CALL dword ptr [BX].BX所指内存双字单元的值是子程  
序的偏移量和段值 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)