计算机等级考试汇编语言教程之六 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/137/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E7_c98_137776.htm 4.0 利用子程序与中断 已经掌握了汇编语言？没错，你现在已经可以去破译别人代码中的秘密。然而，我们还有一件重要的东西没有提到，那就是自程序和中断。这两件东西是如此的重要，以至于你的程序几乎不可能离开它们。 4.1 子程序 在高级语言中我们经常要用到子程序。高级语言中，子程序是如此的神奇，我们能够定义和主程序，或其他子程序一样的变量名，而访问不同的变量，并且，还不和程序的其他部分相冲突。 然而遗憾的是，这种“ 优势 ”在汇编语言中是不存在的。 汇编语言并不注重如何减轻程序员的负担；相反，汇编语言依赖程序员的良好设计，以期发挥CPU的最佳性能。汇编语言不是结构化的语言，因此，它不提供直接的“ 局部变量 ”。如果需要“ 局部变量 ”，只能通过堆或栈自行实现。 从这个意义上讲，汇编语言的子程序更像GWBASIC中的GOSUB调用的那些“ 子程序 ”。所有的“ 变量 ”(本质上，属于进程的内存和寄存器)为整个程序所共享，高级语言编译器所做的，将局部变量放到堆或栈中的操作，只能自行实现。 参数的传递是靠寄存器和堆栈来完成的。高级语言中，子程序(函数、过程，或类似概念的东西)依赖于堆和栈来传递。 让我们来简单地分析一下一般高级语言的子程序的执行过程。无论C、C 、BASIC、Pascal，这一部分基本都是一致的。 调用者将子程序执行完成时应返回的地址、参数压入堆栈 子程序使用BP指针 偏移量对栈中的参数寻址，并取出、完成操作 子程序使

用RET或RETF指令返回。此时，CPU将IP置为堆栈中保存的地址，并继续予以执行 毋庸置疑，堆栈在整个过程中发挥着非常重要的作用。不过，本质上对子程序最重要的还是返回地址。如果子程序不知道这个地址，那么系统将会崩溃。 调用子程序的指令是CALL，对应的返回指令是RET。此外，还有一组指令，即ENTER和LEAVE，它们可以帮助进行堆栈的维护。 CALL指令的参数是被调用子程序的地址。使用宏汇编的时候，这通常是一个标号。CALL和RET，以及ENTER和LEAVE配对，可以实现对于堆栈的自动操作，而不需要程序员进行PUSH/POP，以及跳转的操作，从而提高了效率。 作为一个编译器的实现实例，我用Visual C 编译了一段C 程序代码，这段汇编代码是使用特定的编译选项得到的结果，正常的RELEASE代码会比它精简得多。包含源代码的部分反汇编结果如下(取自Visual C 调试器的运行结果，我删除了10条int 3指令，并加上了一些注释，除此之外，没有做任何修改)： 1: int myTransform(int nInput){ 00401000 push ebp . 保护现场原先的EBP指针 00401001 mov ebp,esp 2: return (nInput*2 3) % 7. 00401003 mov eax,dword ptr [nInput] . 取参数 00401006 lea eax,[eax eax 3] . LEA比ADD加法更快 0040100A cdq 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